Engineers 3D

IT & ENGINEERING SERVICES

TRAINING PROJECTS PLACEMEMNTS

Contents [hide]

Hadoop Course Overview

Hadoop Development course teaches the skill set required for the learners how to setup Hadoop Cluster, how to store Big Data using Hadoop (HDFS) and how to process/analyze the Big Data using Map-Reduce Programming or by using other Hadoop ecosystems. Attend Hadoop Training demo by Real-Time Expert.

Hadoop Training Course Prerequisites

- Basic **Unix** Commands
- **Core Java** (OOPS Concepts, Collections , Exceptions) for Map Reduce Programming
- **SQL** Query knowledge for Hive Queries

Hadoop Course System Requirements

- Any Linux flavor OS (Ex: Ubuntu/Cent OS/Fedora/RedHat Linux) with 4 GB RAM (minimum), 100 GB HDD

- Java 1.6+
- Open-SSH server & client
- MYSQL Database
- Eclipse IDE
- VMWare (To use Linux OS along with Windows OS)

Hadoop Training Course Duration

- 50 Hours, daily 1:30 Hours

Hadoop Course Content

Introduction to Hadoop

- High Availability
- Scaling
- Advantages and Challenges

Introduction to Big Data

- What is Big data
- Big Data opportunities,Challenges
- Characteristics of Big data

Introduction to Hadoop

- Hadoop Distributed File System
- Comparing Hadoop & SQL
- Industries using Hadoop
- Data Locality
- Hadoop Architecture
- Map Reduce & HDFS
- Using the Hadoop single node image (Clone)

Hadoop Distributed File System (HDFS)

- HDFS Design & Concepts
- Blocks, Name nodes and Data nodes
- HDFS High-Availability and HDFS Federation
- Hadoop DFS The Command-Line Interface
- Basic File System Operations
- Anatomy of File Read,File Write
- Block Placement Policy and Modes
- More detailed explanation about Configuration files
- Metadata, FS image, Edit log, Secondary Name Node and Safe Mode
- How to add New Data Node dynamically,decommission a Data Node dynamically (Without stopping cluster)
- FSCK Utility. (Block report)
- How to override default configuration at system level and Programming level
- HDFS Federation
- ZOOKEEPER Leader Election Algorithm
- Exercise and small use case on HDFS

Map Reduce

- Map Reduce Functional Programming Basics
- Map and Reduce Basics
- How Map Reduce Works
- Anatomy of a Map Reduce Job Run
- Legacy Architecture ->Job Submission, Job Initialization, Task Assignment, Task Execution, Progress and Status Updates
- Job Completion, Failures
- Shuffling and Sorting
- Splits, Record reader, Partition, Types of partitions & Combiner
- Optimization Techniques -> Speculative Execution, JVM Reuse and No. Slots
- Types of Schedulers and Counters
- Comparisons between Old and New API at code and Architecture Level
- Getting the data from RDBMS into HDFS using Custom data types
- Distributed Cache and Hadoop Streaming (Python, Ruby and R)
- YARN
- Sequential Files and Map Files
- Enabling Compression Codec’s
- Map side Join with distributed Cache
- Types of I/O Formats: Multiple outputs, NLINEinputformat
- Handling small files using CombineFileInputFormat

Map Reduce Programming – Java Programming

- Hands on “Word Count” in Map Reduce in standalone and Pseudo distribution Mode
- Sorting files using Hadoop Configuration API discussion
- Emulating “grep” for searching inside a file in Hadoop
- DBInput Format
- Job Dependency API discussion
- Input Format API discussion,Split API discussion
- Custom Data type creation in Hadoop

NOSQL

- ACID in RDBMS and BASE in NoSQL
- CAP Theorem and Types of Consistency
- Types of NoSQL Databases in detail
- Columnar Databases in Detail (HBASE and CASSANDRA)
- TTL, Bloom Filters and Compensation

HBase

- HBase Installation, Concepts
- HBase Data Model and Comparison between RDBMS and NOSQL
- Master & Region Servers
- HBase Operations (DDL and DML) through Shell and Programming and HBase Architecture
- Catalog Tables
- Block Cache and sharding
- SPLITS
- DATA Modeling (Sequential, Salted, Promoted and Random Keys)
- JAVA API’s and Rest Interface
- Client Side Buffering and Process 1 million records using Client side Buffering
- HBase Counters
- Enabling Replication and HBase RAW Scans
- HBase Filters

- Bulk Loading and Co processors (Endpoints and Observers with programs)
- Real world use case consisting of HDFS,MR and HBASE

Hive

- Hive Installation, Introduction and Architecture
- Hive Services, Hive Shell, Hive Server and Hive Web Interface (HWI)
- Meta store, Hive QL
- OLTP vs. OLAP
- Working with Tables
- Primitive data types and complex data types
- Working with Partitions
- User Defined Functions
- Hive Bucketed Tables and Sampling
- External partitioned tables, Map the data to the partition in the table, Writing the output of one query to another table, Multiple inserts
- Dynamic Partition
- Differences between ORDER BY, DISTRIBUTE BY and SORT BY
- Bucketing and Sorted Bucketing with Dynamic partition
- RC File
- INDEXES and VIEWS
- MAPSIDE JOINS
- Compression on hive tables and Migrating Hive tables
- Dynamic substation of Hive and Different ways of running Hive
- How to enable Update in HIVE
- Log Analysis on Hive
- Access HBASE tables using Hive
- Hands on Exercises

Pig

- Pig Installation
- Execution Types
- Grunt Shell
- Pig Latin
- Data Processing
- Schema on read
- Primitive data types and complex data types
- Tuple schema, BAG Schema and MAP Schema
- Loading and Storing
- Filtering, Grouping and Joining
- Debugging commands (Illustrate and Explain)
- Validations,Type casting in PIG
- Working with Functions
- User Defined Functions
- Types of JOINS in pig and Replicated Join in detail
- SPLITS and Multiquery execution
- Error Handling, FLATTEN and ORDER BY
- Parameter Substitution
- Nested For Each
- User Defined Functions, Dynamic Invokers and Macros
- How to access HBASE using PIG, Load and Write JSON DATA using PIG
- Piggy Bank
- Hands on Exercises

SQOOP

- Sqoop Installation
- **Import Data**.(Full table, Only Subset, Target Directory, protecting Password, file format other than CSV, Compressing, Control Parallelism, All tables Import)
- **Incremental Import**(Import only New data, Last Imported data, storing Password in Metastore, Sharing Metastore between Sqoop Clients)
- **Free Form Query Import**
- **Export data to RDBMS,HIVE and HBASE**
- Hands on Exercises

HCatalog

- HCatalog Installation
- Introduction to HCatalog
- About Hcatalog with PIG,HIVE and MR
- Hands on Exercises

Flume

- Flume Installation
- Introduction to Flume
- Flume Agents: Sources, Channels and Sinks
- Log User information using Java program in to HDFS using LOG4J and Avro Source, Tail Source
- Log User information using Java program in to HBASE using LOG4J and Avro Source, Tail Source
- Flume Commands
- Use case of Flume: Flume the data from twitter in to HDFS and HBASE. Do some analysis using HIVE and PIG

More Ecosystems

- HUE.(Hortonworks and Cloudera)

Oozie

- Workflow (Action, Start, Action, End, Kill, Join and Fork), Schedulers, Coordinators and Bundles.,to show how to schedule Sqoop Job, Hive, MR and PIG
- Real world Use case which will find the top websites used by users of certain ages and will be scheduled to run for every one hour
- Zoo Keeper
- HBASE Integration with HIVE and PIG
- Phoenix
- Proof of concept (POC)

SPARK

- Spark Overview
- Linking with Spark, Initializing Spark
- Using the Shell
- Resilient Distributed Datasets (RDDs)
- Parallelized Collections
- External Datasets
- RDD Operations
- Basics, Passing Functions to Spark
- Working with Key-Value Pairs
- Transformations
- Actions
- RDD Persistence

- Which Storage Level to Choose?
- Removing Data
- Shared Variables
- Broadcast Variables
- Accumulators
- Deploying to a Cluster
- Unit Testing
- Migrating from pre-1.0 Versions of Spark
- Where to Go from Here