

Course code: **HADOOP**

Course title: **Designing Big Data solutions using Apache Hadoop & Family**

Days: 5

## Description:

### Course intended for:

The training is aimed at developers, architects and application administrators who want to create or maintain systems based on scalable architectures such as Big Data, in particular specialists for whom performance and volume of processed data is the highest priority. The course is aimed at people currently involved in creation of relational databases who want to gain knowledge of alternative technologies, which gradually displace relational databases from different application areas. The training is also dedicated for people who want to supplement their knowledge of concepts of Big Data, MapReduce, NoSQL and their implementation using Apache Hadoop & Family software.

### Course objective:

Course participants will gain knowledge of cross-cutting concepts such as MapReduce algorithm, Big Data, BigTable, DFS distributed file systems, NoSQL databases. After the training, course participants will be able to choose the right techniques for their projects. In addition to general introduction to Big Data, this training is focusing on a whole Apache Hadoop stack.

### Course strengths:

Training program includes a general introduction to Big Data as well as the overall presentation of the Apache Hadoop. The training is unique because its subject is not fully covered in the literature and knowledge of Big Data and NoSQL is highly fragmented. Training program is constantly updated due to a rapid development of Big Data solutions.

### Requirements:

Participants are required to have basic knowledge of databases and basic programming skills in Java.

### Course parameters:

5 \* 7 hours of lectures and workshops at a ratio of 1:3. During the workshops, in addition to simple exercises, participants will solve problems by implementing its own data processing

algorithms using the MapReduce paradigm, model NoSQL data structures and perform basic administrative tasks. Group size: max. 8-10 people.

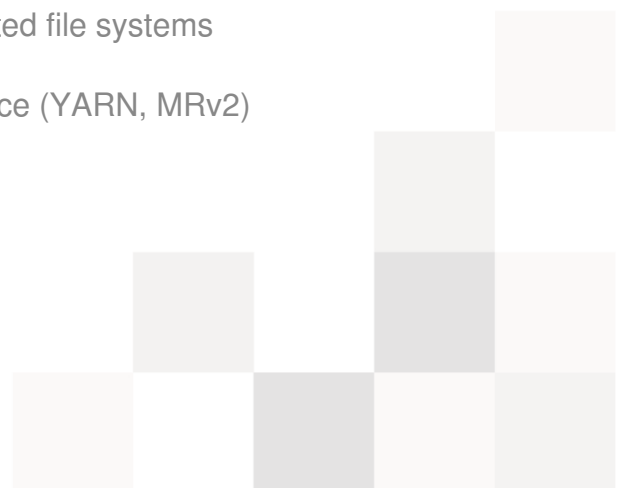
## Course curriculum:

### 1. Introduction

- I. Big Data, BigTable, BigQuery, MapReduce
- II. MapReduce in details
- III. MapReduce compared to other distributed processing techniques such as MPI, PVM etc.
- IV. Apache Hadoop & Family

### 2. Apache Hadoop

- I. Architecture
- II. Hadoop 1.0 vs 2.0
- III. Hadoop Shell Commands
- IV. Apache Hadoop Distributed File System (HDFS)
  - i. Architecture, NameNodes, DataNodes
  - ii. Federation and clustering
  - iii. File attributes
  - iv. Snapshots
  - v. WebHDFS, HttpFS, FUSE
  - vi. Comparison to other distributed file systems
- V. Apache Hadoop NextGen MapReduce (YARN, MRv2)
  - i. Architecture
    - A. ResourceManager



- B. Scheduler
- C. ApplicationsManager
- D. JobTracker i TaskTracker

- ii. YARN shell
- iii. Hadoop/YARN API
- iv. YARN REST API
- v. MapReduce 1.0 vs MapReduce 2.0, API compatibility
- vi. Examples

## VI. Apache Hadoop administration

- i. Installation and configuration
- ii. Demons, configuration files, log files
- iii. Hadoop On Demand, Hadoop Cluster Setup
- iv. HDFS administration
  - A. Files attributed
  - B. Quota
- v. MaReduce administration
  - A. Jobs management
  - B. Scheduling
- vi. Cluster rebalancing
- vii. Monitoring
- viii. Administration tools

## 3. Apache PIG

### I. Introduction



- i. Architecture
- ii. Work modes
- iii. PigLatin
- iv. Hadoop/YARN API and PigLatin

## II. PigLatin in details

- i. Syntax
- ii. Datatypes
- iii. Operators
- iv. Built-in and user defined functions

## III. Built-in functions

- i. Simple (eval functions)
- ii. Data management
- iii. Mathematical
- iv. Strings
- v. Date time
- vi. Other

## IV. User defined functions (UDF)

- i. UDF in Java
- ii. UDF in JavaScript
- iii. UDF in Python/Jython/Groovy
- iv. Piggybank

## V. Efficiency

- i. Combiner



- ii. Multi-Query Execution

- iii. Optimization rules

- iv. Good practices

## VI. Testing and troubleshooting

- i. Diagnostics

- ii. Statistics

- iii. PigUnit

## 4. Apache HBase

### I. Introduction

- i. Introduction to NoSQL databases

- ii. The cause of the cloud databases increasing popularity

- iii. Consistency, availability, resistance to partitioning

- iv. CAP theorem

- v. What distinguishes NoSQL database from relational databases

- vi. Basic parameters of NoSQL databases

- vii. Classification and overview of NoSQL databases (Cassandra, HBase, Mongo, Riak, CouchDB, Tokyo Cabinet, Voldemort, etc.)

- viii. The problem of transactions and replication of NoSQL databases, including MongoDB

- ix. Unique features of HBase

### II. HBase architecture

- i. Catalogs

- ii. Master Servers

- iii. Regions and Region Servers



## III. Data model

- i. Conceptual and physical
- ii. Namespaces
- iii. Table
- iv. Row
- v. Column
- vi. Version
- vii. Cell

## IV. HBase

- i. HBase API
- ii. HBase in Apache Hadoop and MapReduce jobs
- iii. REST API, Apache Thrift etc.

## V. Performance

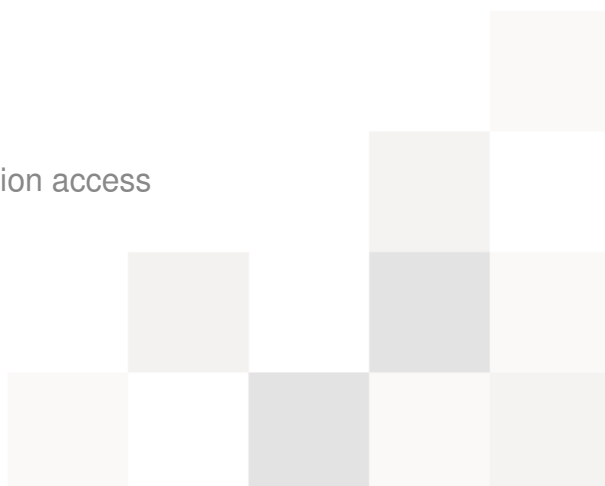
- i. Read optimization
- ii. Write optimization
- iii. JVM/OS/DFS tuning
- iv. Good practices

## VI. Troubleshooting

- i. Log files
- ii. Tools

## VII. Security

- i. Authentication and authorization access
- ii. Data security



## VIII. HBase administration

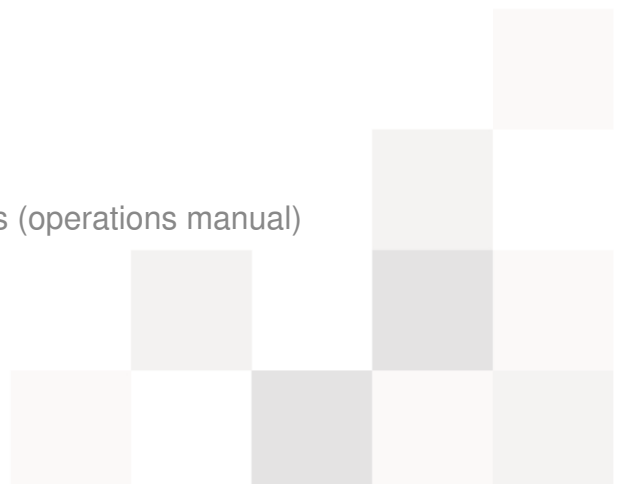
- i. Installation and configuration
- ii. Frequent administration tasks (operations manual)
- iii. Upgrade, migration, backup, data snapshots
- iv. Adding/Removing nodes from/to replica/cluster, nodes resynchronization
- v. Administration panels and tools

## IX. Apache HBase versus other NoSQL databases

- i. Apache Accumulo
- ii. Apache Cassandra

## 5. Apache Hive

- I. Architecture
- II. Hive features
- III. HiveCLI
- IV. HiveQL
- V. PigLatin vs HiveQL
- VI. Tables in Hive
- VII. Hive administration
  - i. Installation and configuration
    - A. Hive Metastore
    - B. HCatalog
    - C. WebHCat
  - ii. Frequent administration tasks (operations manual)
  - iii. Upgrade



## iv. Administration panels and tools

### 6. Apache Avro

#### I. Apache Avro IDL

#### II. Datatypes

#### III. Serialization

#### IV. Avro RPC

### 7. Apache Mahout

#### I. Machine learning, data mining

#### II. Mahout

##### i. Classification algorithms

##### ii. Grouping algorithms

##### iii. Evolutionary and genetic algorithms

##### iv. Reducing number of dimensions

##### v. Others

#### III. Installation and configuration

#### IV. Apache Mahout and Apache Hadoop

#### V. Examples

### 8. Data oriented applications

#### I. Apache Oozie

##### i. MapReduce

##### ii. Pig

##### iii. Hive

##### iv. Subworkflow





## II. Cascading

### 9. Management of Apache Hadoop & Family

#### I. Apache ZooKeeper

#### II. Apache Flume

#### III. Apache Ambari

### 10. Others

#### I. Apache Storm

#### II. Apache Spark

#### III. Cascalog

