Big Data Analytics

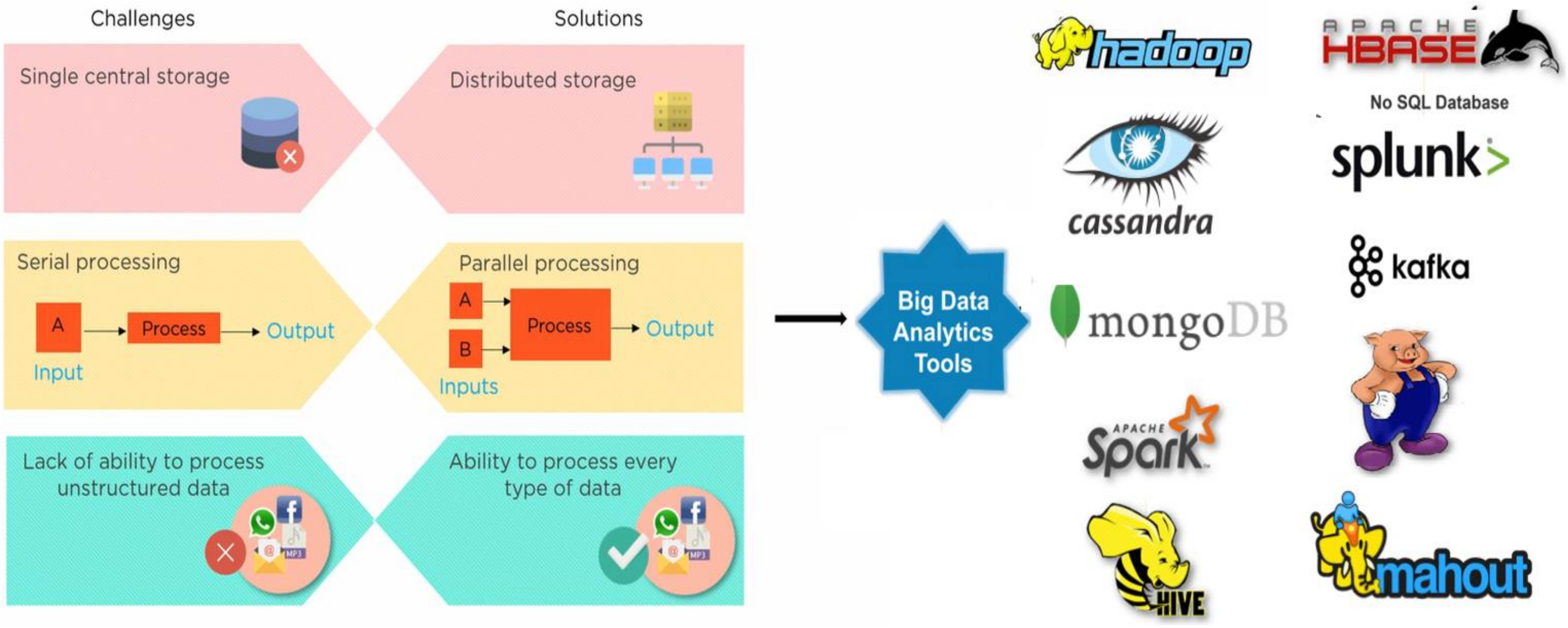
Academic Year 2022-23

Agenda

1. Big Data Challenges
2. Hadoop
3. Core Hadoop Components
4. Hadoop Ecosystem
5. Limitation of Hadoop

Prof. Prakash Parmar
Data Engineer | Data Analyst
Assistant Professor
Vidyalankar Institute of Technology, Mumbai
GATE Instructor: Vidyalankar Classes

Big Data Challenges?



Challenges

Single central storage

Serial processing

A → Process → Output

Input

Lack of ability to process unstructured data

Solutions

Distributed storage

Parallel processing

A, B → Process → Output

Inputs

Ability to process every type of data

→ Big Data Analytics Tools

Big Data Challenges?

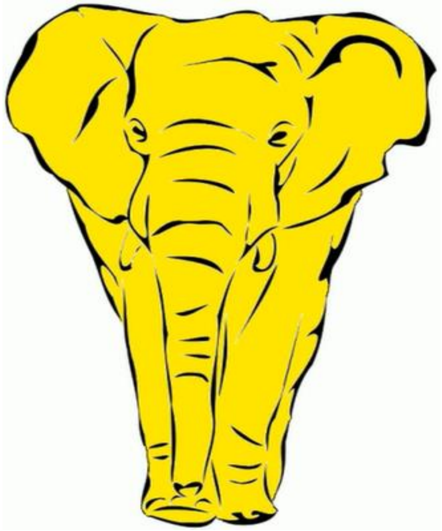
Challenges	Solutions	Tools
Volume: Single storage system- Managing and processing massive amounts of data that exceed the capacity of traditional data processing systems.	Distributed Computing and Storage	**Hadoop**: HDFS for distributed storage and MapReduce for parallel processing.
Velocity: Processing and analyzing streaming data in real-time as it's generated	Stream Processing	**Apache Spark**: Offers in-memory processing for iterative algorithms and interactive data analysis. **Apache Kafka**: Enables real-time data pipelines and stream processing. Apache Flink: Event-driven applications that need low-latency and high-throughput processing
Variety: Dealing with diverse data types and formats (structured, semi-structured, unstructured).	NoSQL database & processing tools	**HBase**: A NoSQL database that provides real-time access to unstructured and semi-structured data, particularly suited for random read/write operations. **MongoDB**: A document-oriented NoSQL database that is flexible for storing unstructured and semi-structured data. **Cassandra**: A distributed NoSQL database designed for handling large amounts of structured and unstructured data with high availability. **Apache Hive**: A data warehousing for analyzing structured data in Hadoop. **Spark SQL**: SQL queries on structured and semi-structured data, including JSON and Parquet files.

Big Data Challenges?

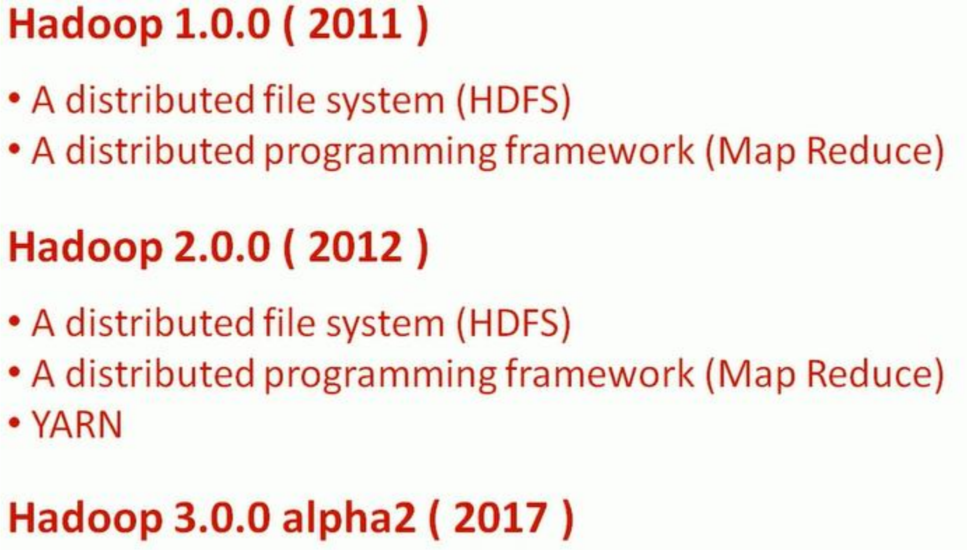
Challenges	Solutions	Tools
Veracity: Ensuring the accuracy and quality of the data, dealing with incomplete or noisy data	Data Quality and Cleansing	**Trifacta**: A data wrangling tool for clean, transform, and enrich data for analytics. **Talend**: ETL tool that helps with data quality and governance
Value: Extracting meaningful insights and value from the data.	Advanced Analytics and Machine Learning	**Apache Mahout**: machine learning algorithms that can be used with Hadoop. **Spark MLlib**: machine learning algorithms library used with Spark
Variability: Handling data that exhibits fluctuations in volume and characteristics	Scalability and Elasticity	**Docker and Kubernetes**: Containerization and orchestration tools that allow applications to be deployed and scaled easily. **Amazon EC2 Auto Scaling**: Automatically adjusts the number of EC2 instances based on load for Amazon Web Services (AWS) applications

Hadoop

Hadoop is an open-source framework that manages big data storage in a distributed way and process it parallelly.



- Distributed **Data storage** and **processing environment**.
- **Uses** commodity hardware
- Developed by: **Apache Software** Foundation in Dec, 2011 (based on google white paper)
- Written in: JAVA
- Current stable version: Hadoop 3.11



Hadoop 1.0.0 (2011)

- A distributed file system (HDFS)
- A distributed programming framework (Map Reduce)

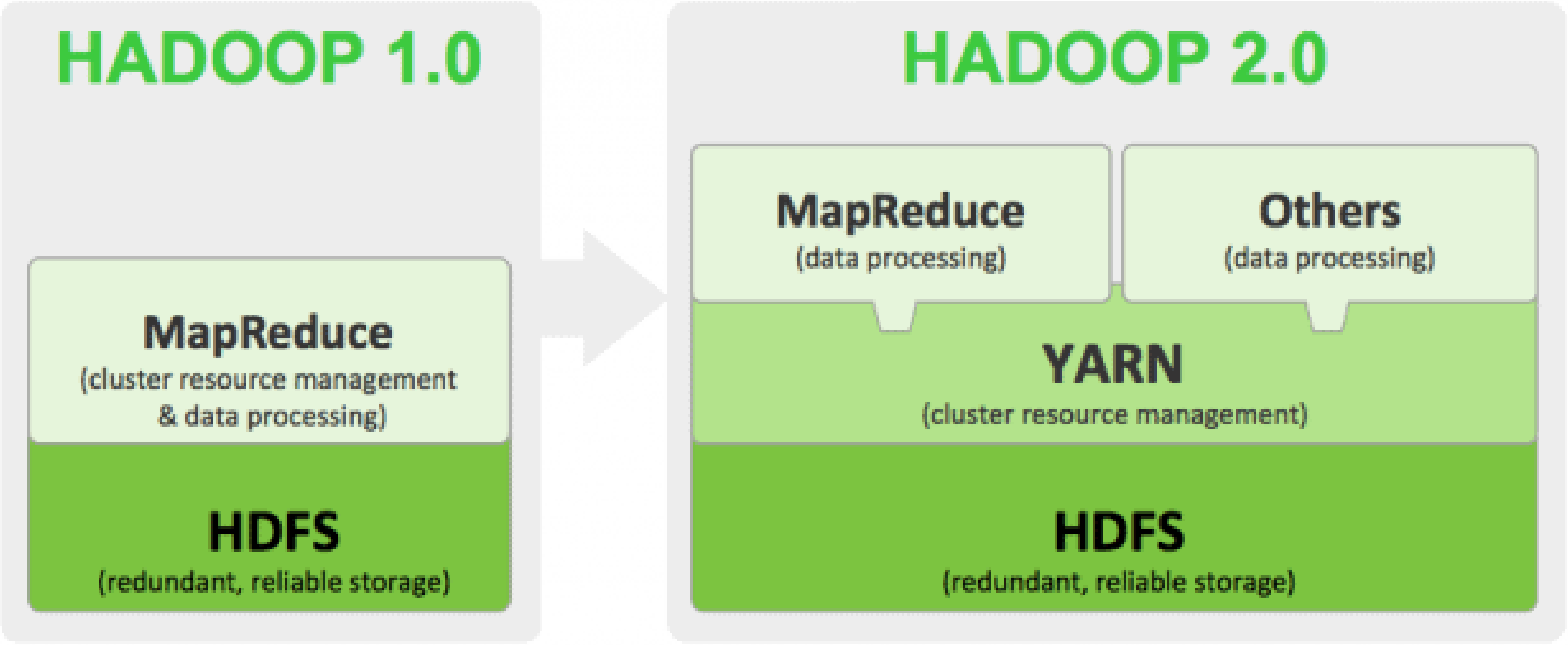
Hadoop 2.0.0 (2012)

- A distributed file system (HDFS)
- A distributed programming framework (Map Reduce)
- YARN

Hadoop 3.0.0 alpha2 (2017)

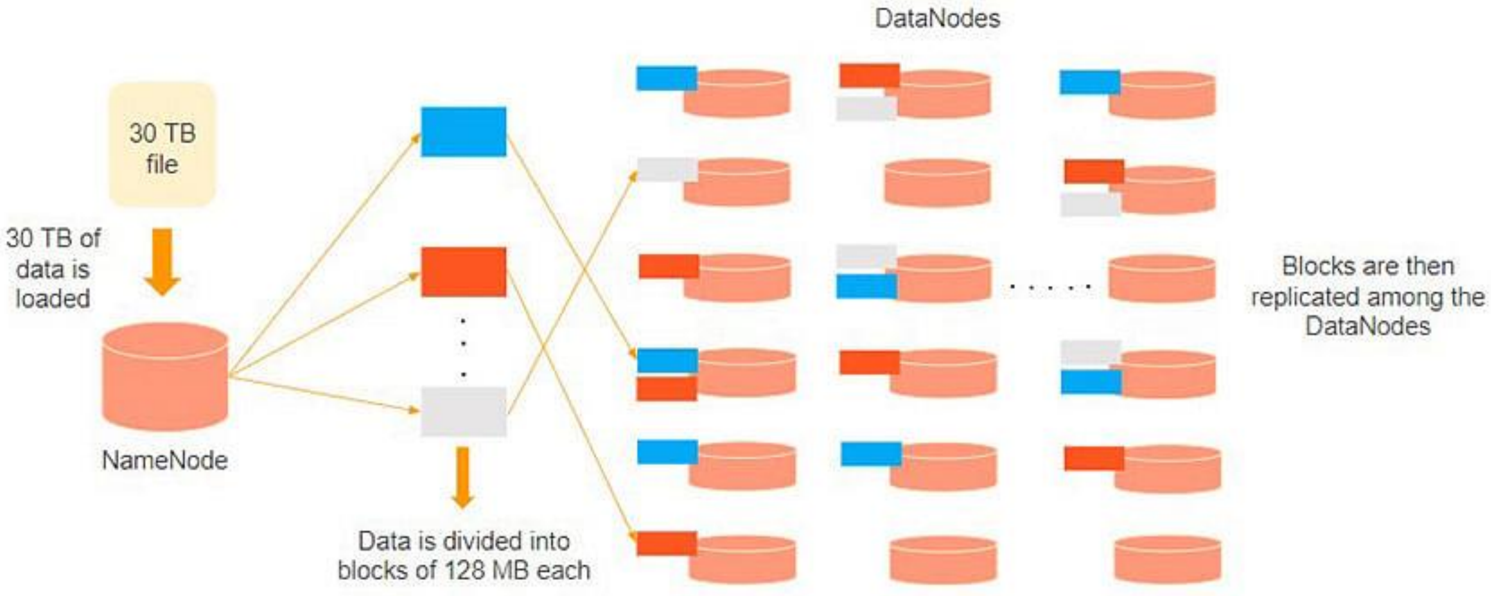
Components of Hadoop

Hadoop mainly have three component-HDFS, YARN and MapReduce


HADOOP 1.0
MapReduce
(cluster resource management & data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant, reliable storage)

HDFS (Hadoop Distributed File System): Storage

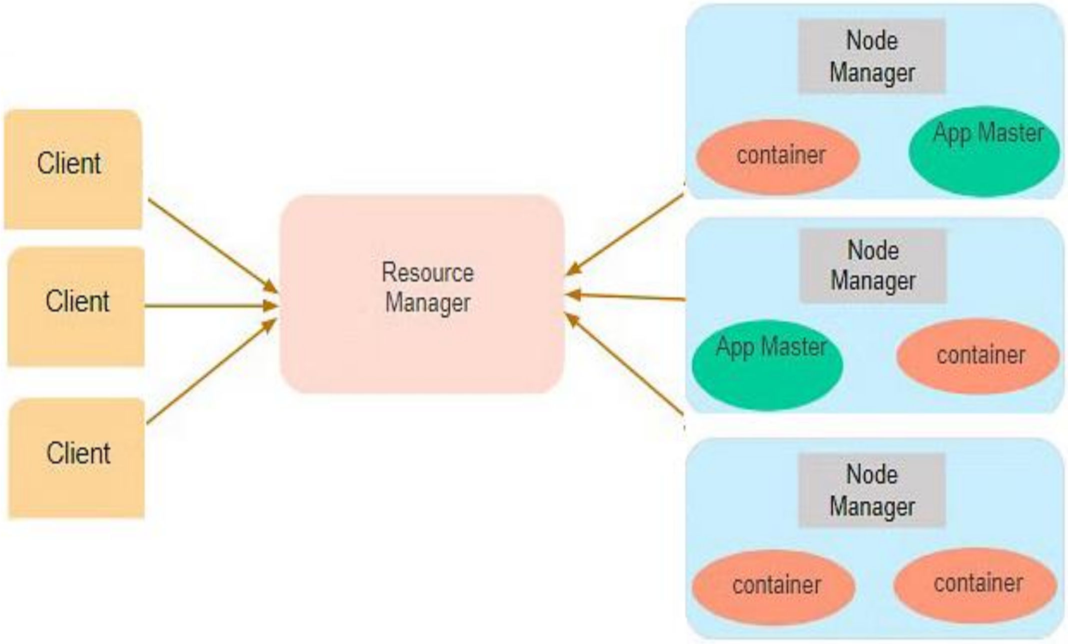
- HDFS stands for Hadoop Distributed File system.
- HDFS is specially designed for storing huge datasets in commodity hardware.
- Stored different types of data i.e., structured, unstructured and semi structured in distributed way.
- There are two components of HDFS - name node and data node. While there is only one name node, there can be multiple data nodes.


DataNodes
30 TB file
30 TB of data is loaded
NameNode
. . . . .
Blocks are then replicated among the DataNodes
Data is divided into blocks of 128 MB each

YARN (Yet Another Resources Negotiator) : Resource Manager

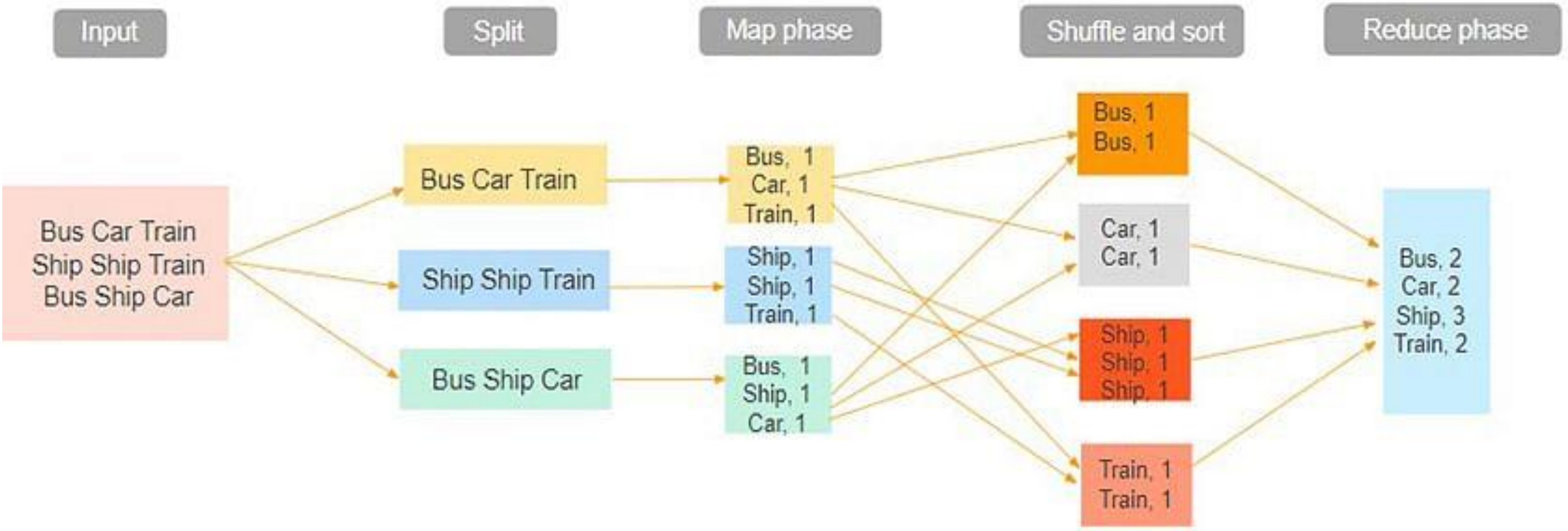
- YARN stands for Yet Another Resources Negotiator.
- Performs all processing activities by allocating resources (RAM, Memory etc) and scheduling tasks.
- **The most important component of YARN is**

1. **Resource manager**: manages resources and schedule application running on top of YARN.
2. **Node Manager**: It is the slave of YARN Architecture. Each Node Manager receives instruction from the Resource Manager and reports and handles containers on a single node. It manages container lifecycle, node health, log management, Node, and container resource usage
3. **Application Master**: It is a framework-specific library that negotiates resources from Resource Manager and works with the Node Manager or manages to execute and monitor containers and their resource consumption.

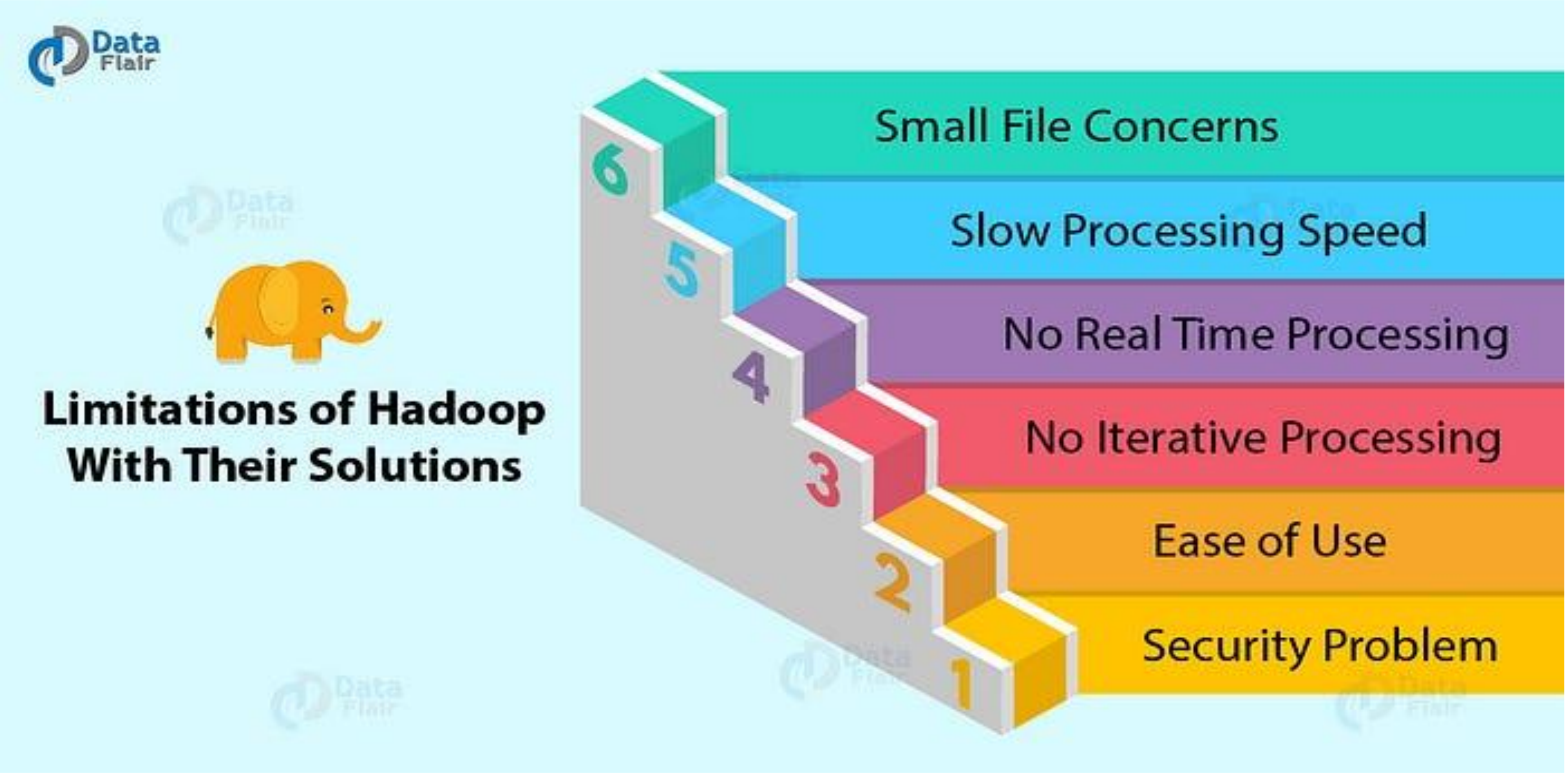


MapReduce: Data Processing using Programming

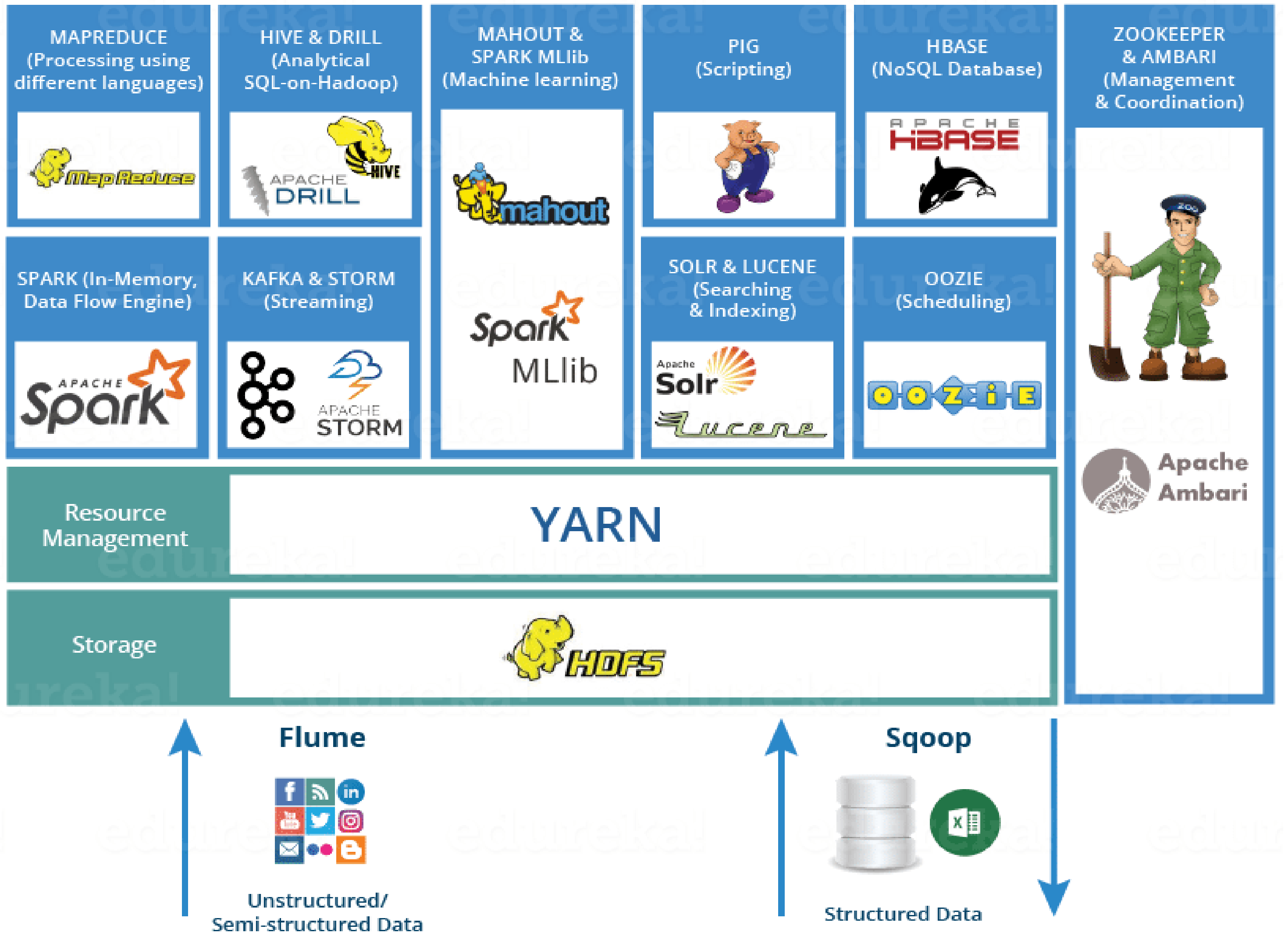
- MapReduce is a framework used for writing application that processes large data sets in a distributed manner with parallels algorithms.
- Map reduce have tow function: Map() and Reduce()
- Map function performs actions like filtering , grouping and sorting.
- Reduce function aggregates and summarizes the results produced by map function.



Hadoop Limitations



Hadoop Ecosystem



MAPREDUCE (Processing using different languages)

HIVE & DRILL (Analytical SQL-on-Hadoop)

MAHOUT & SPARK MLlib (Machine learning)

PIG (Scripting)

HBASE (NoSQL Database)

ZOOKEEPER & AMBARI (Management & Coordination)

SPARK (In-Memory, Data Flow Engine)

KAFKA & STORM (Streaming)

SOLR & LUCENE (Searching & Indexing)

OOZIE (Scheduling)

Resource Management — YARN

Storage — HDFS

Flume

Unstructured/ Semi-structured Data

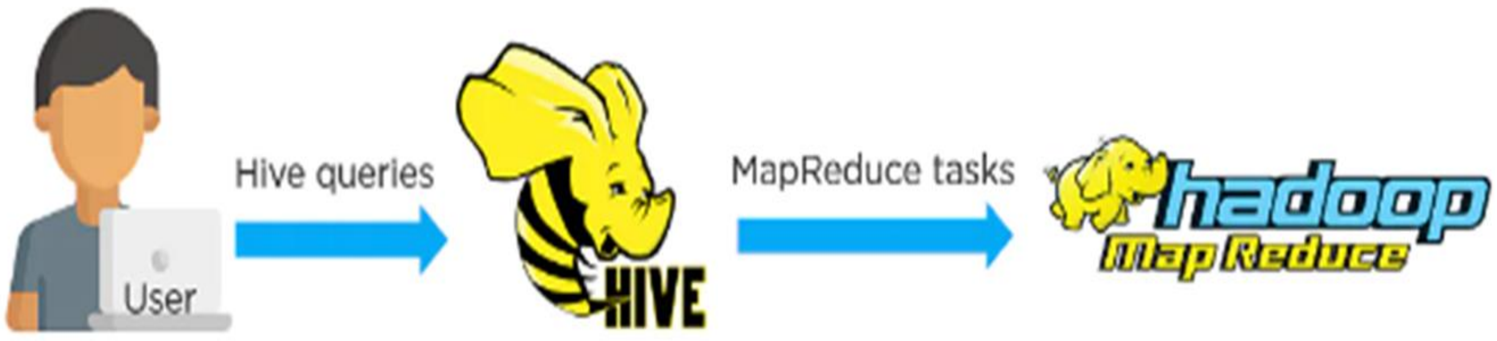
Sqoop

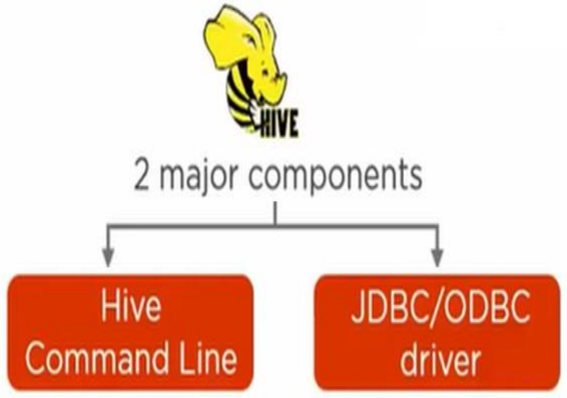
Structured Data

edureka!

HIVE: Data Processing Service using Query

- Hive is a data warehouse system that is used to query (reading, writing, managing) and analyze large datasets stored in the HDFS.
- Developed by Facebook to reduce the task of writing complex query in Map Reduce.
- Hive uses a query language called HiveQL (HIVE Query Language)., which is similar to SQL.
- 2 basic components: Hive command line and JDBC/ODBC driver.
- Used by analyst.
- Works on structured data but does not work on other type of data.



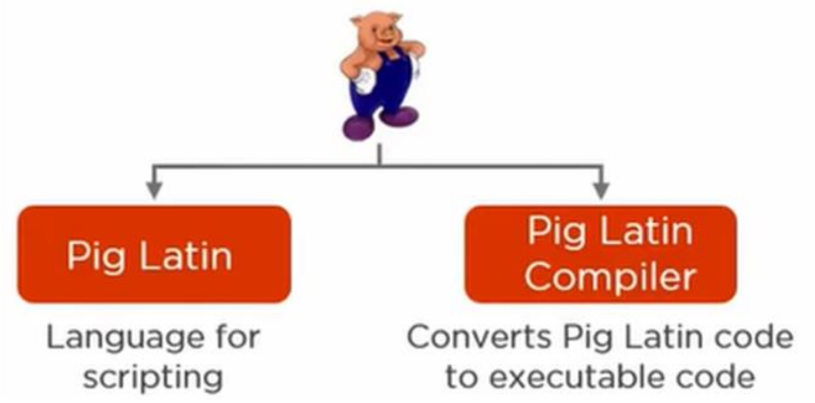


Pig: Data Processing Service using Query

- Pig is a scripting platform that runs on Hadoop clusters, used to process and analyze large datasets.
- Developed by Yahoo to reduce to task of writing complex query in Map Reduce.
- 1 line of pig = approx. 100 lines of Map reduce job.
- Pig scripting language is called PigLatin.
- Works on structured, semi-structured and unstructured data.
- Used by Programmers and researchers.
- 2 basic components: : Pig Latin (Pig language) and the pig runtime (for the execution environment).
- It gives you a platform for building data flow for ETL (Extract, Transform and Load).

Provides a platform for building data flow for ETL

1 lines of Pig Latin script is around 100 lines of MapReduce job



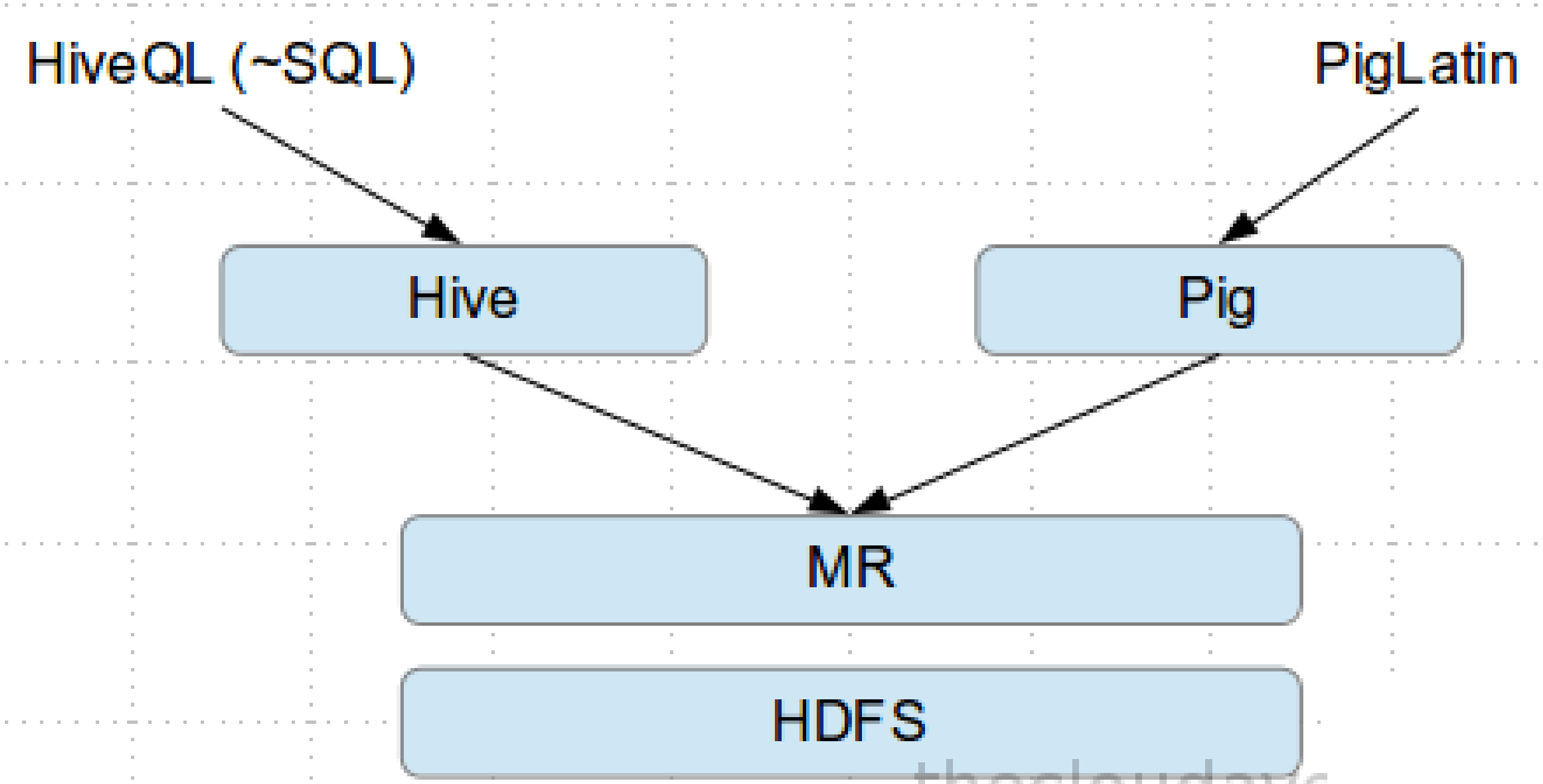
Pig Latin

Language for scripting

Pig Latin Compiler

Converts Pig Latin code to executable code

Difference between MapReduce, Hive & Pig



Difference between MapReduce, Hive & Pig

- Compiled Language
- Language: Java

- Lower level of abstraction
- More lines of code
- More development effort is involved
- Code efficiency is high when compared to Pig and Hive.

- SQL like query language.
- Language: HiveQL (declarative SQLish language)
- Ex: select * from ‘mytable’ ;

- Higher level of abstraction
- Comparatively less lines of code than MapReduce and Pig.
- development effort is less.
- Code efficiency is relatively less.
- Hive is used by Analysts, generating daily report.

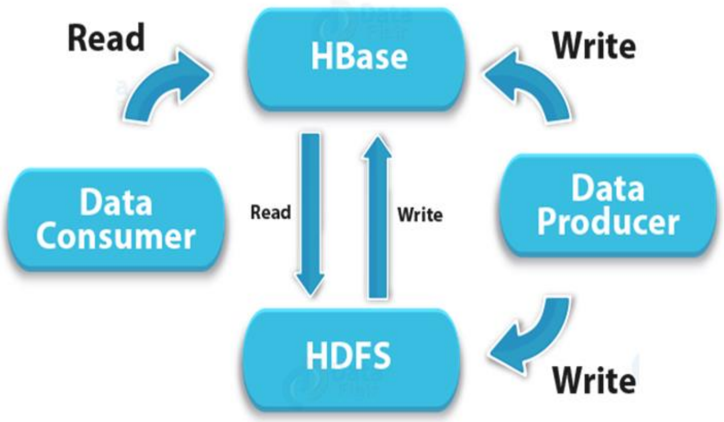
- Scripting Language
- Language: PigLation (Procedural data-flow language)
- Ex: A = load ‘mytable’ ;
-         dump A;

- High level of abstraction
- Comparatively less lines of code than MapReduce
- development effort is less.
- Code efficiency is relatively less.
- Pig is used by Programmers and researchers.

Difference between Pig & Hive

	Pig	Hive
Used by	Mainly used by Researchers and Programmers	Mainly used by Data Analysts
Application	Pig mainly used by Programmers and researchers.	Hive mainly used by Analysts, generating daily report.
Language type	Procedural Data Flow Language	Declarative SQLish Language
Used language	PigLatio Ex: A = load ‘mytable’ ; dump A;	HiveQL Ex: select * from ‘mytable’ ;
Developed by	Yahoo	Facebook
Works on	Structured, semi-structured and Unstructured data	Work on only Structured data.
Avro support	Pig support Avro File format	Hive does not support Avro File format
Partition support	Pig does not support partitions although there is an option for filtering.	Hive support partitions.
Operates on	Operates on the client side of a cluster.	Operates on the server side of a cluster.
Web interface	Pig does not support we interface.	Hive support web interface

Hbase: NoSQL Database



- HBase is an open source **column-oriented non-relational database management system (**NoSQL database) that runs on top of Hadoop Distributed File System (HDFS).
- HBase provides a fault-tolerant way of storing sparse data sets, which are common in many big data use cases.
- Supports all types of data and that is why, it’s capable of handling anything and everything.
- It is written in JAVA and Hbase applications can be written in REST, Avro and Thrift APIs.

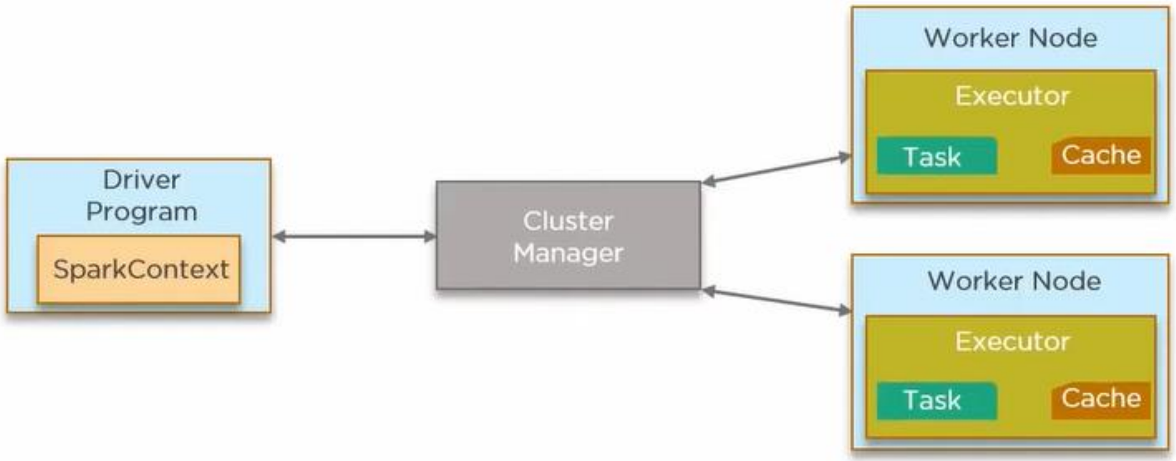
Spark: In-memory Data Processing

- Spark is An open-source framework for real time data analytics in distributed computing environment.
- Written in Scala and was originally developed at the university of California, Berkely.
- It executes in-memory computations to increases speed of data processing over MapReduce.
- 100x faster than Hadoop for large scale data processing by exploiting in-Memory computations. Therefore, it requires high processing power than Map-Reduce

Runs 100x times faster than MapReduce

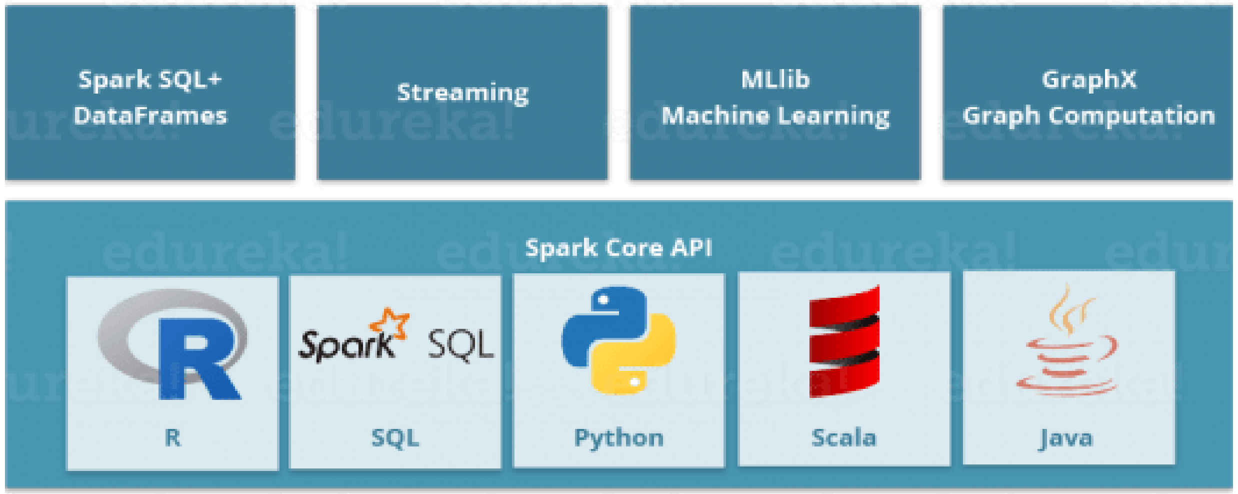
Provides in-memory computation of data

Used to process and analyze real time streaming data such as stock market and banking data



Hadoop v/s Spark

- Spark comes packed with high-level libraries, including support for R, SQL, Python, Scala, Java etc.
- These standard libraries increase the seamless integrations in complex workflow.



"Apache Spark: A Killer or Saviour of Apache Hadoop?" – O'Reily

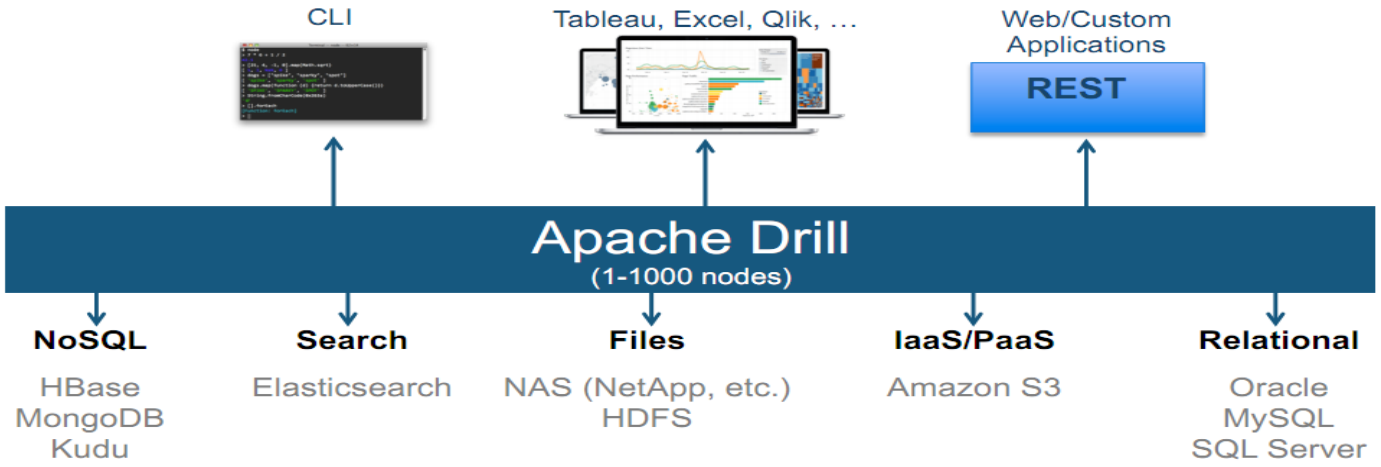
- Answer: This is not an apple-to-apple comparison. Apache Spark best fits for real time processing, whereas Hadoop was designed to store unstructured data and execute batch processing over it. When we combine both, i.e., Apache Spark's ability high processing speed, advance analytics and multiple integration support with Hadoop's low-cost operation on commodity hardware, it gives the best results.
- That is the reason why, Spark and Hadoop are used together by many companies for processing and analyzing their Big Data stored in HDFS.

Apache Drill: SQL on Hadoop

- It’s an open source application which works with distributed environment to analyze large data sets.
- As the name suggests, Apache Drill is used to drill into any kind of data.
- The main power of Apache Drill lies in **combining a variety of data stores just by using a single query**.
- It supports different kinds NoSQL databases and file systems, which is a powerful feature of Drill. For example: Azure Blob Storage, Google Cloud Storage, HBase, MongoDB, MapR-DB HDFS, MapR-FS, Amazon S3, Swift, NAS and local files.
- Apache Drill basically follows the ANSI SQL.
- It is a replica of Google Dremel.

APACHE DRILL

SQL-on-Everything with Apache Drill



Mahout: Machine Learning

- Mahout provides an environment for creating machine learning applications which are scalable.
- Mahout provides a command line to invoke various Machine learning algorithms. It has a predefined set of library which already contains different inbuilt algorithms for different use cases.
- It performs **collaborative filtering, clustering and classification.**

ZooKeeper: Coordinator

- Apache Zookeeper is the coordinator of any Hadoop job which includes a combination of various services in a Hadoop Ecosystem.
- Before Zookeeper, it was very difficult and time consuming to coordinate between different services in Hadoop Ecosystem.
- The services earlier had many problems with interactions like common configuration while synchronizing data. Even if the services are configured, changes in the configurations of the services make it complex and difficult to handle. The grouping and naming was also a time-consuming factor.
- Due to the above problems, Zookeeper was introduced. It saves a lot of time by performing synchronization, configuration maintenance, grouping and naming.

Oozie: Job Scheduler

- Apache Oozie as a clock and alarm service inside Hadoop Ecosystem. For Apache jobs, Oozie has been just like a scheduler. It schedules Hadoop jobs and binds them together as one logical work.
- There are two kinds of Oozie jobs:

1. Oozie workflow: These are sequential set of actions to be executed.
2. Oozie Coordinator: These are the Oozie jobs which are triggered when the data is made available to it.

Apache Flume: Data Ingesting Service

- The Flume is a service which helps in ingesting unstructured and semi-structured data into HDFS.
- It gives us a solution which is reliable and distributed and helps us in collecting, aggregating and moving large amount of data sets.
- It helps us to ingest online streaming data from various sources like network traffic, social media, email messages, log files etc. in HDFS.

- **The flume agent has 3 components: source, sink and channel.**
  1. **Source**: it accepts the data from the incoming streamline and stores the data in the channel.
  2. **Channel**: it acts as the local storage or the primary storage. A Channel is a temporary storage between the source of data and persistent data in the HDFS.
  3. **Sink**: Then, our last component i.e. Sink, collects the data from the channel and commits or writes the data in the HDFS permanently.

Apache Sqoop: Data Ingesting Service

- Sqoop is also data ingesting service.
- The major difference between Flume and Sqoop is that:
  - ✓ Flume only ingests unstructured data or semi-structured data into HDFS.
  - ✓ While Sqoop can import as well as export structured data from RDBMS or Enterprise data warehouses to HDFS or vice versa.

Apache Solr and Lucene

Apache Solr and Apache Lucene are the two services which are used for searching and indexing in Hadoop Ecosystem.
•Apache Lucene is based on Java, which also helps in spell checking.
•If Apache Lucene is the engine, Apache Solr is the car built around it. Solr is a complete application built around Lucene.
•It uses the Lucene Java search library as a core for search and full indexing.

- Core component in a Hadoop Ecosystem for processing.
- Help in writing application that processes large data sets using distributed and parallels algorithms.

Apache Ambari: Cluster Manager

- Ambari is an Apache Software Foundation Project which aims at making Hadoop ecosystem more manageable.
- The Ambari provides software for provisioning, managing and monitoring Apache Hadoop clusters.

- **Hadoop cluster provisioning:**
  - ✓ It gives us step by step process for installing Hadoop services across a number of hosts.
  - ✓ It also handles configuration of Hadoop services over a cluster.
- **Hadoop cluster management:**
  - ✓ It provides a central management service for starting, stopping and re-configuring Hadoop services across the cluster.
- **Hadoop cluster monitoring:**
  - ✓ For monitoring health and status, Ambari provides us a dashboard.
  - ✓ The Amber Alert framework is an alerting service which notifies the user, whenever the attention is needed. For example, if a node goes down or low disk space on a node, etc.

***Learn Fundamentals &
Enjoy Engineering***

Thank You

Prof. Prakash Parmar
Data Engineer | Data Analyst
Assistant Professor
Vidyalankar Institute of Technology, Mumbai
GATE Instructor: Vidyalankar Classes