**COURSE OVERVIEW**  
Course on Apache Spark & Scala is a 4 days(32 hours) course which will cover different concepts of Big Data, Challenges in Big Data Processing, Approach to Big Data Problems using Apache Spark, specifics of Spark like it's Components, Installation Steps, RDDs, Transformations, Actions, Lazy Execution, Integration with HDFS .

**Objective**:

After the completion of this course, you will be able to:

. Learn scala programming

* . Understand Big Data and the challenges associated
* Find an approach to Big Data problems with Apache Spark
* Implement Apache Spark Concepts
* Apply Scala for Spark
* Understand data frame concept and How to run SQL queries using Spark-SQL
* Follow latest emerging trends like MLib, GraphX based on Spark

**Prerequisite:**

1. Knowledge of Hadoop Eco-system
2. Knowledge of Scala

**Hardware/Software requirements:**

1. 8 GB RAM windows machine
2. Internet connection for setting up SBT/Maven project
3. Virtualization feature on the machine should be enabled

**COURSE SYLLABUS**

**Day 1**

Basics:

* Hello World
* Primitive Types
* Type inference
* Vars vs Vals
* Lazy Vals
* Methods
* Pass By Name
* No parens/Brackets
* Default Arguments
* Named Arguments

Classes:

* Introduction
* Inheritance
* Main/Additional Constructors
* Private Constructors
* Uniform Access
* Case Classes
* Objects
* Traits

Collections:

* Lists
* Collection Manipulation
* Simple Methods
* Methods With Functions
* Use Cases With Common Methods
* Tuples

Types:

* Type parameterization
* Covariance
* Contravariance
* Type Upper Bounds
* 'Nothing' Type

Options:

* Option Implementation
* Like Lists
* Practice Application

Anonymous Classes:

* Introduction
* Structural Typing
* Anonymous Classes With Structural Typing

Special Methods:

* Apply
* Update

Closure and functions

Currying:

* Introduction
* Applications

Implicits:

* Implicit Values/Parameters
* Implicit Conversions
* With Anonymous Classes
* Implicit Classes

For Loops:

* Introduction
* Coding Style
* With Options
* And flatMap
* Guards
* Definitions

Var Args:

* Introduction
* Ascribing the \_\* type

Partial Functions:

* Introduction
* Match
* Match Values/Constants
* Match Types
* Extractors
* If Conditions
* Or

**Working with XML & JSON**

**Day 2:**

**Introduction of Spark**

Evolution of distributed systems

Why we need new generation of distributed system?

Limitation with Map Reduce in Hadoop,

Understanding need of Batch Vs. Real Time Analytics

Batch Analytics - Hadoop Ecosystem Overview, Real Time Analytics Options

Introduction to stream and in memory analysis

What is Spark?

A Brief History: Spark

**Honds-On**

1. **Installing Spark and sbt**
2. **Integrating Spark in Eciplse**
3. **Running Spark in Eclipse and Spark Standalone cluster**

**Using Scala for creating Spark Application**

Invoking Spark Shell

Creating the SparkContext

Loading a File in Shell

Performing Some Basic Operations on Files in Spark Shell

Building a Spark Project with sbt

Running Spark Project with sbt, Caching Overview

Distributed Persistence

Spark Streaming Overview

Example: Streaming Word Count

Testing Tips in Scala

Performance Tuning Tips in Spark

Shared Variables: Broadcast Variables

Shared Variables: Accumulators

**Day 3:**

**Running SQL queries using Spark SQL**

## Starting Point: SQLContext

## Creating DataFrames

## DataFrame Operations

## Running SQL Queries Programmatically

## Interoperating with RDDs

### Inferring the Schema Using Reflection

### PInferring the Schema Using Reflection

# Data Sources

## Generic Load/Save Functions

### Save Modes

### Saving to Persistent Tables

## Parquet Files

### Loading Data Programmatically

### Partition Discovery

## Schema Merging

## JSON Datasets

## Hive Tables

## JDBC To Other Databases

## Troubleshooting

# Performance Tuning

## Caching Data In Memory

## Compatibility with Apache Hive

### Unsupported Hive Functionality

**Honds-On**

1. **Running SQL Quries with MySql**
2. **Running Hive queries**
3. **Reading JSON file and storing it as a Parquet format**

**Spark Streaming**

Micro batch

Discretized Streams (DStreams)

Input DStreams and Receivers

Dstream to RDD

Basic Sources

Advanced Sources

Transformations on DStreams

Output Operations on DStreams

Design Patterns for using foreachRDD

DataFrame and SQL Operations

Checkpointing

Socket stream

File Stream

Stateful operations

How stateful operations work?

Window Operations

Join Operations

**Honds-On**

1. **Network-wordcount with Spark Streaming**
2. **Processing Flume data with Spark Streaming**
3. **Processing Kafka data with Spark Streaming**
4. **Processing Twitter data with Spark Streaming**

**Day 4:**

# **Spark ML Programming**

Main Concepts

ML Dataset

ML AlgorithmsModel Selection via Cross-Validation

**Honds-On**

1. **Clustering with K-means**
2. **Classification examples**
3. **Linear regression techniques**

**Tuning Spark**

[Data Serialization](http://spark.apache.org/docs/latest/tuning.html#data-serialization)

[Memory Tuning](http://spark.apache.org/docs/latest/tuning.html#memory-tuning)

[Determining Memory Consumption](http://spark.apache.org/docs/latest/tuning.html#determining-memory-consumption)

[Tuning Data Structures](http://spark.apache.org/docs/latest/tuning.html#tuning-data-structures)

[Serialized RDD Storage](http://spark.apache.org/docs/latest/tuning.html#serialized-rdd-storage)

[Garbage Collection Tuning](http://spark.apache.org/docs/latest/tuning.html#garbage-collection-tuning)

[Other Considerations](http://spark.apache.org/docs/latest/tuning.html#other-considerations)

[Level of Parallelism](http://spark.apache.org/docs/latest/tuning.html#level-of-parallelism)

[Memory Usage of Reduce Tasks](http://spark.apache.org/docs/latest/tuning.html#memory-usage-of-reduce-tasks)

[Broadcasting Large Variables](http://spark.apache.org/docs/latest/tuning.html#broadcasting-large-variables)

[Data Locality](http://spark.apache.org/docs/latest/tuning.html#data-locality)

[Summary](http://spark.apache.org/docs/latest/tuning.html#summary)

**Job Scheduling and Monitoring**

[Overview](http://spark.apache.org/docs/latest/job-scheduling.html#overview)

[Scheduling Across Applications](http://spark.apache.org/docs/latest/job-scheduling.html#scheduling-across-applications)

[Dynamic Resource Allocation](http://spark.apache.org/docs/latest/job-scheduling.html#dynamic-resource-allocation)

[Configuration and Setup](http://spark.apache.org/docs/latest/job-scheduling.html#configuration-and-setup)

[Resource Allocation Policy](http://spark.apache.org/docs/latest/job-scheduling.html#resource-allocation-policy)

[Request Policy](http://spark.apache.org/docs/latest/job-scheduling.html#request-policy)

[Remove Policy](http://spark.apache.org/docs/latest/job-scheduling.html#remove-policy)

[Graceful Decommission of Executors](http://spark.apache.org/docs/latest/job-scheduling.html#graceful-decommission-of-executors)

[Scheduling Within an Application](http://spark.apache.org/docs/latest/job-scheduling.html#scheduling-within-an-application)

[Fair Scheduler Pools](http://spark.apache.org/docs/latest/job-scheduling.html#fair-scheduler-pools)

[Default Behavior of Pools](http://spark.apache.org/docs/latest/job-scheduling.html#default-behavior-of-pools)

[Configuring Pool Properties](http://spark.apache.org/docs/latest/job-scheduling.html#configuring-pool-properties)