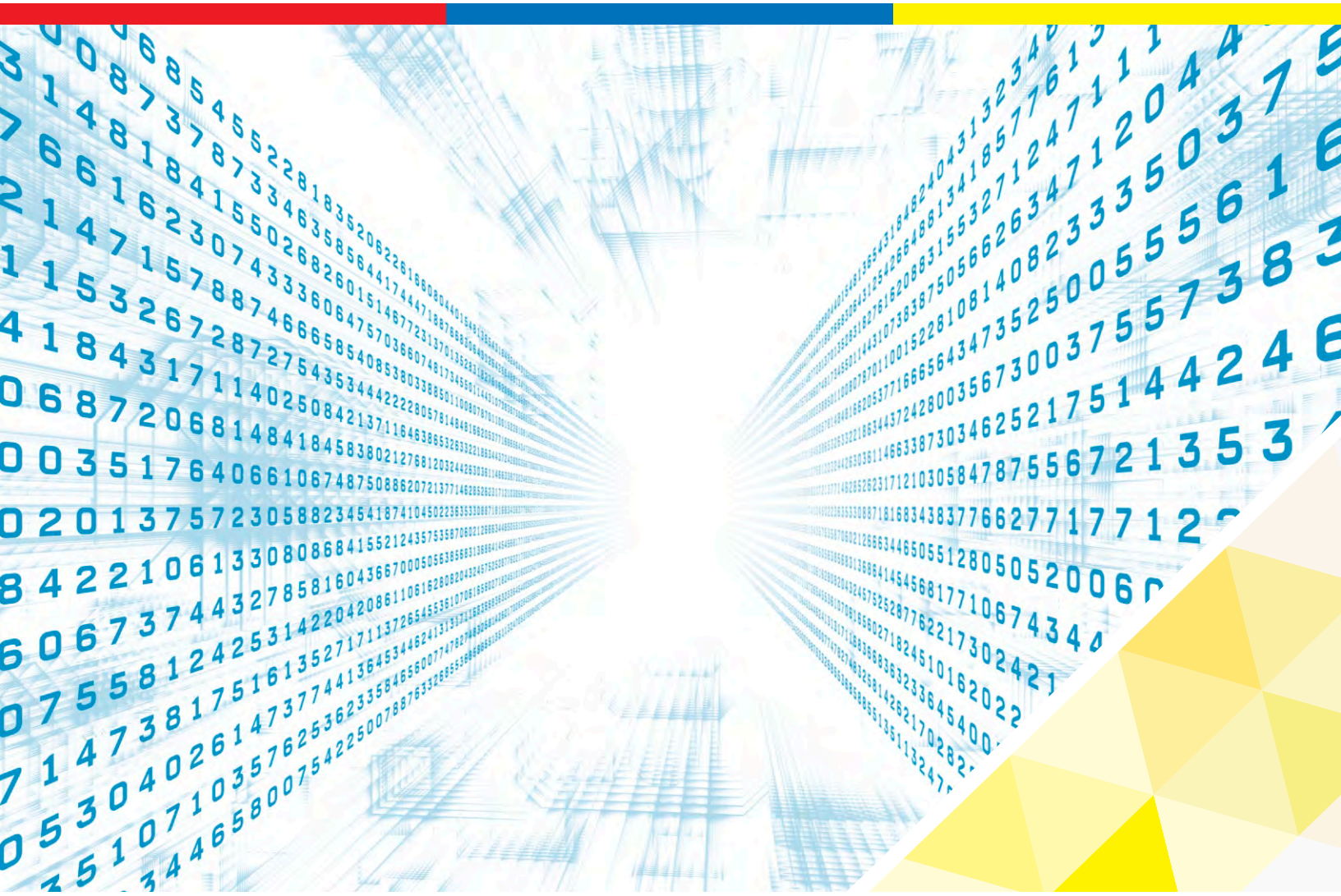
ACL Digital
an ALTEN group company

Innovate, Digitize, Transform

# Hadoop vs Apache Spark



## Introduction

"Any sufficiently advanced technology is indistinguishable from magic."- said Arthur C. Clark. Big data technologies and implementations are gaining traction and moving at a fast pace with novel innovations happening in its space. Hadoop, NoSQL, MongoDB and Apache Spark are the buzzwords with big data technologies - leaving a digital trace of data in every individual's life which can be used for analysis. Future of big data analytics market will revolve around Internet of Things (IoT), Sentiment Analysis, Trend Analysis, increase in sensor driven wearables, etc.

Hadoop and Spark are popular Apache projects in the big data ecosystem. Apache Spark is an open-source platform, based on the original Hadoop MapReduce component of the Hadoop ecosystem. Here we come up with a comparative analysis between Hadoop and Apache Spark in terms of performance, storage, reliability, architecture, etc.

# **Comparative Analysis**: Hadoop vs Apache Spark

## Hadoop - Overview

Apache developed Hadoop project as open-source software for reliable, scalable, distributed computing.

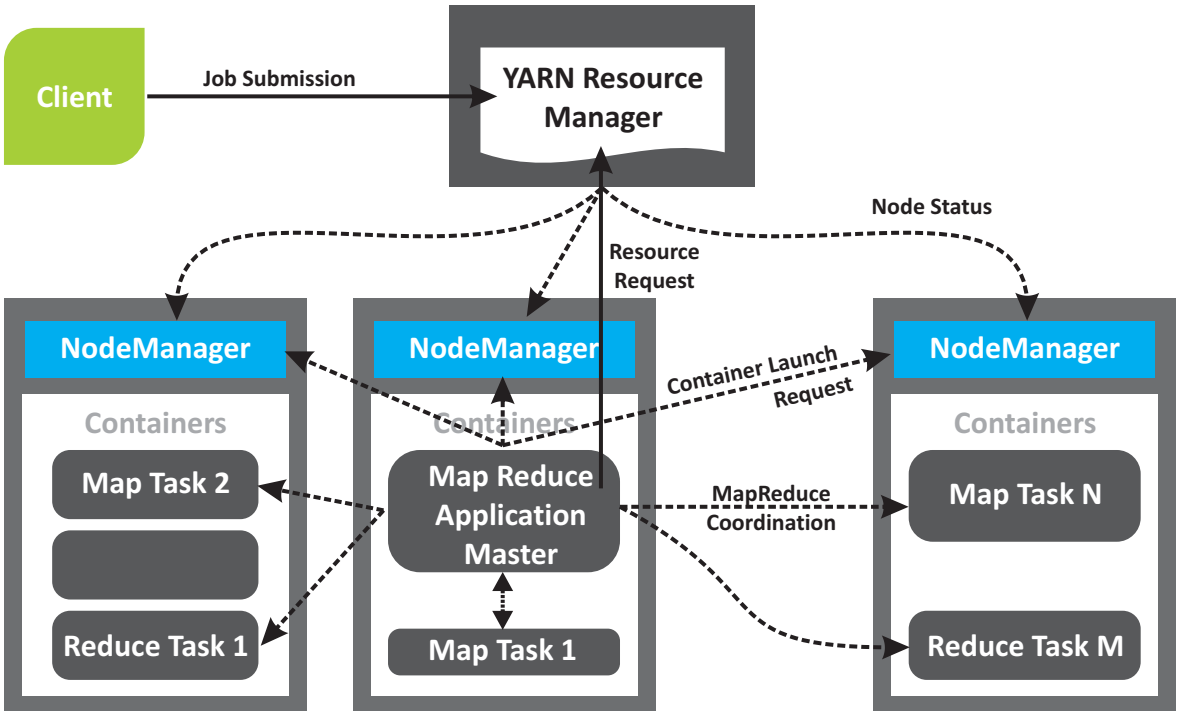
The Apache Hadoop software library is a framework that allows distributed processing of large datasets across clusters of computers using simple programming models. Hadoop can be easily scaled-up to multi cluster machines, each offering local storage and computation. Hadoop libraries are designed in such a way that it can detect the failed cluster at application layer and can handle those failures by it. This ensures high-availability by default.

The project includes these modules:

- Hadoop Common: These are Java libraries and utilities required for running other Hadoop modules. These libraries provide OS level and filesystem abstractions and contain the necessary Java files and scripts required to start and run Hadoop.
- Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.
- Hadoop YARN: A framework for job scheduling and cluster resource management.
- Hadoop MapReduce: A YARN-based system for parallel processing of large datasets.

## Hadoop - Architecture

Hadoop MapReduce, HDFS and YARN provide a scalable, fault-tolerant and distributed platform for storage and processing of very large datasets across clusters of commodity computers. Hadoop uses the same set of nodes for data storage as well as to perform the computations. This allows Hadoop to improve the performance of large scale computations by combining computations along with the storage.




## Hadoop Distributed File System – HDFS

HDFS is a distributed filesystem that is designed to store large volume of data reliably. HDFS stores a single large file on different nodes across the cluster of commodity machines. HDFS overlays on top of the existing filesystem. Data is stored in fine grained blocks, with default block size of 128MB. HDFS also stores redundant copies of these data blocks in multiple nodes to ensure reliability and fault tolerance. HDFS is a distributed, reliable and scalable file system.

## Hadoop YARN

YARN (Yet Another Resource Negotiator), a central component in the Hadoop ecosystem, is a framework for job scheduling and cluster resource management. The basic idea of YARN is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons.

## Hadoop MapReduce

MapReduce is a programming model and an associated implementation for processing and generating large datasets with a parallel, distributed algorithm on a cluster. Mapper maps input key/value pair to set of intermediate pairs. Reducer takes this intermediate pairs and process to output the required values. Mapper processes the jobs in parallel on every cluster and Reducer process them in any available node as directed by YARN.

## Hadoop - Advantages:

For organizations considering a big data implementation, Hadoop has many features that make it worth considering, including:

- **Flexibility** - Hadoop enables businesses to easily access new data sources and tap into different types of data to generate value from that data.
- **Scalability** – Hadoop is a highly scalable storage platform, as it can store and distribute very large datasets across hundreds of inexpensive commodity servers that operate in parallel.
- **Affordability** - Hadoop is open source and runs on commodity hardware, including cloud providers like Amazon.
- **Fault-Tolerance** - Automatic replication and ability to handle the failures at application layer makes Hadoop fault tolerant by default.

## Apache Spark - Overview

It is a framework for analyzing data analytics on a distributed computing cluster. It provides in-memory computations for increasing speed and data processing over MapReduce. It utilizes the Hadoop Distributed File System (HDFS) and runs on top of existing Hadoop cluster. It can also process both structured data in Hive and streaming data from different sources like HDFS, Flume, Kafka, and Twitter.

## Spark Stream

Spark Streaming is an extension of the core Spark API. Processing live data streams can be done using Spark Streaming, that enables scalable, high-throughput, fault-tolerant stream. Input Data can be from any sources like Web Stream (TCP sockets), Flume, Kafka, etc., and can be processed using complex algorithms with high-level functions like map, reduce, join, etc. Finally, processed data can be pushed out to filesystems (HDFS), databases, and live dashboards. We can also apply Spark's graph processing algorithms and machine learning on data streams.

## Spark SQL

Apache Spark provides a separate module Spark SQL for processing structured data. Spark SQL has an interface, which provides detailed information about the structure of the data and the computation being performed. Internally, Spark SQL uses this additional information to perform extra optimizations.

## Datasets and DataFrames

A distributed collection of data is called as Dataset in Apache Spark. Dataset provides the benefits of RDDs along with utilizing the Spark SQL's optimized execution engine. A Dataset can be constructed from objects and then manipulated using functional transformations.

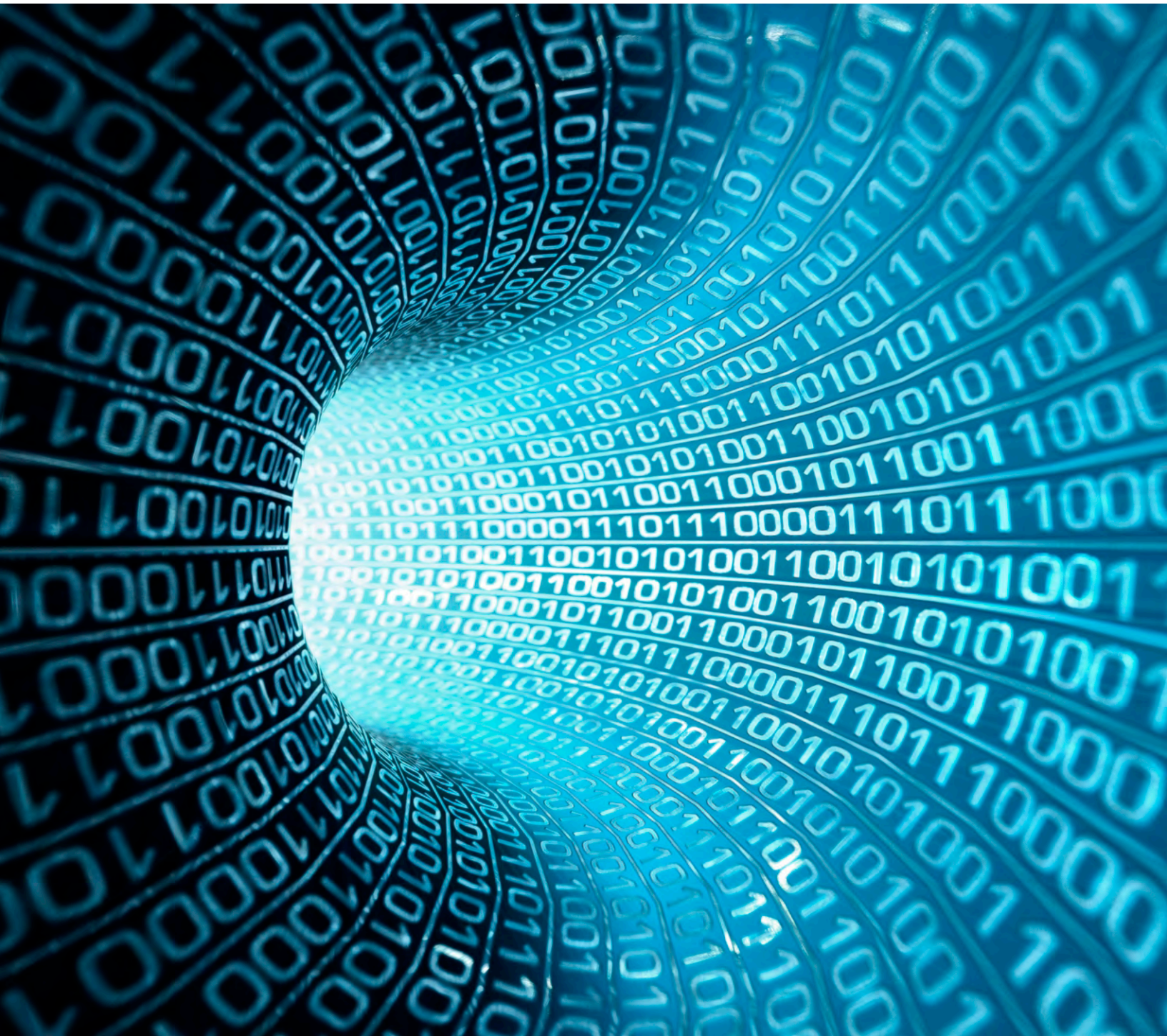
A DataFrame is a dataset organized into named columns. It is equally related to a relational database table or a R/Python data frame, but with richer optimizations under the hood. A DataFrame can be constructed, using various data source like structured data file or Hive tables or external databases or existing RDDs.

## Resilient Distributed Datasets (RDDs)

Spark works on fault-tolerant collection of elements that can be operated on in parallel, the concept called resilient distributed dataset (RDD). RDDs can be created in two ways, parallelizing an existing collection in driver program, or referencing a dataset in an external storage system, such as a shared filesystem, HDFS, HBase, etc.

## Apache Spark - Advantages

- **Speed** – Spark process data in-memory, making it run 100x faster than applications in Hadoop clusters. Also this enables even 10x faster when running on disk. This also reduces number of read/write to disc. Spark stores its intermediate processing data in-memory.
- **Ease of Use** – Spark lets developers to write applications in Java, Python or Scala, so that they can create and run their applications on their familiar programming languages.
- **Combines SQL, Streaming and Complex Analytics** – In addition to simple "map" and "reduce" operations, Spark supports streaming data, SQL queries, and complex analytics such as graph algorithms and machine learning out-of-the-box.
- **Runs Everywhere** – Spark runs on any system, either Hadoop, or standalone, or in the cloud. It can access diverse data sources including HDFS, HBase, S3 and Cassandra.



## Conclusion

Hadoop stores data on disk, whereas Spark stores data in-memory. Hadoop uses replication to achieve fault tolerance whereas Spark uses resilient distributed datasets (RDD), which guarantees fault tolerance that minimizes network I/O. Spark can run on top of HDFS along with other Hadoop components. Spark has become one more data processing engine in Hadoop ecosystem, providing more capabilities to Hadoop stack for businesses.

For developers, there is no difference between the two. Hadoop is a framework, where one can write MapReduce jobs using Java classes. Whereas Spark is a library that enables writing code for parallel computation via function calls. For administrators running a cluster, there is an overlap in general skills, such as cluster management, code deployment, monitoring configuration, etc.

Here is a quick comparison guideline before concluding.

| Aspects | Hadoop | Apache Spark |
|---|---|---|
| Difficulty | MapReduce is difficult to program and needs abstractions. | Spark is easy to program and does not require any abstractions. |
| Interactive Mode | There is no in-built interactive mode, except Pig and Hive. | It has interactive mode. |
| Streaming | Hadoop MapReduce just get to process a batch of large stored data. | Spark can be used to modify in real time through Spark Streaming. |
| Performance | MapReduce does not leverage the memory of the Hadoop cluster to the maximum. | Spark has been said to execute batch processing jobs about 10 to 100 times faster than Hadoop MapReduce. |
| Latency | MapReduce is disk oriented completely. | Spark ensures lower latency computations by caching the partial results across its memory of distributed workers. |
| Ease of coding | Writing Hadoop MapReduce pipelines is complex and lengthy process. | Writing Spark code is always more compact. |