**Description**

**Apache Hadoop**, the open source data management software that helps organizations analyze massive volumes of structured and unstructured data, is a very hot topic across the tech industry. Employed by such big named websites as eBay, Facebook, and Yahoo, Hadoop can be hosted on cloud environment like **Windows HDinsight Service** , where we need to pay for the computing resource we use.

Class format is 50% Lecture/50% Lab. Ten hands-on exercises make up the lab portion of the class that include setting up Hadoop in pseudo distributed mode, managing files in HDFS, writing map reduce programs in Java, Hadoop monitoring, sqoop, hive and pig.

**COURSE OUTLINE :**

**Day 1**

**Duration : 3 Hours**

Introduction to BigData

• Which data is called as BigData

• What are business use cases for BigData

• BigData requirement for traditional Data warehousing and BI space

• BigData solutions

Introduction to Hadoop

• The amount of data processing in today’s life

• What Hadoop is why it is important

• Hadoop comparison with traditional systems

• Hadoop history

• Hadoop main components and architecture

Hadoop Distributed File System (HDFS)

• HDFS overview and design

• HDFS architecture

• HDFS file storage

• Component failures and recoveries

• Block placement

• Balancing the Hadoop cluster

Hadoop Deployment

• Different Hadoop deployment types

• Hadoop distribution options

• Hadoop competitors

• Hadoop installation procedure

• Distributed cluster architecture

• Lab: Hadoop Installation

Working with HDFS

• Ways of accessing data in HDFS

• Common HDFS operations and commands

• Different HDFS commands

• Internals of a file read in HDFS

• Data copying with ‘distcp’

• Lab: Working with HDFS

**Durtion : 5 hours**

**Map-Reduce Abstraction**

• What MapReduce is and why it is popular

• The Big Picture of the MapReduce

• MapReduce process and terminology

• MapReduce components failures and recoveries

• Working with MapReduce

• Lab: Working with MapReduce

**Programming MapReduce Jobs**

• Java MapReduce implementation

• Map() and Reduce() methods

• Java MapReduce calling code

• Lab: Programming Word Count

**Input/Output Formats and Conversion Between Different Formats**

• Default Input and Output formats

• Sequence File structure

• Sequence File Input and Output formats

• Sequence File access via Java API and HDS

• MapFile

• Lab: Input Format

• Lab: Format Conversion

**MapReduce Features**

• Joining Data Sets in MapReduce Jobs

• How to write a Map-Side Join

• How to write a Reduce-Side Join

• MapReduce Counters

• Built-in and user-defined counters

• Retrieving MapReduce counters

• Lab: Map-Side Join

**YARN (Hadoop2.0) features:** In this class, you will learn what is Yarn and its components**.** We shall how YARN has become the architectural center of Hadoop that allows multiple data processing engines such as **interactive SQL, real-time streaming, data science and batch processing** to handle data stored in a single platform, unlocking an entirely new approach to analytics. We shall look into **Giraph** which is an iterative graph processing system built for high scalability to solve some problem more effectively by processing data as a graph in the Hadoop.

**Day 2**

**Duration : 4 hours**

**Hive -** This class will help you in understanding Hive concepts, Loading and Querying Data in Hive and Hive UDF.

**Topics -**Hive Background, Hive Use Case, About Hive, Hive Vs Pig, Hive Architecture and Components, Metastore in Hive, Limitations of Hive, Comparison with Traditional Database, Hive Data Types and Data Models, Partitions and Buckets, Hive Tables(Managed Tables and External Tables), Importing Data, Querying Data, Managing Outputs, Hive Script, Hive UDF, Hive Demo on Healthcare Data set. Different kinds of file format suppoted in Hive such as ORC, Avro, Parquet and Sequencefile.

**Kafka:**

**Duration : 4 hours**

* Introduction
* Basic Kafka Concepts
* Kafka vs Other Messaging Systems
* Intra-Cluster Replication
* An Inside Look at Kafka’s Components
* Log Administration, Retention, and Compaction
* Hardware and Runtime Configurations
* Monitoring and Alerting
* Cluster Administration
* Securing Kafka
* Using Kafka Connect to Move Data
* Writing data into Kafka
* Reading data from Kafka

**Day 3:**

**Duration : 4 hours**

**NoSQL :**This class will cover NoSQL in general and HBase in particular. We will see demos on Bulk Loading , Filters. You will also learn what Zookeeper is all about, how it helps in monitoring a cluster, why HBase uses Zookeeper.

**Topics -**HBase Data Model, HBase Shell, HBase Client API, Data Loading Techniques, ZooKeeper Data Model, Zookeeper Service, Zookeeper, Demos on Bulk Loading, Getting and Inserting Data, Filters in HBase.,

**Duration : 4 hours**

**Flume , Sqoop and Nifi**

**Data Loading** : Here we will learn different data loading options available in Hadoop and will look into details about Flume and Sqoop to demonstrate how to bring various kind of files such as Web server logs , stream data, RDBMS, twetter ‘s tweet into HDFS

**Day 4.**

**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 Jupyter**
2. **Running Spark in Jupyter and Spark Standalone cluster**

**Using Python for creating Spark Application**

Invoking Spark Shell

Creating the SparkContext

Loading a File in PySpark Shell

Performing Some Basic Operations on Files in Spark Shell

Caching Overview

Distributed Persistence

Spark Streaming Overview

Example: Streaming Word Count

Performance Tuning Tips in Spark

Shared Variables: Broadcast Variables

Shared Variables: Accumulators

**Day 5**

**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**

**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)

.

**Hadoop Project Discussion:**

 A detailed brainstorming on specific functional use cases and technical questions.