Table des matières

[I Introduction to Big Data 1](#_Toc177313170)

[1 Course introduction 1](#_Toc177313171)

[2 What is Big Data? 1](#_Toc177313172)

[Impact of Big data 5](#_Toc177313173)

[Parallel Processing, Scaling, and Data Parallelism 6](#_Toc177313174)

[Big Data tools & Ecosystem 8](#_Toc177313175)

[Open source and Big Data 9](#_Toc177313176)

[Beyound the Hype 11](#_Toc177313177)

[Big Data Use Cases 12](#_Toc177313178)

[Glossary: 13](#_Toc177313179)

[Moduel 2 Introduction to Hadoop ecosystem: 15](#_Toc177313180)

[Introduction to Hadoop 15](#_Toc177313181)

[Introduction to MapReduce 17](#_Toc177313182)

[HDFS 23](#_Toc177313183)

# I Introduction to Big Data

## 1 Course introduction

This course is for anyone purchasing a data engineering, data architect , data scientist and ML engineer role.

At the end of this course, you will be able to identify Big Data and explain its characteristics, explain how the Apache Hadoop ecosystem fits into the Big Data stack and to leverage APACHE Spark to gain insights from Big Data.

## 2 What is Big Data?

**Big Data definition**

“The basic idea behind the phrase ‘Big Data’ is that everything we do increasingly leaving a digital trace (or data), which we can use and analyse to become smarter. The driving forces in this brave new world are access to ever-increasing volumes of data and our ever-increasing technological capability to mine that data for commercial insights.” – Bernard Marr

**Big Data vs Small data**

Small Data:

.Small enough for human inference

.Accumulated slowly

.Relatively consistent and structured data usually stored in known forms such as Json and XML

.Mostly located in storage systems within Enterprises or data centers

Big Data:

.Data generated in huge volumes and could be structured, semi-structured or unstructured

.Needs processing to generate insights for human consumption

.Arrives continuously at enormous speed from multiple sources

.Comprise any form of data including video, photo, and more

.Distributed on the cloud or on server farms

**Big Data life cycle:**

In reality Big Data is the entire life cycle of working with large volumes of data.

Phase 1: Business case

Big Data collection is initiated as a result of a business problem or requirement.

Phase 2: Data collection

As data is collected, It gets stored using a framework for distributed storage such as **Hadoop HDFS**

Phase 3: Data Modelling

To make sense of all the data collected, Map and Reduce tasks and scripts create a data model to store it in a database.

This data model includes the various data entities(or objects, and the relationship and rules between those entities. **Hadoop YARN, Hadoop Map Reduce**.

Phase 4: Data Processing

After modelling, data is ready to be processed.

Tools such as **Apache Spark and HIVE** are used to produce meaningful information from the modelled data.

Phase 5: Data visualization

Finally, the processed data is visualized for BI.

From there meaningful decisions are taken creating a continuous cycle of Big Data.

**Characteristics of Big Data:**

“Big Data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight,

Decision making, and process automation.” -Gartner

**The four V’s of Big Data**

1. Velocity:
2. Description

. Data that is generated fast

. Process that never stops

1. Attributes

. Batch

.Close to real time

.Streaming

1. Drivers

.Improved connectivity and hardware

.Rapid response times

.upscaling of precalculated analysis

1. Volume:
2. Description

.Scale of data

.Increased amount of stored data

1. Attributes

.Petabytes

.Exabytes

.Zetabyte..to name just a few

c)Drivers:

. Increase in data sources

. Higher resolution sensors

1. Variety
2. Description

. Data that comes from machines, people, and processes

.Structured, semi-structured, and unstructured data

1. Attributes

.Structure, Complexity, and origin

1. Drivers

.Mobile technologies

.Scalable infrastructure

.Resilience

.Fault recovery

.Efficient storage and retrieval

1. Veracity
2. Description

.Quality, origin, and conformity of facts

.Accuracy of data

.Data comes from people and processes

1. Attributes

.Consistency and completeness

.Integrity

.Ambiguity

c)Drivers

.cost and tracability

.Robust ingestion

.ETL mechanisms

The fifth V of Big Data is value, it is the outcome of making intelligent decision from leveraging the previous 4 V’s.

Produce value in the form of faster and smarter business decisions increase efficient use of resources, and discover new business opportunities.

Big data supports innovation and thus produces value

## Impact of Big data

Generating and using Big Data.

.Recommendation algorithms are based on big data

.Virtual personal assistance too

.Google Now makes recommendations before the user asks for them

.Big Data forecasts future needs and behavior

.Powerful machine learning algorithms drive business decisions that increase efficiency

.Data scientists and big data engineers bring this value to companies

Iot – Internet of things

An internet-enabled connected network of smart devices such as sensors, processors, embedded devices and communication hardware

Data collected, analysed, and acted upon for benefits such as improving customer experience, enhanced productivity, and increased revenue.

Major components of IoT:

Thing or device

--Wireless network—

Gateway

--Wireless network—

Cloud

--Wireless network—

Analytics

--Wireless network—

User Interface for decision-making

## Parallel Processing, Scaling, and Data Parallelism

Why parallel processing?

**Linear processing** is the resolution of a problem through a linear way:

Problem -> Instruction 1 -> Instruction 2 -> Instruction N -> Output

If an error occurs during this process, it will start all over again

**Parallel processing**

Problem

Instruction 1 Instuction 2 Instuction 3

Output

If an error occurs in one of the instructions, it will be fixed locally thus not restarting the entire program.

Advantages are that it reduces large data sets in fraction of time.

Also, less memory and compute requirements need as set of instruction are distributed to smaller execution nodes.

Finally this system is more flexible execution nodes being addable and removable in function of the needs

**What is data scaling?**

Technic to manage store and process the flow of data.

A single execution node can be upgraded vertically up but it is not efficient, it is called scall up.

A better strategy is to scale out (horizontal scaling)

Compute clusters(multiple execution nodes) are able to solve problems called “**embarrassingly parallel**”

If any execution process fails, it has no impact on the others and can simply rerun.

The “**not so easy**” parallel problem is the need for a computer cluster to see each of its execution nodes coordinate to solve it

**Data locality:**

When data is present in the same storage unit part of an execution node compute part, it is processed in there.

In the Hadoop ecosystem, bringing compute to the data when possible is a central idea in the design of the cluster. The output will also be brought on the same node.

**Fault tolerance:**

Is the ability of a computer to remain performant without interruption when one or multiple of its component fail.

This work for Hadoop primary data storage system HDFS and other storage systems.

Let’s say that there is 3 partitions in a storage execution node N1:

P1 P2 and P3, those partitions are distributed all along the computer cluster in order to always keep data available.

When N1 fails, a new execution node be created then partitions are copied and transferred to it.

Therefore, Hadoop ecosystem is reliable on 99.9999%

## Big Data tools & Ecosystem

Categories of Big Data tooling/Big Data Ecosystem:

1: Data technologies

.Capture, process and share data at any scale and in any format

.Work with structured and unstructured data

.Leverage high-performance, parallel Big Data processing

Key technologies:

Include Hadoop, HDFS, Spark, Cloudera & Databricks

2: Analytics & visualisation

.Big Data analytics examines large amounts of data

.Analyzed data is visualised

Popularanalytics tools available are Tableau,

Palentir, SAS, Pentatho, and Teradata

3: Business Intelligence

Business intelligence offers a range of tools that provide a quick and easy way to transform data into actionable insights

.Such insights inform organisation’s strategic and tactical business decisions

Exempes include Cognos, Oracle, PowerBi, Business Objects, and Hyperion

4: Cloud Provididers

Cloud providers offer fundamental infrastructure and support with shared ressources including computing, storage, networking and analytical software

Exemples are AWS, IBM GCP and Oracle

5: NoSql Databases

NoSql databases are best suited for Big Data processing

.Store and process vast amounts of data at scale

.Store information in JSON documents instead of relational tables

.NoSQL databases types include pure document databases, key-value stores , wide column databases, and graph databases

Exemples include MongoDB, CouchDB, Cassandra, Redis

6: Programming tools

.Perform large-scale analytical tasks and operationalize Big Data

.Provide all necessary functions for the Big Data life cycle

R, Python, Scala and Julia are common programming tools

## Open source and Big Data

4 types of open source:

.Public domain

.Copyleft

.Permissive

.Lesser General Public Liicense

See Jim Whitehurst quotes on open source

The community garden of code

Committer: ability to submit the code directly

Contributor: must pass by committer to see their code submitted

User

User Group

Hadoop ecosystem is the main open source Big Data project:

.MapReduce

Framework that allow code to run at scale on a Hadoop cluster.

It is still used but not as much as more modern data computation framework Apache Spark

.File System (HDFS)

Store and manages datasets, 70% of world’s data is from there

.Ressource manager YARN

Default resource for many data applications as Hive and Spark

More modern container-based manager like Kubernets increase its popularity

All big data applications available are built on Hadoop.

Some systems integrate tightly with the Hadoop ecosystem:

Hive,

Apache Spark,

Apache Hbase, A Large NoSql data store, often reside on the same cluster of an Hadoop ecosystem.

HDP, Hortonworks Data Plateform provide a set of data tools that are already configured to work together

# Beyound the Hype

Sources of Big Data are :

Social data

Machine data

Transactional data

Types of big data are structured,

Exemples are names , dates, credit card number

Some source of it are relational data base , Spreadsheets

semi-structured

metadata, emails

Some sources are XLM and Json data

and unstructured

80% of data in the world

Includes images, audio or video files

.Emergence of cloud computing have contributed to increasing potential of Big Data.

.It allows users to access highly scalable computing and storage resources through the Internet

.Organisations can expand server capacity to process Big Data in function of needs

## Big Data Use Cases

Industries leveraging Big Data:

Retail, finance, Manufacturing, Automobile, Telecom, Insurance

.Data driven companies have a competitive edge of peers

.Leverage data insights to improve decision making, enter new markets, and enhance customer experiences.

Leading users are Financial services, Technology and Telecomunications

**Big data in retail:**

Price analytics

Understand market segmentation, identify best price points, and perform margin analysis

Sentiment Analysis:

.Leverage social media to gauge customer perception on a product and devise an effective marketing strategy

**In insurance:**

Fraud analytics

.Spot fraudulent claims and detect anomalies

Risk Assessment

.Use predictive modelling to identify high-risk customer

**Telecoms:**

.Improved network security

.Marketing

.Real time network analytics

.Optemized pricing packages

**Manufacturing:**

Predictive Maintenance:

.Analyse equipment usage patterns and predict equipment failure,

Maintenance requirements, and parts replacements

Production optimization

.Understand the production lines,

Analyze increased production time to recommend optimizations

**In aumotive industry**

.Perdicts malfunctions, pre-emptive ordering of parts and repair suggestions

.Self driven cars

**Finance**

Fraud detection

Targeted service to their customers

Algorithmic trading

## Glossary:

**Apache Spark:**

An open-source, in-memory application framework used for distributed data processing and iterative analysis of large data sets.

**Apache HBase:**

A robust NoSQL datastore that efficiently manages storage and computation resources independently of the Hadoop ecosystem.

**Business intelligence (BI):**

Encompasses various tools and methodologies designed to convert data into actionable insights efficiently.

**Big data:**

|  |  |
| --- | --- |
|  | Data sets whose volume, velocity, or variety exceeds the capacity of conventional relational databases to effectively manage, capture, and process with minimal latency. Key characteristics of big data include substantial volume, high velocity, and diverse variety. |

**Big data analytics**

Uses advanced analytic techniques against large, diverse big data sets that include structured, semi-structured, and unstructured data from different sources and sizes, from terabytes to zettabytes. It helps companies gain insights from the data collected by IoT devices.

**Big data programming tools**

Programming tools are the final component of big data commercial tools. These programming tools perform large-scale analytical tasks and operationalize big data. They also provide all necessary functions for data collection, cleaning, exploration, modeling, and visualization. Some popular tools you can use for programming include R, Python, SQL, Scala, and Julia.

**Committer**

Most open-source projects have formal processes for contributing code and include various levels of influence and obligation to the project: Committer, contributor, user, and user group. Typically, committers can modify the code directly.

**Cloud computing**

Allows customers to access infrastructure and applications over the internet without needing on-premises installation and maintenance. By leveraging cloud computing, companies can utilize server capacity on-demand and rapidly scale up to handle the extensive computational requirements of processing large data sets and executing complex mathematical models.

**Cloud providers**

Offer essential infrastructure and support, providing shared computing resources encompassing computing power, storage, networking, and analytical software. These providers also offer software as a service model featuring specific solutions, enabling enterprises to gather, process, and visualize data efficiently. Prominent examples of cloud service providers include AWS, IBM, GCP, and Oracle.

**Extract, transform, and load (ETL) process**

A systematic approach that involves extracting data from various sources, transforming it to meet specific requirements, and loading it into a data warehouse or another centralized data repository.

**Hadoop**

An open-source software framework that provides dependable distributed processing for large data sets through the utilization of simplified programming models.

**Hadoop Distributed File System (HDFS)**

A file system distributed on multiple file servers, allowing programmers to access or store files from any network or computer. It is the storage layer of Hadoop. It works by splitting the files into blocks, creating replicas of the blocks, and storing them on different machines. It is built to access streaming data seamlessly. It uses a command-line interface to interact with Hadoop.

# Moduel 2 Introduction to Hadoop ecosystem:

## Introduction to Hadoop

Hadoop is an open-source framework used to process enormous data sets.

Hadoop is a set of open source programs for management of the data

It allows to run applications on clusters

Hadoop isn’t a database but an ecosystem that handles parallel jobs or processes

Hodoop is optamized to handle structured semi-structured and unstructured data

A little bit of history:

1999 Apache software foundation

2002 Nutch web search engine created

Parallel computation

2006 Nutch was divided into Web crawler, distributed systems

The distributed processing was called Hadoop

2008 Yahoo released Hadoop as an open-source project

How does Hadoop work?

THE CORE COMPONENTS OF Hadoop include:

Hadoop Common:

It’s an essential part f Apache Hadoop Framework that refers to the collection of common utilities and libraries that support other Hadoop Modules.

HDFS(Hadoop distributed file systems)is the storage component of Hadoop

It handles large datasets running on commodity hardwares.

A commodity hardware is low specifications industry-grade hardware and scales a single Hadoop cluster to hundreds or even thousands.

The next component is Map Reduce which is a processing unit of Hadoop and an important core component to the Hadoop framework

MapReduce processes data by splitting large data amounts of data into smaller units and processes them simultaneously

For a while MapReduce was the only way to access the data stored in the HDFS there I snow other ways to do so like Hive and Pig.

The last component is YARN which is short for “Yet Another Resource Negotiator”

It prepares RAM and CPU for Hadoop to run data in batch, stream, interactive and graph processing which are stored in HDFS.

The challenges of Hadoop

Hadoop is not good:

for processing transactions due to his lack of random access

When work cannot be parallelized

When there are dependencies within the data (dependencies arrive when record one must be processed before record 2)

For low latency data access (like trading online trading or voice IP)

For processing lot of small files (although there is work done in this area such IBM’s Adaptive MapReduce)

For intensive calculation with little data

For dealing with shortcoming of Hadoop new tools like Hive have been built on top of Hadoop

Hive provided SQL-like query and provide users with strong statistical functions.

Pig was popular for its multi query approach to cut down the number of time the data is scanned

## Introduction to MapReduce

MapReduce is a programming model used in Hadoop for processing Big Data

It enables massive scalability across hundreds or thousands.

As the processing component MapReduce is the heart of Apache Hadoop

It is a processing technique for distributed computing. It is based on Java.

Distributed computing is a system or machine with multiple components located on different machines. Each component has its own job but the components communicate to each other

To run on one system to the end user. The MapReduce algorithm consist of two important tasks

Map and Reduce (many MapReduce programs are written in Java but can also be devlopped in C++, Python, Ruby, R and so on).

As the name suggest, the MapReduce framework

Contains two tasks, Map and Reduce:

Map takes in an input file and performs some mapping tasks by processing and extracting important data information and to key value pairs and this are the preliminary output list.

Some more reorganisation goes on before the preliminary output is sent to the Reducer.

The Reducer works with multiple map functions and aggregates the pairs using the keys.

MapReduce keep track of its task by creating a unique key

How MapReduce works

First we have the Map step which takes a set of data and converts it into another set

Where individual elements are broken down into key/value pairs elements.

The key is the name and the value is the content.

The input data is a file that is saved in the Hadoop file system which is called HDFS.

Haw Map Reduce work:

Now let’s assume that we have an input file that contains names of people and we would like

To do a work count on the unique names occurrences

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Why use MapReduce?

The advantages of MapReduce is its ability to allow a high level of parallel jobs across multiple nodes.

A node is an independent computer used for processing ad storing big volumes of data.

In Hadoop we have two types nodes. The name node and the data node.

MapReduce allows for splitting and running independent tasks in parallel by dividing

Each task which in turn saves time.

MapReduce is very flexible and can process data that come in tabular and non-tabular forms.

Therefore, MapReduce provides business value to organisations regardless of how their data is structured.

It also provides support for different languages and provides a platform analysis, data and more. MapReduce has a couple of use cases and here we have some of them displayed.

Common use cases:

Social media :

Social media plateforms can use MapReduce to analyze who visited your profile and who viewed your posts.

Recommmandations:

Create a recommender’s system for users and provide suggestions for them based on their interest.

Financial industries:

Can be used for fraud detection by analysing behaviours of buyers and tracking down anomalies

Advertisement industry:

Can be used to analyse and understand the interaction with ads and the engagement levels

Hadoop Ecosystem:

HDFS: Hadoop Distributed File System

Store the data collected from ingestion and distributes the data across multiple nodes.

MapReduce:

Is used for making Big Data manageable by processing them in clusters

YARN: Yet Another Resource Negociator

Is the resource manager across clusters

The Hadoop ecosystem is made up of components that support one another

The extended ecosystem is made of components that support one another

We can examine the Hadoop ecosystem based on the various stages:

Ingest Data:

When the data is received from multiple sources

Flume & Sqoop are responsible for ingesting the data

and transferring them to the storage component, HDFS and HBase.

Store Data:

Examples:

HDFS & HBase

Then the data is distributed to a MapReduce framework like Pig & Hive to process and analyse the data, and the processing is done by parallel computing.

After all that is done, tools loke Hue are used to access the refined data

**Ingesting is the first step of Big Data:**

When you get data from different sources, you may use tools like Flume and Sqoop:

**Flume** is a distributed service that collects aggregates and transfers Big Data to the storage system.

It uses a simple extensible data model that allows for online analytic application

**Sqoop** is an open-source product designed to transfer bulk data between data base relational

Systems and Hadoop.

Sqoop looks in the relational database systems and summarizes the schema.

It then generates MapReduce code to import and export data as needed.

Sqoop allows you to quickly develop any other MapReduce application that use the records that Sqoop stores into HDFS

**Store Data**

**HBase** is a non-relational database that runs on the top of HDFS

It provides real time wrangling access to the Hadoop file system

Hbase uses tables to store data in indexes and allow for random access of data which makes lookups faster

**Cassandra** is a scalable NoSQL data base designed to have no single point of failure

**Analyze data**

**Pig** is used to analyse large amounts of data

Operates on the client side of cluster

A procedural data flow language that follows an order and set of commands

**Hive** is used for creating reports

Operates on the server side of a cluster

A declarative programming language which means it allows users to express which data they want to receive.

**Access Data** last stage where users have access to the analysed and refined data:

Tools like **Impala** are often used

It’s a scalable platform easy to for everyone for searching and access data in Hadoop.

No code

**Hue:**

Stands for Hadoop user experience

It allows to upload, browse and query data

You can run Pig jobs and workflow in Hue.

It also provides a SQL editor for several query languages like Hive and mySQL

Recap:

Stage1: Ingest

Flume & Sqoop

Stage 2: Store data

HDFS & HBASE (tables) & Cassandra (NoSQL)

Stage 3 : Analyse

Pig(SQL editor) & Hive

Stage 4: Access

HUE (Hadoop user experience) & Impala (no code)

## HDFS

Hadoop Distributed File System

A distributed file system is a file system that is distributed on multiple servers

And allows programmers to access or store files from any network or computer.

HDFS is the storage layer of Hadoop

It works by splitting the files into blocks, creating replicas of the blocks and store them on different machines.

It is built to access streaming data seamlessly

Streaming means that HDFS provides a constant bitrate when transferring data rather than having the data being transferred in waves

HDFS uses a command line interface to interact with Hadoop.

Key features of HDFS are

That its cost efficient:

The commodity hardware that stores the data is not expansive, and therefore reduces storage costs

It can store large amounts of data as large aas petabytes in any format (tabular and none tabular) It splits large amounts of data into small chunks called blocks

One of the great features of HDFS is its ability to replicate and minimize the costs associated with data losses when there is a failure of one of the hardware units

That capability makes HDFS Fault Tolerant in the event of a data loss of one of the computers, the data can be found on another computer and work continuous

HDFS is also highly scalable; A single cluster can be scaled into hundreds of nodes

Portability is also one of the key features as HDFS is designed to easily move from one platform to another

**HFS concepts:**

**Blocks**

When HDFS receives files, files are broken into smaller chunks called blocks.

A Block is the minimum amount of data that can be read or written

It provides fault tolerance.

Depending on your system configuration, the default block size could be 64 or 128 megabytes

For example if we had a 500 megabyte file with a default block chunk size of 128 megabytes,

The file will be divided into 3 blocks of 128 megabytes and one block of 116 megabytes.

The only time you will have equal block size is the time you will have file size being a multiple of the default block size.

**Nodes**

A node is a single system which is responsible to store and process data

Think about it as one machine or computer in which data is stored.

Remember that HDFS follows the primary/secondary concept

HDFS has two types of nodes.

The primary node known as the named node, regulates file access to the client and maintains, manages and assign tasks to the secondary node also known as the data node.

There can be hundreds of data nodes in the HDFS that manage the storage system.

They perform read and write requests at the instruction of the name node.

**Rack(support) awareness in HDFS**

When performing operations like read and write, it is important that the name node maximizes performance by choosing the data nodes closest to themselves

This could be by choosing data nodes on the same rack or in nearby racks.

This is called

Rack awareness.

A rack is the collection of about forty to fifty data nodes using the same network switch.

Rack awareness is used to reduce the network traffic and improve cluster performance.

To achieve rack awareness, the name node keeps the rack ID information.

Replication is done by rack awareness as well.

It is done by being sure replicas of a data node are in different racks.

So, if a rack is down, you can still obtain the data from another rack.

HDFS is known for optimizing replication.

HDFS uses the rack awareness concept to create replicas to make sure that the data is reliable and that the network bandwidth is properly utilized.

**Replication**

Creating a copy of the data block

Copies are created for backup purposes

Replication factor is defined as the number of times the data block was copied

Depending on your configuration, you can set the number of copies you want

If the replication factor is 2, it will create two copies of each block.

Then copies are distributed in different racks.

**Read and Write Operations**

HDFS allows write once read many operations

It means that you cannot edit files that are already stored in HDFS, but you can add new data to them.

**Read operation**

Let’s see how the read operation works. Assuming that we have a text file,

The client will send a request to the primary node which is the named node, containing the location of the data nodes containing blocks.

The named node will verify that the client has the correct privileges and provide the client with the locations.

A client in HDFS interacts with the primary and secondary nodes to fulfill a user’s request.

The client will then send a request to the closest data nodes through an FS Data Input stream object by calling the read method to read all the files.

And when the client is done, the client will use the close method to end the session

**Write operation**

Just like in the read operation, the name node confirms that the client has the right privileges.

The name node then makes sure that the file doesn’t already exist in the system

If the file already exist, the client will receive an IO exception.

If the file doesn’t exist, the client receives a write permission together with the data nodes.

Once the client is done, the data nodes start creating replicas and send a confirmation to the client

Hadoop follows the concept of a primary, secondary node architecture

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

The structure is such that per cluster there is one named node and multiple data nodes.

Internally a file is split into one or multiple blocks and these blocks are stored in a set of data nodes.

The name node oversees opening, closing, renaming file operations, and mapping file blocks to the data nodes

The data nodes are responsible for read and write requests from the client and perform the client and perform the creation and deletion of file blocks based on instructions form the named node.