

**School of InfoComm Technology**

**Distributed Data Pipelines**

Diploma in Data Science (DS)

Diploma in Information Technology (IT)

October 2023 Semester

**INDIVIDUAL ASSIGNMENT 1**

(Section A)

**Deadline for Submission:**

**15th Dec 2023 (Friday), 2359 Hours**

|  |  |  |
| --- | --- | --- |
| Student Name | : | Akul Arun |
| Student Number | : | S10243523 |
|  |  |  |

**Penalty for late submission:**

10% of the marks will be deducted every day after the deadline.

**NO** submission will be accepted after 17th Dec 2023, 23:59.

Table of Contents

[Overview 2](#_Toc153589502)

[What is Hadoop? 2](#_Toc153589503)

[What is Apache Spark? 3](#_Toc153589504)

[Data Ingestion 4](#_Toc153589505)

[Data Storage 5](#_Toc153589506)

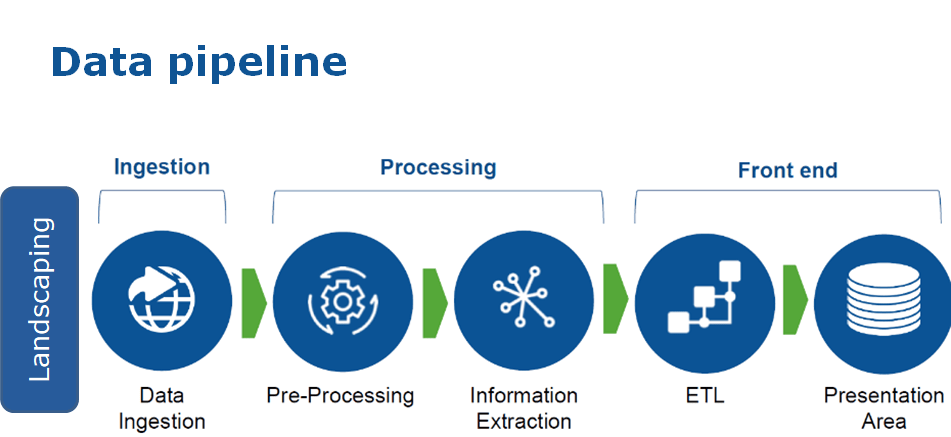
[Data Processing 6](#_Toc153589507)

[Data Application 7](#_Toc153589508)

[Similarities Between Hadoop & Apache Spark 7](#_Toc153589509)

# Overview

Both Hadoop and Apache Spark are extremely important software that supports the workloads of many companies. Hadoop is used for big data processing while Apache Spark complements Hadoop and enables advanced and quick analytics and machine learning capabilities as it uses in-memory caching. I recently got more interested in Apache Spark due to the fintech event I attended, and many experts shared with me the importance of it for company operations. I will be sharing about the differences Hadoop and Apache Spark have for pipelining.



# What is Hadoop?

Introduced in 2005, Hadoop is an open-source framework used to store and process large amounts of data across clusters of distributed computers.

Hadoop consists of 4 main modules:

* **Hadoop Distributed File System (HDFS)** – Data Storage
* **MapReduce** – Data Processing
* **YARN** – Resource Management
* **Hadoop Common** – Collection of Libraries

Hadoop Ecosystem

A diagram of a company

Description automatically generated

# What is Apache Spark?

Open-sourced in 2010, Apache Spark is a powerful, fast, and open-source data processing engine that enables large-scale data processing and advanced analytics.

Core Components:

* Spark Core
* Spark Streaming
* Spark SQL
* MLlib
* GraphX

Programming Languages supported:

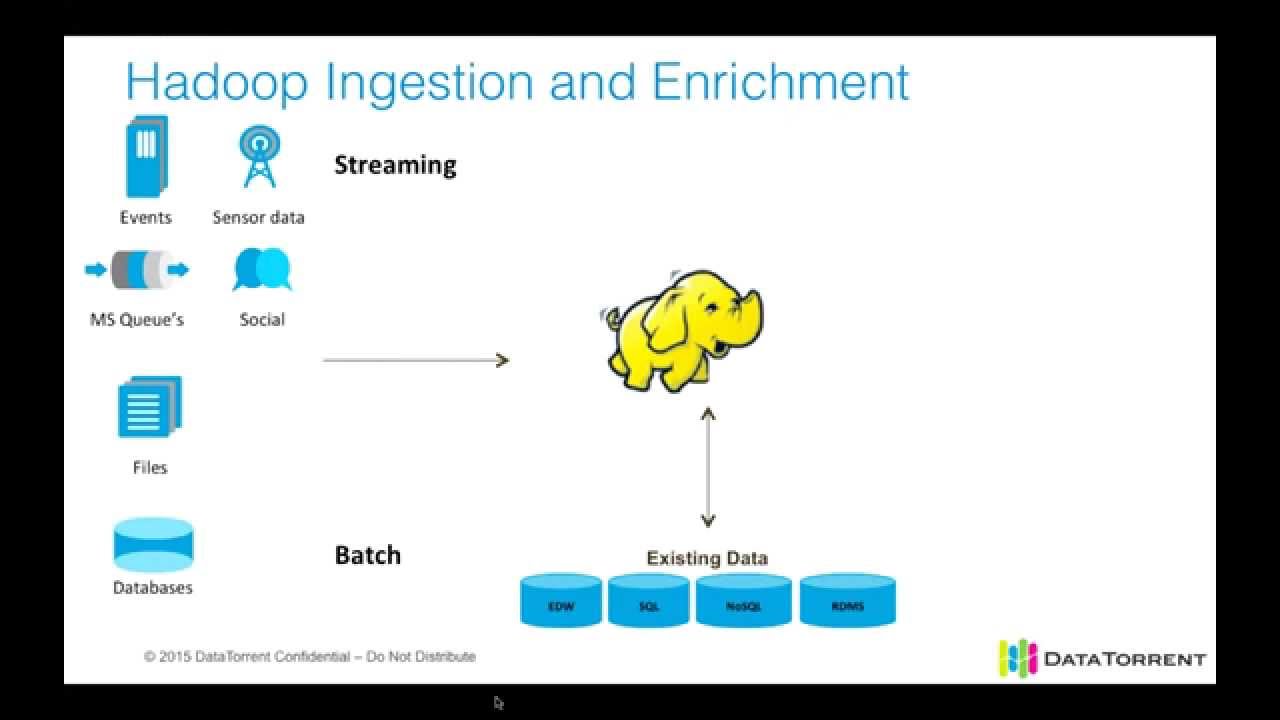
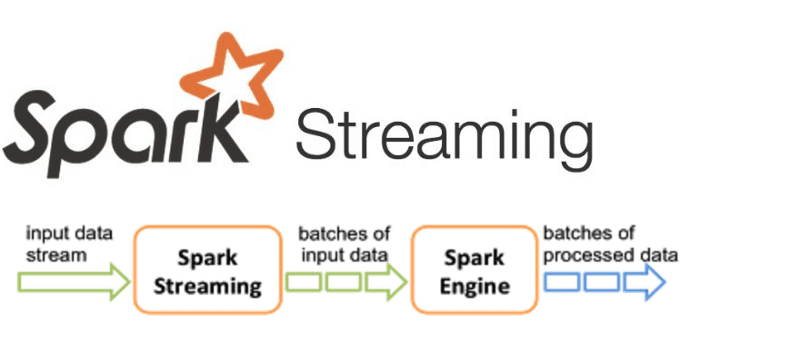
* Python
* Java
* Scala
* R

Apache Spark Ecosystem

A diagram of a computer system

Description automatically generated

# Data Ingestion



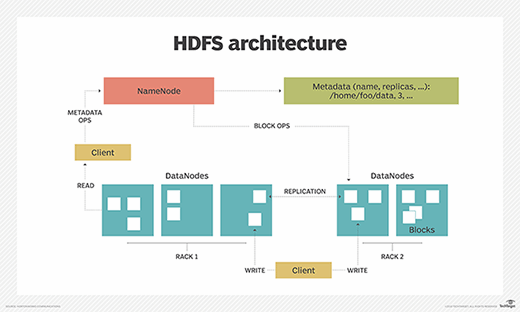
Ingestion in Apache Spark

Ingestion in Hadoop

|  |  |  |
| --- | --- | --- |
| Feature | Hadoop | Apache Spark |
| Data Ingestion | Primarily through tools like Flume and Sqoop | Supports various data sources, including HDFS, Hive, Kafka |
| **Real-time Ingestion** | Limited support for real-time data ingestion | Suited for real-time streaming data, supports Spark Streaming |
| **Latency** | Higher latency due to batch processing | Lower latency, suitable for near real-time and streaming use cases |
| **Ease of Use** | Requires custom MapReduce programs for ingestion | Offers higher-level APIs for easier data ingestion (e.g., Spark SQL, Dataframes) |

Top of Form

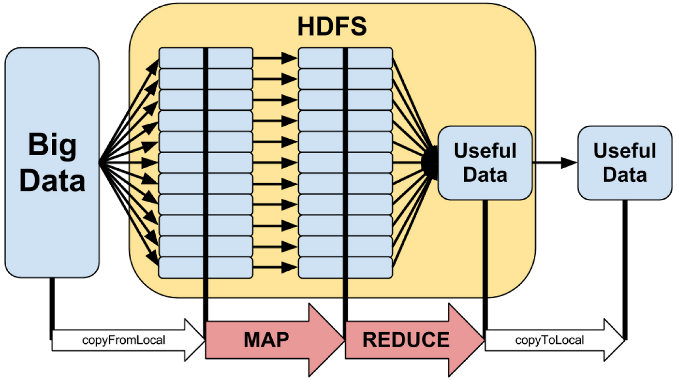
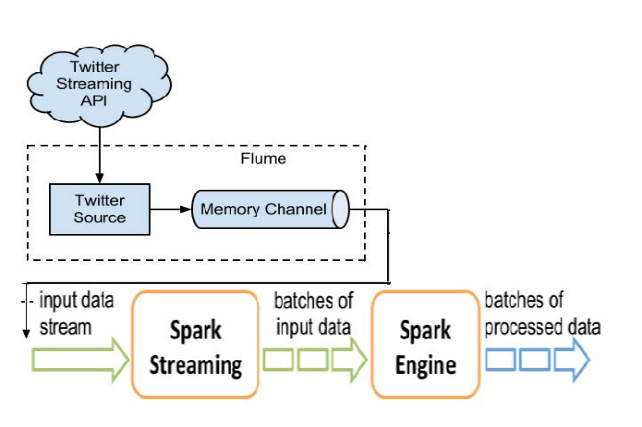
# Data Storage



|  |  |  |
| --- | --- | --- |
| **Feature** | Hadoop | Apache Spark |
| Storage System | Primary storage in Hadoop Distributed File System (HDFS) | Supports various storage systems, including HDFS, local file systems, and cloud storage |
| **Storage Format** | Typically uses plain text or binary formats like SequenceFile | Supports various data formats (Parquet, Avro, ORC) for optimized storage and processing |
| **Compression** | Commonly uses compression codecs like Gzip or Snappy | Offers compression options and supports various compression algorithms for data optimization |
| **Data Partitioning** | HDFS relies on block-based storage and replication for fault tolerance | Supports data partitioning with Resilient Distributed Datasets (RDDs) for parallel processing and fault tolerance |
| **Memory Usage** | Disk-based storage and processing | Leverages in-memory processing for faster access to data |
| **Ease of Access** | Access through Hadoop ecosystem tools (Hive, Pig, etc.) | Direct access using Spark APIs, including Spark SQL and DataFrames for simplified querying |
| **Data Catalog** | Often relies on external tools like Hive for metadata management | Built-in metadata management for efficient data cataloging and querying |

Top of Form

# Data Processing



|  |  |  |
| --- | --- | --- |
| **Feature** | Hadoop | Apache Spark |
| Processing | Primarily batch-oriented processing using MapReduce | Hybrid model supporting both batch and real-time processing with Spark Core and Spark Streaming |
| **Data Processing Model** | MapReduce programming model | Directed Acyclic Graph (DAG) model with RDDs (Resilient Distributed Datasets) and DataFrames |
| **Latency** | Higher latency due to batch processing | Lower latency, suitable for near real-time and streaming use cases |
| **Ease of Programming** | Requires writing complex MapReduce programs in Java | Provides higher-level APIs in Scala, Java, Python, and SQL, simplifying programming tasks |
| **Iterative Processing** | Inefficient for iterative algorithms | Efficient support for iterative algorithms, beneficial for machine learning and graph processing |
| **In-Memory Processing** | Limited support for in-memory processing | Leverages in-memory processing for faster data access and iterative computation |
| **Streaming Capabilities** | Limited support for real-time streaming | Comprehensive streaming capabilities with Spark Streaming for continuous data processing |

Top of Form

# Data Application

|  |  |  |
| --- | --- | --- |
| **Machine Learning** | Limited ML support, often requiring integration with external tools | Integrated MLlib library for machine learning, enabling a wide range of analytics and modeling tasks |
| **Graph Analytics** | Limited support for graph processing | GraphX library provides efficient graph processing capabilities |
| **Trends** | Big data applications | Real-time analytics and machine learning applications |

Top of Form

# Similarities Between Hadoop & Apache Spark

|  |  |
| --- | --- |
| Similarity Feature | Hadoop & Apache Spark |
| Distributed Processing | Both enable distributed data processing |
| **Scalability** | Scalable architectures allowing horizontal scaling |
| **Fault Tolerance** | Built-in fault tolerance mechanisms |
| **Cluster Management** | Utilize cluster management systems like YARN |

Top of Form