Overview on Apache Hadoop

- *Apache Hadoop is an **open source framework** based on Java that is used to efficiently store and process large datasets ranging in size from gigabytes to petabytes of data.*
- *Hadoop is designed to scale up from a single computer to thousands of clustered computers, with each machine offering local computation and storage.*
- *It uses **distributed storage** and **parallel processing** to handle big data and analytics jobs, breaking workloads down into smaller workloads that can be run at the same time.*

How does Hadoop work?

- *Hadoop allows the distribution of datasets across a cluster of commodity hardware. Processing is performed in parallel on multiple servers simultaneously.*
- *Software clients input data into Hadoop.*
    - ***HDFS** handles metadata and the distributed file system.*
    - ***MapReduce** then processes and converts the data.*
    - ***YARN** divides the jobs across the computing cluster.*
- *All Hadoop modules are designed with a fundamental assumption that hardware failures should be automatically handled in software by the framework.*

History of Hadoop

- *Hadoop has its origins in the early era of the **World Wide Web**.*
- *As the Web grew to millions and then billions of pages, the task of searching and returning search results became one of the most prominent challenges.*
- *Startups like Google, Yahoo, and AltaVista began building frameworks to automate search results.*
- *One project called **Nutch** was built by computer scientists **Doug Cutting** and **Mike Cafarella** based on Google's early work on **MapReduce** and **Google File System**.*
- *Nutch was eventually moved to the Apache open source software foundation and was split between Nutch and Hadoop.*
- *Yahoo made Hadoop, **open sourced** in 2008.*
- *Hadoop is referred to as an acronym for **High Availability Distributed Object Oriented Platform**, it was originally named after Cutting's son's toy elephant.*

Modules of Hadoop

Hadoop consists of four main modules:

1. ***Hadoop Distributed File System (HDFS)***
    - *HDFS is a distributed file system in which individual Hadoop nodes operate on data that resides in their local storage.*
    - *This removes network latency, providing high-throughput access to application data.*

   - *In addition, administrators don't need to define schemas up front.*

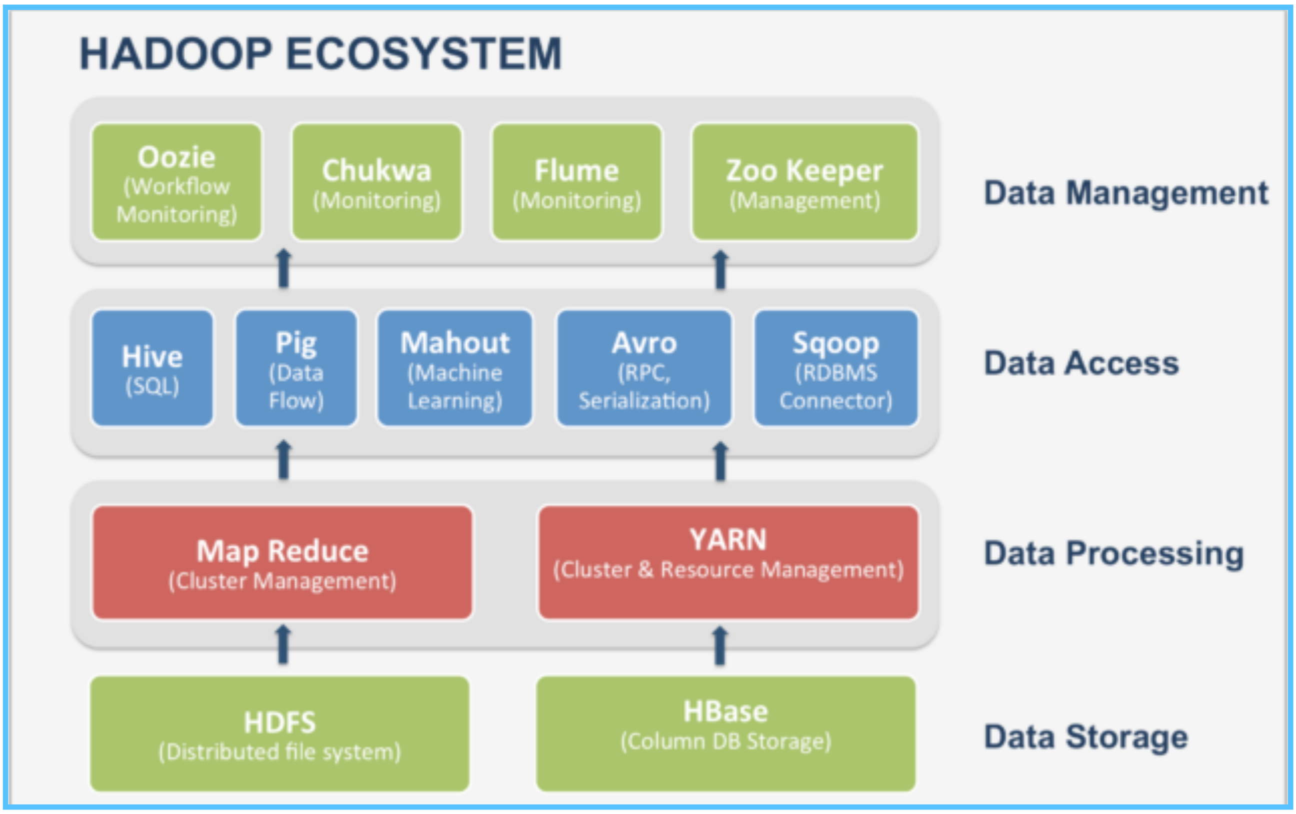
2. ***Yet Another Resource Negotiator (YARN)***
   - *Manages and monitors cluster nodes and resource usage.*
   - *It schedules jobs and tasks.*
3. ***MapReduce***
   - *A framework that helps programs do the **parallel computation** on data.*
   - *The map task takes input data and converts it into a dataset that can be computed in **key value** pairs.*
   - *The output of the map task is consumed by reduce tasks to aggregate output and provide the desired result.*
4. ***Hadoop Common***
   - *Provides common Java libraries and utilities that can be used across all modules.*

- *Beyond HDFS, YARN, and MapReduce, the entire Hadoop open source ecosystem like Apache Pig, Apache Hive, Apache HBase, Apache Spark, Presto, and Apache Zeppelin help to collect, store, process, analyze, and manage big data.*

Overview on Hadoop Ecosystem

Hadoop has a large ecosystem of open source tools that can augment and extend the capabilities of the core module.

- ***Apache Hive***


HADOOP ECOSYSTEM
Oozie (Workflow Monitoring)
Chukwa (Monitoring)
Flume (Monitoring)
Zoo Keeper (Management)
Data Management
Hive (SQL)
Pig (Data Flow)
Mahout (Machine Learning)
Avro (RPC, Serialization)
Sqoop (RDBMS Connector)
Data Access
Map Reduce (Cluster Management)
YARN (Cluster & Resource Management)
Data Processing
HDFS (Distributed file system)
HBase (Column DB Storage)
Data Storage

  - *A data warehouse that allows programmers to work with data in HDFS using a query language called **HiveQL**, which is similar to SQL.*

- ***Apache HBase***
  - *An open source non-relational distributed database often paired with Hadoop.*
- ***Apache Pig***
  - *A tool used as an abstraction layer over MapReduce to analyze large sets of data and enables functions like filter, sort, load, and join.*
- ***Apache Impala***
  - *Open source, massively parallel processing SQL query engine often used with Hadoop*
- ***Apache Sqoop***

  - *A command-line interface application for efficiently transferring bulk data between relational databases and Hadoop*
- ***Apache ZooKeeper***
  - *An open source server and high performance coordination service for distributed applications.*
- ***Apache Oozie***
  - *A workflow scheduler for Hadoop jobs*
- ***Ambari***
  - *A web-based tool for provisioning, managing and monitoring Hadoop clusters.*
- ***Avro***
  - *A data serialization system.*
- ***Mahout***
  - *A scalable machine learning and data mining library.*
- ***Submarine***
  - *A unified AI platform for running machine learning and deep learning workloads in a distributed cluster.*
- ***Tez***
  - *A generalized data flow programming framework, built on YARN; being adopted within the Hadoop ecosystem to replace MapReduce.*

Benefits of Hadoop

- ***Scalability***
    - *Hadoop is the primary tools to store and process huge amounts of data quickly.*
    - *It is achieved by distributed computing model which enables the fast processing of data that can be rapidly scaled by adding computing nodes.*
- ***Low cost***
    - *It an open source framework.*
    - *Hadoop is a low-cost option for the storage and management of big data.*
- ***Flexibility***
    - *Hadoop allows for flexibility in data storage as data does not require preprocessing before storing it*
- ***Resilience***
    - *Hadoop allows for **fault tolerance** and **system resilience**, if one of the hardware nodes fail, jobs are redirected to other nodes.*
    - *Data stored on one Hadoop cluster is replicated across other nodes within the system to fortify against the possibility of hardware or software failure.*

Challenges of Apache Hadoop

- ***MapReduce complexity and limitations***
  - *MapReduce can be a difficult tool to utilize for **complex jobs**, such as interactive analytical tasks.*
  - *The MapReduce ecosystem is quite large, with many components for different functions that can make it difficult to determine what tools to use.*
- ***Security***
  - ***Data sensitivity** and **protection** can be issues as Hadoop handles such large datasets.*
- ***Governance and management***
  - *It does not have many **robust tools** for data management and governance, nor for data quality and standardization.*
- ***Talent gap***
  - *Hadoop has an acknowledged talent gap. Finding developers with the combined requisite skills in Java to program MapReduce, operating systems, and hardware can be difficult.*

Popular Use cases of Hadoop

Here are some common uses cases for Apache Hadoop:

- ***Analytics and big data***
  - *A wide variety of companies and organizations use Hadoop for research, production data processing, and analytics that*

require processing terabytes or petabytes of big data, storing diverse datasets, and data parallel processing.

- ***Data storage and archiving***
  - *As Hadoop enables mass storage on commodity hardware, it is useful as a low-cost storage option for all kinds of data, such as transactions, click streams, or sensor and machine data.*

- ***Data lakes***
  - *Since Hadoop can help store data without preprocessing, it can be used to complement to data lakes, where large amounts of unrefined data are stored.*
- ***Marketing analytics***
  - *Marketing departments often use Hadoop to store and analyze customer relationship management (CRM) data.*
- ***Risk management***
  - *Banks, insurance companies, and other financial services companies use Hadoop to build risk analysis and management models.*
- ***AI and machine learning***
  - *Hadoop ecosystems help with the processing of data and model training operations for machine learning applications.*

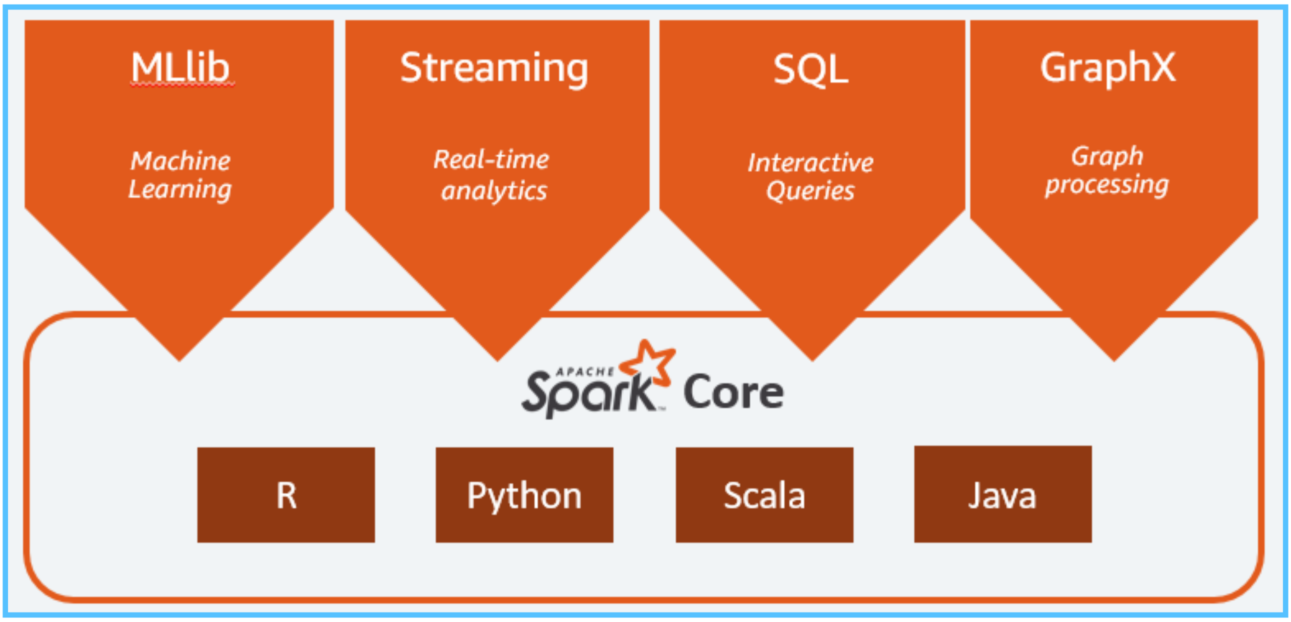
Overview on Apache Spark

- *Apache Spark is an* ***open source analytics engine*** *used for big data workloads. It is written in Scala.*
- *It can handle both batches as well as real-time analytics and data processing workloads.*
- *Spark was developed in 2009 at UC Berkeley's AMPLab. Today, it is maintained by the Apache Software Foundation and boasts the largest open-source community in big data, with over 1,000 contributors.*
- *Spark provides development APIs in* ***Java, Scala, Python*** *and* ***R****, and supports code reuse across multiple workloads—batch processing, interactive queries, real-time analytics, machine learning, and graph processing.*
- *It is a unified analytics engine for large-scale data processing with built-in modules for SQL.*
- *Spark's analytics engine processes data* ***10 to 100 times faster*** *than Hadoop for smaller workloads.*
- *It scales by distributing processing workflows across large clusters of computers, with built-in parallelism and fault tolerance.*
- *Spark can run on Apache Hadoop, Apache Mesos, Kubernetes, on its own, in the cloud—and against diverse data sources.*

Overview on Apache Spark Ecosystem

Apache Spark, the largest open-source project in data processing, is the only processing framework that combines data and artificial intelligence (AI). This enables users to perform large-scale data *transformations and analyses, and then run state-of-the-art machine learning (ML) and AI algorithms. The Spark framework includes:*

- *Spark Core as the foundation for the platform*
- *Spark SQL for interactive queries*
- *Spark Streaming for real-time analytics*
- *Spark MLlib for machine learning*
- *Spark GraphX for graph processing*



Spark Core

- *Underlying **execution engine** that schedules and dispatches tasks and coordinates input and output (I/O) operations.*

Spark SQL

- *Gathers information about structured data to enable users to optimize structured data processing.*

Spark Streaming and Structured Streaming

- *Both add stream processing capabilities.*
- ***Spark Streaming** takes data from different streaming sources and divides it into micro-batches for a continuous stream.*
- ***Structured Streaming**, built on Spark SQL, reduces latency and simplifies programming.*

Machine Learning Library (MLlib)

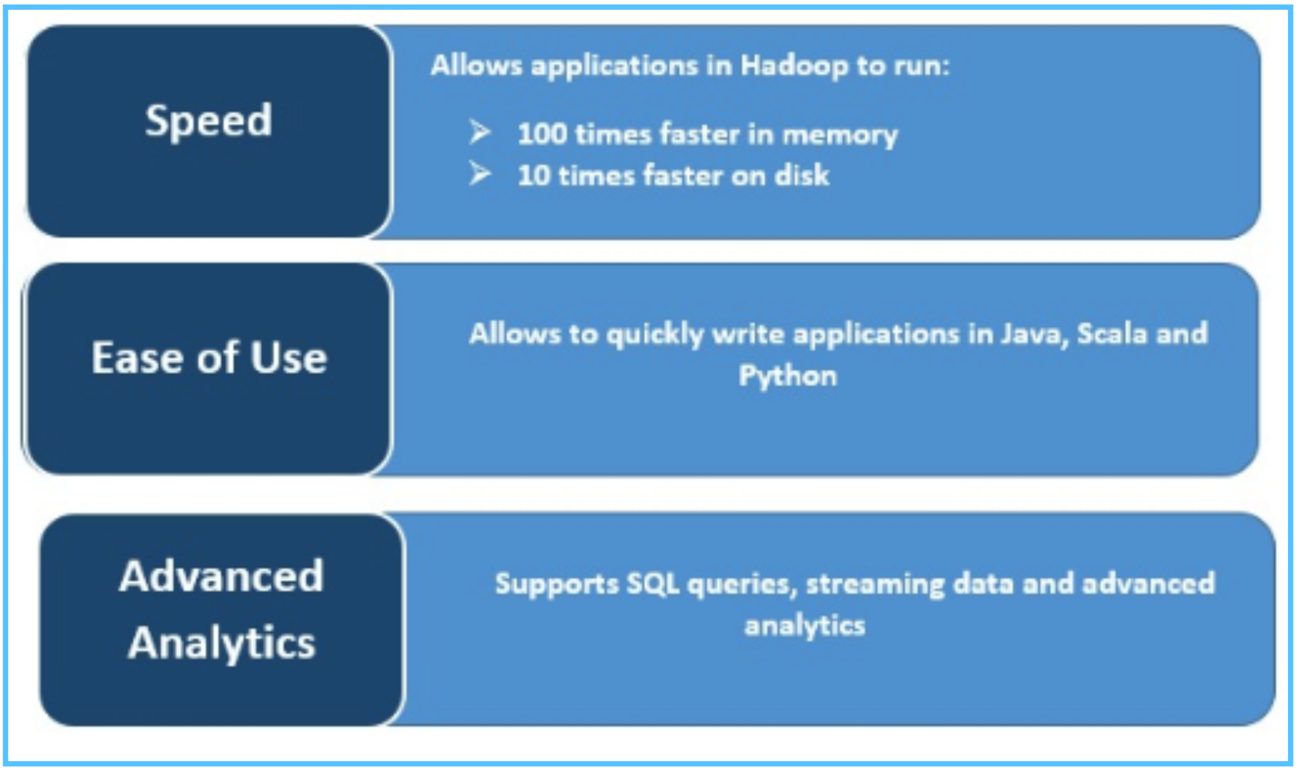
- *A set of machine learning algorithms for scalability plus tools for **feature selection** and building ML pipelines.*
- *The primary API for MLlib is DataFrames, which provides uniformity across different programming languages like Java, Scala and Python.*

GraphX

- *User-friendly computation engine that enables interactive building, modification and analysis of scalable, graph-structured data.*

Benefits of Apache Spark

- ***Speed***
  - *Spark executes very fast by caching data in memory across multiple parallel operations.*
  - *The main feature of Spark is its in-memory engine that increases the processing speed; making it up to 100 times faster than MapReduce when processed in-memory, and 10 times faster on disk, when it comes to large scale data processing.*
  - *Spark makes this possible by reducing the number of reading/writing to disk operations.*



Speed

Allows applications in Hadoop to run:

- 100 times faster in memory
- 10 times faster on disk

Ease of Use

Allows to quickly write applications in Java, Scala and Python

Advanced Analytics

Supports SQL queries, streaming data and advanced analytics

- ***Real-time stream processing***

  - *Apache Spark can handle real-time streaming along with the integration of other frameworks.*

- ***Supports Multiple Workloads***
  - *Apache Spark can run multiple workloads, including interactive queries, real-time analytics, machine learning, and graph processing.*
  - *One application can combine multiple workloads seamlessly.*
- ***Increased Usability***
  - *The ability to support several programming languages makes it **dynamic**.*
  - *It allows you to quickly write applications in Java, Scala, Python, and R; giving you a variety of languages for building your applications.*
- ***Advanced Analytics***
  - *Spark supports SQL queries, machine learning, stream processing, and graph processing.*

Difference between Apache Hadoop and Apache Spark

Feature	Hadoop	Spark
Architecture	Hadoop stores and processes data on external storage.	Spark stores and process data on internal memory.
Performance	Hadoop processes data in batches.	Spark processes data in real time.
Cost	Hadoop is affordable.	Spark is comparatively more expensive.
Scalability	Hadoop is easily scalable by adding more nodes	Spark is comparatively more challenging.
Machine learning	Hadoop integrates with external libraries to provide machine learning capabilities.	Spark has built-in machine learning libraries.
Ease of Use	Difficult to use.	Easier to use.
Latency	It is high latency computing framework.	It is a low latency computing and can process data interactively
Security	Better security features	Its security is currently in its infancy
Graph Processing	Algorithms like PageRank is used.	Spark comes with a graph computation library called GraphX.