HADOOP PROGRAMMING (5 Days)

This training course introduces the students to Apache Hadoop and key Hadoop ecosystem projects: Pig, Hive, Sqoop, Impala, Oozie, HBase, and Spark.

This intensive training course uses lectures and hands-on labs that help students learn theoretical knowledge and gain practical experience of Apache Hadoop and related Apache projects.

Audience

Business Analysts, IT Architects, Technical Managers and Developers

Prerequisites

Participants should have the general knowledge of programming in Java and SQL as well as experience working in Unix environments (e.g. running shell commands, etc.)

Course Outline

Chapter 1. MapReduce Overview

- The Client – Server Processing Pattern
- Distributed Computing Challenges
- MapReduce Defined
- Google's MapReduce
- The Map Phase of MapReduce
- The Reduce Phase of MapReduce
- MapReduce Explained
- MapReduce Word Count Job
- MapReduce Shared-Nothing Architecture
- Similarity with SQL Aggregation Operations
- Example of Map & Reduce Operations using JavaScript
- Problems Suitable for Solving with MapReduce
- Typical MapReduce Jobs
- Fault-tolerance of MapReduce
- Distributed Computing Economics
- MapReduce Systems

Chapter 2. Hadoop Overview

- Apache Hadoop
- Apache Hadoop Logo
- Typical Hadoop Applications
- Hadoop Clusters
- Hadoop Design Principles
- Hadoop's Core Components
- Hadoop Simple Definition
- High-Level Hadoop Architecture
- Hadoop-based Systems for Data Analysis
- Other Hadoop Ecosystem Projects
- Hadoop Caveats
- Hadoop Distributions
- Cloudera Distribution of Hadoop (CDH)
- Cloudera Distributions
- Hortonworks Data Platform (HDP)
- MapR

Chapter 3. Hadoop Distributed File System Overview

- Hadoop Distributed File System (HDFS)
- HDFS High Availability
- HDFS "Fine Print"
- Hadoop Security
- HDFS Rack-awareness
- Data Blocks
- Data Block Replication Example
- HDFS NameNode Directory Diagram
- Accessing HDFS
- Examples of HDFS Commands
- Client Interactions with HDFS for the Read Operation
- Read Operation Sequence Diagram
- Client Interactions with HDFS for the Write Operation
- Communication inside HDFS

Chapter 4. MapReduce with Hadoop

- Hadoop's MapReduce
- MapReduce v1
- JobTracker and TaskTracker (the Classic MapReduce)
- YARN (MapReduce v2)
- YARN vs MRv1
- YARN As Data Operating System
- MapReduce Programming Options
- Java MapReduce API
- The Structure of a Java MapReduce Program
- The Mapper Class
- The Reducer Class
- The Driver Class
- Compiling Classes
- Running the MapReduce Job
- The Structure of a Single MapReduce Program
- Combiner Pass (Optional)
- Hadoop's Streaming MapReduce
- Python Word Count Mapper Program Example
- Python Word Count Reducer Program Example
- Setting up Java Classpath for Streaming Support
- Streaming Use Cases
- The Streaming API vs Java MapReduce API
- Amazon Elastic MapReduce

Chapter 5. Apache Pig Scripting Platform

- What is Pig?
- Pig Latin
- Apache Pig Logo
- Pig Execution Modes
- Local Execution Mode
- MapReduce Execution Mode
- Running Pig
- Running Pig in Batch Mode
- What is Grunt?
- Pig Latin Statements
- Pig Programs
- Pig Latin Script Example
- SQL Equivalent
- Differences between Pig and SQL
- Statement Processing in Pig
- Comments in Pig
- Supported Simple Data Types
- Supported Complex Data Types
- Arrays
- Defining Relation's Schema
- The bytearray Generic Type
- Using Field Delimiters
- Referencing Fields in Relations

Chapter 6. Apache Pig HDFS Interface

- The HDFS Interface
- FSShell Commands (Short List)
- Grunt's Old File System Commands

Chapter 7. Apache Pig Relational and Eval Operators

- Pig Relational Operators
- Example of Using the JOIN Operator
- Example of Using the Order By Operator
- Caveats of Using Relational Operators
- Pig Eval Functions
- Caveats of Using Eval Functions (Operators)
- Example of Using Single-column Eval Operations
- Example of Using Eval Operators For Global Operations

Chapter 8. Apache Pig Miscellaneous Topics

- Utility Commands
- Handling Compression
- User-Defined Functions
- Filter UDF Skeleton Code

Chapter 9. Apache Pig Performance

- Apache Pig Performance
- Performance Enhancer - Use the Right Schema Type
- Performance Enhancer - Apply Data Filters
- Use the PARALLEL Clause
- Examples of the PARALLEL Clause
- Performance Enhancer - Limiting the Data Sets
- Displaying Execution Plan
- Compress the Results of Intermediate Jobs
- Example of Running Pig with LZO Compression Codec

Chapter 10. Apache Oozie

- What is Oozie?
- Apache Oozie Logo
- Oozie Terminology
- Directed Acyclic Graph
- Oozie Job Types
- Oozie Architecture
- Oozie Configuration
- Oozie Workflows
- The Flow Of Oozie Workflows
- More on Oozie Workflows
- Oozie Workflow Control Nodes
- More Oozie Workflow Control Nodes
- A Workflow Example
- A More Complex Workflow Example
- Oozie Coordinator
- The Pig Action Template
- The Pig Action Template

Chapter 11. Hive

- What is Hive?
- Apache Hive Logo
- Hive's Value Proposition
- Who uses Hive?
- Hive's Main Sub-Systems
- Hive Features
- The "Classic" Hive Architecture
- The New Hive Architecture
- HiveQL
- Where are the Hive Tables Located?
- "Legacy" Hive Command-line Interface (CLI)
- The Beeline Command Shell

Chapter 12. Hive Command-line Interface

- Hive Command-line Interface (CLI)
- The Hive Interactive Shell
- Running Host OS Commands from the Hive Shell
- Interfacing with HDFS from the Hive Shell
- The Hive in Unattended Mode
- The Hive CLI Integration with the OS Shell
- Executing HiveQL Scripts
- Comments in Hive Scripts
- Variables and Properties in Hive CLI
- Setting Properties in CLI
- Example of Setting Properties in CLI
- Hive Namespaces
- Using the SET Command
- Setting Properties in the Shell
- Setting Properties for the New Shell Session

Chapter 13. Hive Data Definition Language

- Hive Data Definition Language
- Creating Databases in Hive
- Using Databases
- Creating Tables in Hive
- Supported Data Type Categories
- Common Numeric Types
- String and Date / Time Types
- Miscellaneous Types
- Example of the CREATE TABLE Statement
- Working with Complex Types
- Table Partitioning
- Table Partitioning
- Table Partitioning on Multiple Columns
- Viewing Table Partitions
- Row Format
- Data Serializers / Deserializers
- File Format Storage
- More on File Formats
- The EXTERNAL DDL Parameter
- Example of Using EXTERNAL

- Creating an Empty Table
- Dropping a Table
- Table / Partition(s) Truncation
- Alter Table/Partition/Column
- Views
- Create View Statement
- Why Use Views?
- Restricting Amount of Viewable Data
- Examples of Restricting Amount of Viewable Data
- Creating and Dropping Indexes
- Describing Data

Chapter 14. Hive Data Manipulation Language

- Hive Data Manipulation Language (DML)
- Using the LOAD DATA statement
- Example of Loading Data into a Hive Table
- Loading Data with the INSERT Statement
- Appending and Replacing Data with the INSERT Statement
- Examples of Using the INSERT Statement
- Multi Table Inserts
- Multi Table Inserts Syntax
- Multi Table Inserts Example

Chapter 15. Hive Select Statement

- HiveQL
- The SELECT Statement Syntax
- The WHERE Clause
- Examples of the WHERE Statement
- Partition-based Queries
- Example of an Efficient SELECT Statement
- The DISTINCT Clause
- Supported Numeric Operators
- Built-in Mathematical Functions
- Built-in Aggregate Functions
- Built-in Statistical Functions
- Other Useful Built-in Functions
- The GROUP BY Clause
- The HAVING Clause
- The LIMIT Clause
- The ORDER BY Clause
- The JOIN Clause
- The CASE … Clause
- Example of CASE … Clause

Chapter 16. Apache Sqoop

- What is Sqoop?
- Apache Sqoop Logo
- Sqoop Import / Export
- Sqoop Help
- Examples of Using Sqoop Commands
- Data Import Example
- Fine-tuning Data Import
- Controlling the Number of Import Processes
- Data Splitting
- Helping Sqoop Out
- Example of Executing Sqoop Load in Parallel
- A Word of Caution: Avoid Complex Free-Form Queries
- Using Direct Export from Databases
- Example of Using Direct Export from MySQL
- More on Direct Mode Import
- Changing Data Types
- Example of Default Types Overriding
- File Formats
- The Apache Avro Serialization System
- Binary vs Text
- More on the SequenceFile Binary Format
- Generating the Java Table Record Source Code
- Data Export from HDFS
- Export Tool Common Arguments
- Data Export Control Arguments
- Data Export Example
- Using a Staging Table
- INSERT and UPDATE Statements
- INSERT Operations
- UPDATE Operations
- Example of the Update Operation
- Failed Exports
- Sqoop2
- Sqoop2 Architecture

Chapter 17. Cloudera Impala

- What is Cloudera Impala?
- Impala's Logo
- Impala Architecture
- Benefits of Using Impala
- Key Features
- How Impala Handles SQL Queries
- Impala Programming Interfaces
- Impala SQL Language Reference
- Differences Between Impala and HiveQL
- Impala Shell
- Impala Shell Main Options
- Impala Shell Commands
- Impala Common Shell Commands
- Cloudera Web Admin UI
- Impala Browse-based Query Editor

Chapter 18. Apache HBase

- What is HBase?
- HBase Design
- HBase Features
- HBase High Availability
- The Write-Ahead Log (WAL) and MemStore
- HBase vs RDBS
- HBase vs Apache Cassandra
- Not Good Use Cases for HBase
- Interfacing with HBase
- HBase Thrift And REST Gateway
- HBase Table Design
- Column Families
- A Cell's Value Versioning
- Timestamps
- Accessing Cells
- HBase Table Design Digest
- Table Horizontal Partitioning with Regions
- HBase Compaction
- Loading Data in HBase
- Column Families Notes
- Rowkey Notes
- HBase Shell
- HBase Shell Command Groups
- Creating and Populating a Table in HBase Shell
- Getting a Cell's Value
- Counting Rows in an HBase Table

Chapter 19. Apache HBase Java API

- HBase Java Client
- HBase Scanners
- Using ResultScanner Efficiently
- The Scan Class
- The KeyValue Class
- The Result Class
- Getting Versions of Cell Values Example
- The Cell Interface
- HBase Java Client Example
- Scanning the Table Rows
- Dropping a Table
- The Bytes Utility Class

Chapter 20. Introduction to Functional Programming

- What is Functional Programming (FP)?
- Terminology: First-Class and Higher-Order Functions
- Terminology: Lambda vs Closure
- A Short List of Languages that Support FP
- FP with Java
- FP With JavaScript
- Imperative Programming in JavaScript
- The JavaScript map (FP) Example
- The JavaScript reduce (FP) Example
- Using reduce to Flatten an Array of Arrays (FP) Example
- The JavaScript filter (FP) Example

Chapter 21. Introduction to Apache Spark

- What is Spark
- A Short History of Spark
- Where to Get Spark?
- The Spark Platform
- Spark Logo
- Common Spark Use Cases
- Languages Supported by Spark
- Running Spark on a Cluster
- The Driver Process
- The Executor and Worker Processes
- The Spark Application Architecture
- Interfaces with Data Storage Systems
- Limitations of Hadoop's MapReduce

- Spark vs MapReduce
- Spark as an Alternative to Apache Tez
- The Resilient Distributed Dataset (RDD)
- Spark Streaming (Micro-batching)
- Spark SQL
- Example of Spark SQL
- Spark Machine Learning Library
- GraphX

Chapter 22. The Spark Shell

- The Spark Shell
- The Spark Shell UI
- The Spark Context (sc) and SQL Context (sqlContext)
- The Shell Spark Context
- Loading Files
- Saving Files
- Basic Spark ETL Operations

Chapter 23. Spark RDDs

- The Resilient Distributed Dataset (RDD)
- Ways to Create an RDD
- Custom RDDs
- Supported Data Types
- RDD Operations
- RDDs are Immutable
- Spark Actions
- RDD Transformations
- Other RDD Operations
- Chaining RDD Operations
- RDD Lineage
- The Big Picture
- Parallelized Collections
- More on parallelize() Method
- The Pair RDD
- Where do I use Pair RDDs?
- Example of Creating a Pair RDD with Map
- Example of Creating a Pair RDD with keyBy
- Miscellaneous Pair RDD Operations
- RDD Caching
- RDD Persistence
- The Tachyon Storage

Chapter 24. Parallel Data Processing with Spark

- Running Spark on a Cluster
- Spark Stand-alone Option
- The High-Level Execution Flow in Stand-alone Spark Cluster
- Data Partitioning
- Data Partitioning Diagram
- Single Local File System RDD Partitioning
- Multiple File RDD Partitioning
- Special Cases for Small-sized Files
- Parallel Data Processing of Partitions
- Spark Application, Jobs, and Tasks
- Stages and Shuffles
- The "Big Picture"

Chapter 25. Introduction to Spark SQL

- What is Spark SQL?
- Uniform Data Access with Spark SQL
- Hive Integration
- Hive Interface
- Integration with BI Tools
- Spark SQL is No Longer Experimental Developer API!
- What is a DataFrame?
- The SQLContext Object
- The SQLContext API
- Changes Between Spark SQL 1.3 to 1.4
- Example of Spark SQL (Scala Example)
- Example of Working with a JSON File
- Example of Working with a Parquet File
- Using JDBC Sources
- JDBC Connection Example
- Performance & Scalability of Spark SQL

Lab Exercises

Lab 1. Learning the Lab Environment
Lab 2. The Hadoop Distributed File System
Lab 3. Hadoop Streaming MapReduce
Lab 4. Programming Java MapReduce Jobs on Hadoop
Lab 5. Getting Started with Apache Pig
Lab 6. Apache Pig HDFS Command-Line Interface
Lab 7. Working with Data Sets in Apache Pig
Lab 8. Using Relational Operators in Apache Pig
Lab 9. Apache Oozie
Lab 10. The Hive Shell
Lab 11. Hive Data Definition Language
Lab 12. Using Select Statement in HiveQL
Lab 13. Table Partitioning in Hive
Lab 14. Data Import and Export with Sqoop
Lab 15. Using Impala
Lab 16. Using HBase Shell
Lab 17. Elements of Functional Programming with JavaScript
Lab 18. The Spark Shell
Lab 19. Spark ETL and HDFS Interface
Lab 20. Common Map / Reduce Programs in Spark
Lab 21. Spark SQL