Hadoop Development

Introduction

- What is Bigdata?
- Evolution of Bigdata
- Types of Data and their Significance
- Need for Bigdata Analytics
- Why Bigdata with Hadoop?
- History of Hadoop
- Why Hadoop is in demand in market nowadays?
- Limitations of SQL based Tools
- Hadoop Nodes
- Hadoop Rack
- Hadoop Cluster
- Architecture of Hadoop
- Characteristics of Namenode
- Workaround with Datanodes
- Significance of JobTracker and Tasktrackers
- Hase co-ordination with JobTracker
- Secondary Namenode usage and Workaround
- Hadoop Releases and their Significance
- Introduction to Hadoop Release-1
- Hadoop Daemons in Hadoop Release-1
- Introduction to Hadoop Release-2
- Hadoop Daemons in Hadoop Release-2
- Hadoop Cluster Demo
- Hadoop 2.x Cluster Architecture
- A Typical Production Hadoop Cluster
- Hadoop Cluster Modes
- Hadoop 2.x Configuration Files
- Single node cluster and Multi node cluster setup
- Hadoop installation
- Introduction to Hadoop FS and Processing Environment's UIs
- How to read and write files
- Basic Unix commands for Hadoop
- Hadoop FS shell
- Hadoop releases practical
- Hadoop daemons practical
- Common Hadoop Shell Commands
- An Overview of Hadoop Administration
- How Hadoop is getting two categories Projects
- New projects on Hadoop
- Hadoop Storage – HDFS (Hadoop Distributed file system)
- Hadoop Processing Framework (Map Reduce / YARN)

- Alternates of Map Reduce
- Why NOSQL is in much demand instead of SQL
- Distributed warehouse for HDFS
- YARN Architecture
- Significance of Scalability of Operation
- Use cases where not to use Hadoop
- Use cases where Hadoop Is used
- Facebook,Twitter,Snapdeal, Flipkart

Hadoop Java API

- Hadoop Classes
- What is MapReduceBase?
- Mapper Class and its Methods
- What is Partitioner and types
- MapReduce Use Cases
- Traditional way VS MapReduce way
- Significance of MapReduce
- Hadoop 2. X MapReduce Architecture
- Hadoop 2. MapReduce Program
- Understanding Input Splits
- Relationship between Input Splits and HDFS Blocks
- MapReduce: Combiner & Partitioner
- Hadoop specific Data types
- Working on Unstructured Data Analytics
- What is an Iterator and its usage techniques
- Types of Mappers and Reducers
- What is Output collector and its Significance
- Workaround with Joining of datasets
- Complications with MapReduce
- Mapreduce Anatomy
- Anagram example,Teragen Example,Terasort Example
- WordCount Example
- Working with multiple mappers
- Working with weather data on multiple Data nodes in a Fully distributed Architecture
- Use Cases where MapReduce anatomy fails
- Advanced MapReduce
- Counters
- Distributed Cache
- MRunit
- Joins in MapReduce
- Reduce Side Join
- Replicated Join
- Composite Join
- Cartesian Product
- Custom Input Format

- Sequence Input Format
- XML File Parsing using MapReduce
- Interview questions based on JAVA MapReduce

Pig Latin - Basic Level

- Introduction to Pig Latin
- History and Evolution of Pig Latin
- Why Pig is used only with Bigdata
- MapReduce VS Pig
- Pig Architecture and Overview of Compiler and Execution Engine
- Programming Structure in Pig
- Pig Running Modes
- Pig Components
- Pig Execution
- Pig Release and Significance of Bugfixes
- Pig Specific Datatypes
- Complex Datatypes
- Bags, Tuples, Fields
- Pig Specific Methods
- Comparison between Yahoo Pig & Facebook Hive
- Shell and Utility Commands
- Working with Grunt Shell
- Grunt commands: 17 in number
- Pig Latin: Relational Operators
- Pig Latin: File Loaders
- Pig Latin: Group Operator
- Cogroup Operator
- Joins and Cogroup
- Union
- Understanding Diagnostic Operators
- Specialized Joins in Pig
- Built in Functions
- Eval Function
- Load and Store Functions
- Math Function
- String Function
- Date Function
- Pig UDF
- Piggybank
- Parameter Substitution
- Pig Streaming
- Pig Use Cases: Aviation and Healthcare
- Pig Data Input Techniques for flatfiles
- Flatfiles: Comma separated, Tab delimited, and fixed width
- Working with Schemaless Approach

- How to attach Schema to a file/table in Pig
- Schema referencing for similar Tables and Files
- Working with Delimiters

Pig Latin - II (Expert)

- Working with Binary Storage and Text Loader
- Bigdata Operations and Read write Analogy
- Filtering Datasets
- Filtering rows with specific condition
- Filtering rows with multiple conditions
- Filtering rows with String Based Conditions
- Sorting DataSets
- Sorting rows with Specific column or columns
- Multi level Sort
- Analogy of a Sort Operation
- Grouping Datasets and Co-grouping data
- Joining DataSets
- Types of Joins supported by Pig Latin
- Aggregate Operations like average, sum, min, max, count
- Flatten Operator
- Creating a UDF (USER DEFINED FUNCTION) using java
- Calling UDF from a Pig Script
- Data validation Scripts

Hive

- Overview of Hive
- Background of Hive
- Hive VS Pig
- Installation and Configuration
- Interacting HDFS using HIVE
- Map Reduce Programs through HIVE
- Hive Architecture and Components
- Hive Commands
- Loading, Filtering, Grouping
- What is Meta Storage and Meta Store
- Derby Database
- HQL
- DDL, DML, and other Sub Languages of Hive
- Data types in Hive
- Partitions and Buckets
- Hive Tables: Managed and External
- Importing Data
- Querying Data
- Managing Outputs

- Hive Scipts
- Hive UDF
- Hive Operators
- Hive Joins, Unions, and Groups
- Sample Programs in Hive
- Alter and Delete in Hive
- Partition in Hive
- Indexing
- Industry Specific Configuration of Hive Parameters
- Authentication & Authorization
- Statistics with Hive
- Archiving in Hive
- Hands-on exercise

Advanced Hive

- Understanding Hive Releases
- Hive and OLTP
- OLAP in Hive
- Hive QL: Joining Tables
- Dynamic Partitioning & Bucketing
- Serialization and Deserialization
- Custom Map/Reduce Scripts
- Hive Indexes and Views
- Hive Query Optimizers
- Hive Architecture
- Understanding Thrift Server
- User Defined Functions
- Hue Interface for Hive
- Analyzing Data with Hive Script
- Difference between Hive and Impala
- UDFs in Hive
- Complex Use cases in Hive

Hadoop on Amazon Cloud

- Introduction to Cloud Infrastructure
- Amazon SaaS, Paas and IaaS
- Creating EC2 Instance for Processing
- Creating S3 Buckets
- Deploying Data on to the Cloud
- Choosing size of our instance
- Configuration of EMR Instance
- Creating a virtual cluster on Amazon
- Deploying project and getting stats

HBase & Zookeeper

- Introduction to HBase
- HBase VS RDBMS
- HBase Components
- Hbase Architecture
- HBase Shell
- HBase Client API
- Data Loading Techniques
- Run Modes & Configuration
- HBase Cluster Deployment
- Regionservers and their implementation
- Client API's and their features
- How messaging system works
- Columns and column families
- Configuring hbase-site.xml
- Available Client
- Loading Hbase with semi-structured data
- Internal data storage in hbase
- Timestamps
- Creating table with column families
- MapReduce Integration.
- HBase: Advanced Usage, Schema Design
- Load data from pig to hbase
- Zookeeper Data Model
- Zookeeper Service
- Challenges faced in Distributed Applications
- Coordination
- Znode
- Client API Functions
- Bulk Loading
- Receiving and Inserting Data
- Filters in HBase
- Sqoop architecture
- Data Import and export in SQOOP
- Deploying quorum and configuration throughout the Cluster

Sqoop

- An Overview of Sqoop
- Sqoop Real-life Connect
- Sqoop and its Uses
- Advantages of Sqoop
- Sqoop Processing
- Sqoop Execution Process

- Importing Data Using Sqoop
- Sqoop Import Process
- How to Import data to Hive and HBase?
- How to Export Data from Hadoop using Sqoop?
- Sqoop Alternative
- Sqoop Connector

Flume

- Introduction to Flume
- Introduction to Apache Flume
- Flume Model
- Flume Goals
- Scalability in Flume
- Flume Data Integration
- Flume Installation on Single Node and Multinode Cluster
- Flume Architecture and various Components
- Data Sources: Types and Variants
- Data Target: Types and Variants
- Deploying an agent onto a single node cluster
- Problems associated with Flume
- Interview questions based on Flume

Oozie and Hue

- Introduction to Apache Oozie
- Oozie: Components
- Oozie: Workflow
- Scheduling with Oozie
- Hands-on Training on Oozie Workflow
- Oozie Coordinator
- Oozie Commands
- Oozie Web Console
- Oozie for MapReduce
- Hive in Oozie
- An Overview of Hue
- Hue in Real-time Scenarios
- Use Cases in Hue

MongoDB

- Understanding MongoDB
- NoSQL Databases
- JSON and BSON
- Vertical and Horizontal Scaling
- Data Types
- MongoDB Tools

- Collection and Database
- Schema Design and Modeling
- CRUD Operations in MongoDB
- Indexing and Aggregation
- Replication and Sharding
- MongoDB Cluster Operations

Yarn Architecture

- Introduction to YARN and MR2 daemons
- Active and Standby Namenodes
- Resource Manager and Application Master
- Node Manager
- Container Objects and Container
- Namenode Federation
- Cloudera Manager and Impala
- Load balancing in cluster with namenode federation
- Architectural differences between Hadoop 1.0 and 2.0

Basics of JAVA for Hadoop

- The Java Virtual Machine
- Variables
- Data types
- Constructs: Conditional and Looping
- Types: Wrapper classes
- Object-Oriented JAVA
- Fields and Methods
- Constructors
- Overloading methods
- Garbage collection
- Nested classes
- Overriding methods
- Polymorphism
- Making methods and classes final
- Abstract classes and methods
- Interfaces
- Threads
- Classes
- The I/O Package
- JAVA Security

Spark Ecosystem and BigData

- What and How of Distributed Systems

- What are New Generation Distributed Systems
- What is Big Data and its Limitations
- What are the Limitations of MapReduce in Hadoop
- Processing: Batch and Real-time Big Data Analytics
- Hadoop Ecosystem Overview
- Understanding Apache Spark
- Evolution and Features of Spark
- Spark Ecosystem
- Modes of Spark
- Overview of Spark on a cluster
- Spark Standalone Cluster
- Spark Web User Interface.
- Language Flexibility in Spark
- Architecture of Spark
- Spark and Big Data
- APIs in Spark
- Additional Benefits of Spark
- Various Tasks of Spark on a Cluster
- Apache Spark and Hadoop Ecosystem

Scala and Spark

- Defining Scala
- Features of Scala
- Scala and Spark interdependency
- How to use Scala in other Frameworks
- What is Scala REPL
- Various Scala Operations
- Basic Data Types in Scala
- Understanding Basic Literals
- What are Operators
- Various Types of Operators
- What is Arithmetic Operator
- How to use Logical Operator
- Control Structures in Scala
- Functions and Procedures
- Anonymous Functions
- Objects and Classes
- Collections in Scala: Mutable vs Immutable Collection
- Array
- Array Buffer
- Map and Maps Operations
- Pattern Matching
- Tuples
- Lists
- Streams

Programming Techniques in Scala

- What is a Class in Scala
- Understanding Getters and Setters
- What are Custom Getters and Setters
- General Properties of Getters
- What is an Auxiliary Constructor
- What is a Primary Constructor
- Defining Singletons
- What are Companion Objects
- How to extend a Class
- Overriding Methods
- Traits as Interfaces and Layered Traits

Spark Shell and PySpark

- What is Spark Shell?
- How to create a Spark Context?
- How to load a file in Shell?
- Operations on files in Spark Shell
- Understanding SBT
- How to build a Spark Project with SBT?
- How to run Spark Project with SBT?
- What is a Local Mode?
- Spark Mode
- Distributed Persistence
- Built-in Libraries for Spark
- The PySpark Shell
- The PySpark Shell – Advanced
- Spark Tools
- PySpark Integration with Jupyter Notebook
- Case Study: Analyzing Airlines Data with PySpark

Working with Key/Value Pairs

- Creating Pair RDDs
- Transformations on Pair RDDs
- Aggregations, Grouping Data, Joins, Sorting Data
- Data Partitioning
- Determining an RDDs Partitioner
- Operations that Benefit from Partitioning
- Operations that Affect Partitioning
- Loading and Saving Data

- File Formats: JSON, Comma-Separated and Tab-Separated Values
- File Formats: Sequence Files, Object Files
- Hadoop Input and Output Formats
- Filesystems: Local/Regular FS
- Amazon S3 and HDFS
- Databases: Java Database Connectivity
- Cassandra, HBase, ElasticSearch

RDD and Spark

- Defining RDDs
- Transformations in RDD
- Various Actions in RDD
- Lazy Evaluations
- Passing Functions to Spark: Python, Scala, Java
- How to load data in RDD?
- How to save data through RDD?
- Scala RDD Extensions 00:00
- What are Double RDD Methods?
- RDD Methods
- Java Pair RDD Methods
- General RDD Methods
- Java RDD Methods
- Common Java RDD Methods
- Spark Java Function Classes
- Understanding Key-Value Pair RDD in Scala and Java
- MapReduce and RDD
- Spark and Hadoop Integration
- HDFS and Yarn

Spark Streaming

- What is Spark Streaming?
- Architecture and Abstraction of Spark Streaming
- Streaming Word Count
- What is a Micro Batch?
- Understanding DStreams
- Input DStreams and Receivers
- Basic and Advanced Sources
- Input Sources: Core Sources and Cluster Sizing
- Transformations in Spark Streaming
- Stateless and Stateful Transformations
- Transformations on DStreams
- Spark Streaming and Fault Tolerance
- Driver Fault Tolerance

- Worker Fault Tolerance
- Receiver Fault Tolerance
- Enabling Checkpointing
- Socket Stream and File Stream
- Stateful Operations
- Window Operations and its Types
- Join Operations-Stream-Dataset Joins
- Join Operations-Stream-Stream Joins
- Parallelism Level

Machine Learning with MLlib

- What is Machine Learning?
- System Requirements
- Machine Learning with Spark
- Spam Classification
- Data Types: Understand and working with Vectors
- Algorithms: Statistics, Classification, and Regression
- Mllib Clustering
- Mllib Collaborative Filtering
- Dimensionality Reduction
- Model Evaluation
- Configuring Algorithms
- Caching RDDs to Reuse
- Recognizing Sparsity
- Level of Parallelism
- Pipeline API

Spark SQL and GraphX

- Initializing Spark SQL
- SchemaRDDs
- Caching
- Loading and Saving Data
- Apache Hive, Parquet, JSON
- JDBC/ODBC Server
- Working with Beeline
- Spark SQL UDFs
- Hive UDFs
- What is a Graph-Parallel System?
- What are the Limitations of Graph-Parallel System?
- What is GraphX?
- What is Property Graph?
- What are Graph Operators?
- List of Operators

- Property and Structural Operators
- Subgraphs
- Join Operators
- How to create a Graph using GraphX
- Understanding Hive and Spark SQL
- An Insight into Spark SQL and SQL Context
- Hive and Spark SQL Integration
- Hive Queries through Spark
- Various Testing Tips in Scala
- Shared Variables
- Broadcast Variables
- Accumulators

Real Life Project on Big Data

Live Hadoop project based on industry use-case using Hadoop components like Pig, HBase, MapReduce and Hive to solve real world problems in Big Data Analytics