**NAME**

Big Data Processing with Hadoop and Spark

**PRE-REQUISITE**

1. Knowledge of Java

2. Understanding of SQL queries

**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 **Hortonworks, Cloudera , MapR etc**, 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**

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

Job Scheduling

• How to schedule Hadoop Jobs on the same cluster

• Default Hadoop FIFO Schedule

• Fair Scheduler and its configuration

**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

**Day 2**

**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 3**

**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.

**SQL in Hadoop :** We shall look into different SQL engine avaialbale in hadoop such as Imapla , Hawq, BigSQL and Stinger . we shall learn detail about writing queries in Imapla and its use cases

**Impala detail discussion**: Architecture of Impala, running hive queries in Impala, working with different format of files such as RC, ORC, SequenceFile and Parquet files.

**Hands On**:

• Understanding the map reduce flow in the Hive-SQL

• Creating Static partition table

• Creating Dynamic partition table

• Loading a unstructured text file into table using Regex serde

• Loading a JSON file into table using Json serde

• Creating transaction table

• Creating view and indexes

• Creating ORC, Parquet tables and using compression techniques

• Creating Sequence file table

Writing Java code for UDF

Writing JAVA code to connect with Hive and perform CRUD Operations using JDBC

**Day 4**.

**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.

**Flume , Sqoop and OOZIE**

**Learning Objectives -**In this clss, you will understand working of multiple Hadoop ecosystem components together in a Hadoop implementation to solve Big Data problems. We will discuss multiple data sets and specifications of the project. This module will also cover demo of Apache Oozie Workflow Scheduler for Hadoop Jobs.

**Topics -** Oozie, Oozie Components, Oozie Workflow, Scheduling with Oozie, Demo on Oozie Workflow, Oozie Co-ordinator, Oozie Commands, Oozie Web Console, Hadoop Project Demo.

**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.,

**Yarn Vs Hadoop 1.X**: Understand the architecture of YARN and role of different components such as resource manager, Node Manager and App Master. We shall also understand different kind of processing possible with YARN.

**Kafka** :

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

**Hands On**:

• Using flume to capture and transport network data

• Using flume to capture and transport web server log data

• Using flume to capture and transport Twitter data

• Creating topic and configuring replication factor and no of partition for the same

• Loading data into Kafka topic

• Reading data from Kafka topic

• Using Sqoop to capture RDBMS data into HDFS

• Using Sqoop to capture RDBMS data into Hive

• Using Sqoop to capture RDBMS into Hbase

• Using sqoop exporting data into RDBMS from HDFS

Kafka: Writing Producer and Consumer Java code

Sync and async producer hands on

• Creating Hbase tables and column families

• Writing code for Loading data into Hbase table

• Writing codefor Performing CRUD operations in Hbase

• Creating external table in Hive for integrating Hbase with Hive

• Writing code for Loading data into Hbase using MapReduce

• Writing code for Creating Hbase tables and column families

• Creating and executing Oozie work flow

**Day 5**.

**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

**Using Java 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 maven

Distributed Persistence

Spark Streaming Overview

Example: Streaming Word Count

Performance Tuning Tips in Spark

Shared Variables: Broadcast Variables

Shared Variables: Accumulators

**Hands On**:

• Installing Spark

• Installing SBT and maven for building the project

• Writing code for Converting HDFS data into RDD

• Writing code for Performing different transformation and action

• Understanding tasks, stages related to spark job

• Writing code for using different storage levels and Caching

• Creating broadcast and accumulators and using them

**Day 6**

**Lesson 2: 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

**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

**Hands On**:

Writing code for Creating SparkContext , HiveContext and HbaseContext objects

• Writing code for Running Hive queries using Spark-SQL

• Writing code Loading , transforming text file data and converting that into Dataframe

• Writing code Reading and storing JSON files as Dataframes inside the spark code

• Writing code for Reading and storing PERQUET files as Dataframes

• Reading and Writing data into RDBMS (MySQL for example) using Spark-SQL

• Caching the dataframes

• Writing code for Processing Flume data using Spark streaming

• Writing code for Processing network data using Spark streaming

• Writing code for Processing Kafka data using Spark streaming

• Writing code for Storing the output into HDFS using 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)