

# Big Data Analytics

Presented by  
Dr. Amany AbdElSamea



# Outline

- What is Big Data?
- Big Data Characteristics
- Types of Big Data
- Data Analytics Lifecycle
- Big Data Tools
- Apache Big Data Projects
- Hadoop Ecosystem

# What is Big Data?

- Collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.
- The scale, diversity, and complexity of data require new architecture, techniques, algorithms, and analytics to manage and extract value and hidden knowledge from it.



# Why Big Data?

Key enablers for the growth of “Big Data” are

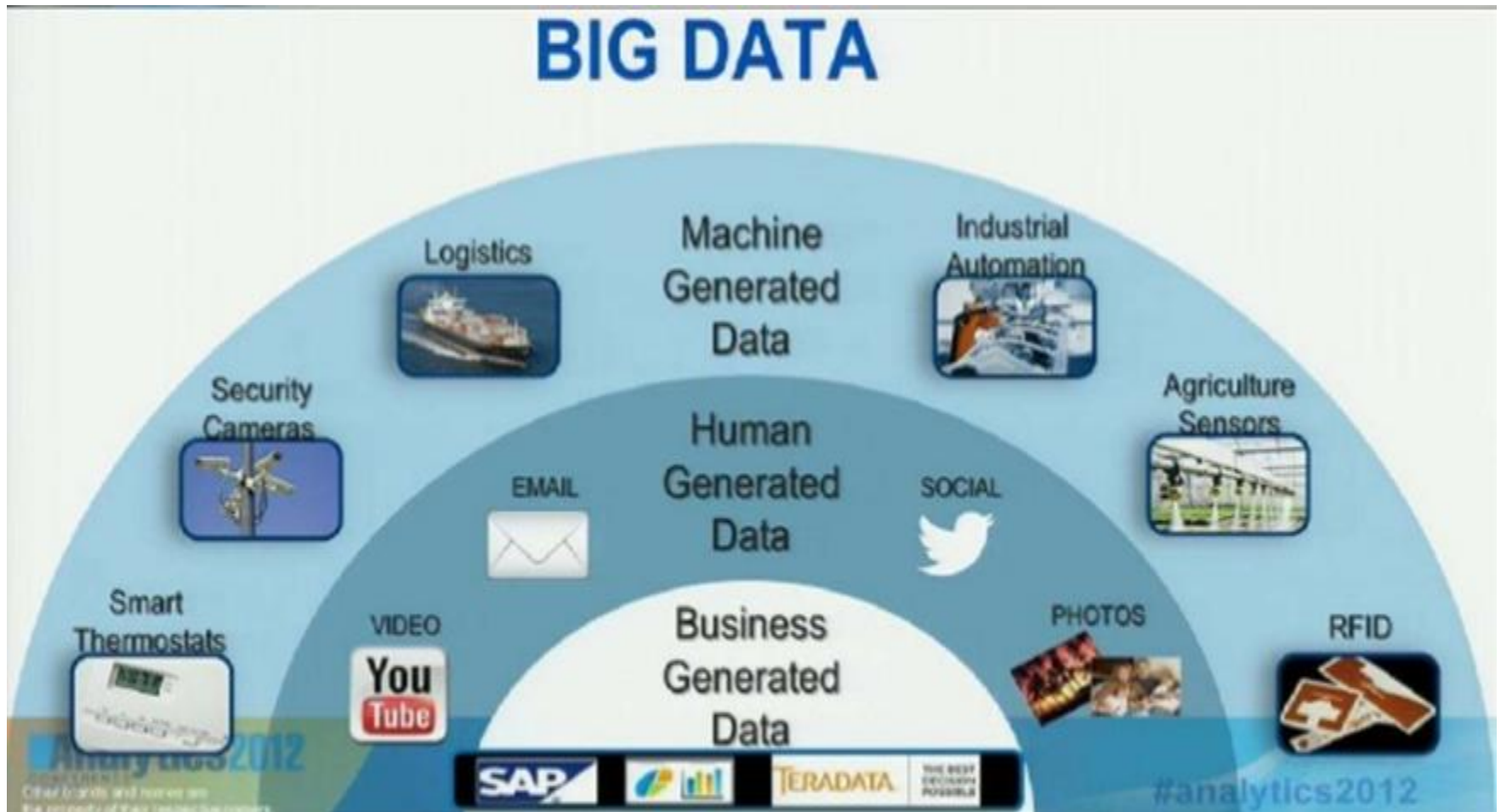
```
graph TD; A[Increase of storage capacities] --- B[Increase of processing power]; B --- C[Availability of data];
```

Increase of storage capacities

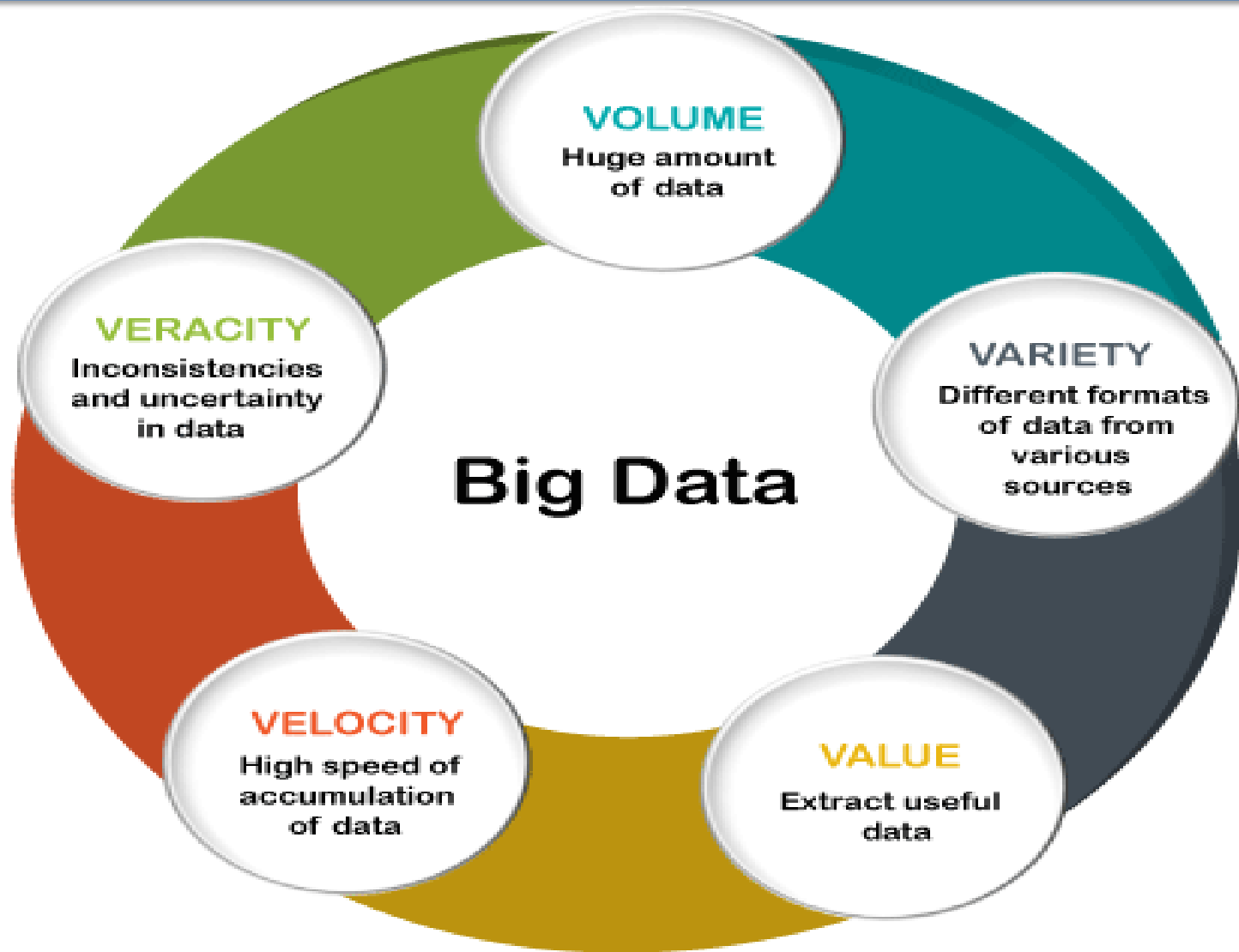
Increase of processing power

Availability of data

# Big Data Sources



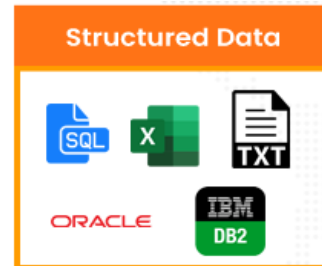
# Big Data Characteristics



# Types of Big Data

1. **Structured data:** Any data that can be processed, easily accessible, and can be stored in a fixed format is called structured data. Data of a well-defined data type, format, or structure

Examples: Relational database tables and CSV files



2. **Unstructured data:** Data that has no inherent structure. Unstructured data in Big Data is where the data format constitutes multitudes of unstructured files (images, audio, log, and video).

Examples: Text documents, images, and video



3. **Semi-structured data:** In Big Data, semi-structured data is a combination of both unstructured and structured types of data. This form of data constitutes the features of structured data but has unstructured information that does not adhere to any formal structure of data models or any relational database. Some semi-structured data examples include XML and JSON.



# Big Data Job Roles

Key roles for a successful analytics project

Business user



Project sponsor



Project manager



Business intelligence analyst



Database administrator (DBA)



Data engineer



Data scientist











# Key Roles cont.,

- **Business user:** Someone who benefits from the end results and can advise the project team on the value of end results and how the project results will be operationalized.
- **Project sponsor:** The project sponsor generally provides the funding and gauges the degree of value from the final outputs of the working team.
- **Project manager:** Ensures that key milestones and objectives are met on time and at the expected quality.
- **Business intelligence analyst:** Provides business-domain expertise with deep understanding of the data, KPIs, key metrics, and analytics from a reporting perspective.
- **Data engineer:** Applies deep technical skills to assist with data extraction from source systems and data ingestion on the analytic sandbox.
- **Database administrator (DBA):** Provisions and configures the database environment to support the analytical needs of the working team.
- **Data scientist:** Provides technical expertise for analytical techniques and data modeling, and applies the proper analytical techniques to given business problems to achieve the overall analytical objectives.

# Data Analytics Life Cycle

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
					
<b>Business Issue Understanding</b>	<b>Data Understanding</b>	<b>Data Preparation</b>	<b>Exploratory Analysis and Modeling</b>	<b>Validation</b>	<b>Visualization and Presentation</b>
Define business objectives	Collect initial data	Gather data from multiple sources	Develop methodology	Evaluate results	Communicate results
Gather required information	Identify data requirements	Cleanse	Determine important variables	Review process	Determine best method to present insights based on analysis and audience
Determine appropriate analysis method	Determine data availability	Format	Build model	Determine next steps	Craft a compelling story
Clarify scope of work	Explore data and characteristics	Blend	Assess model	Results are valid → proceed to step 6	Make recommendations
Identify deliverables		Sample		Results are invalid ← revisit steps 1-4	

# Data Repositories

A data repository is a data storage entity in which data has been isolated for analytical or reporting purposes.

- **Data Warehouse:**

A data warehouse is a centralized repository that stores large volumes of data from multiple sources in order to more efficiently organize, analyze, and report on it. Unlike a data mart and lake, it covers multiple subjects and is already filtered, cleaned, and defined for a specific use.

- **Data Mart:**

A data mart is a subset of a data warehouse designed to deliver specific data to a specific user for a specific application. This type of repository is focused on a single subject. For example, a human resources database may contain data marts for employees, benefits, and payroll, respectively.

- **Data Lake**

A data lake stores raw data from different sources. “Raw” data means it has not been filtered or structured and it does not have a predetermined use case. This makes it easier and less expensive to edit, but also requires more work selecting, organizing, and cleaning it to use it.

# Data warehouse vs. Data lake vs. Data mart

	Data warehouses	Data lakes	Data marts
Usage	The data analysis and reporting needs of an entire organization	The reporting needs of different kinds and difficulty, predictive analytics	The reporting needs of a specific operational department or subject
Data stored (typically)	Larger volumes of structured data; processed	Huge volumes of structured and unstructured data; raw	A limited amount of structured data; processed
Data sources	An array of external and internal sources, covering different areas of business	Any external or internal sources	Few sources linked to one business area
Size	Larger than 100 GB	Larger than 100 GB	Smaller than 100 GB
Ease of creation	Difficult to set up	Difficult to set up	Easy to set up

# Big Data Tools

## Data Stores



## Data Processing Layer



## Stream Processing



## Data Presentation Layer



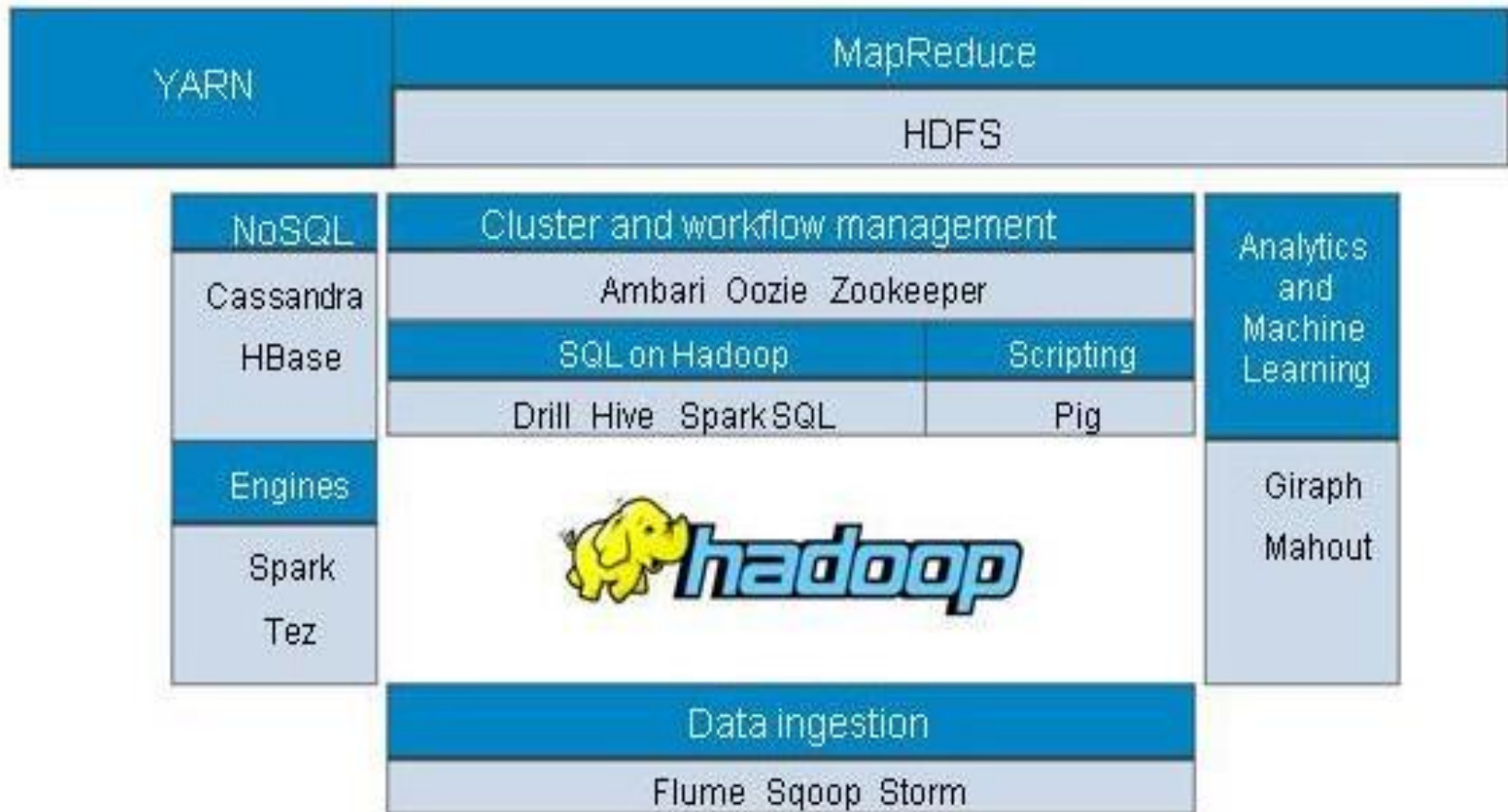
## Strategic/Predictive Analytics



## Data Visualization



# Apache Big Data Projects



# Apache Hadoop



- Apache Hadoop project develops open-source software for reliable, scalable, distributed computing
- The Apache Hadoop software library is a framework that allows for the distributed processing of large datasets across clusters of computers using simple programming models.
- This library enables us to use parallel processing capability to handle huge volumes of data using flexible infrastructure
- Hadoop is written in Java
- **To summarize , Hadoop offers:**
  - A scalable, flexible, and reliable distributed computing big data framework for a cluster of systems.
  - It provides massive data storage facility, enormous computational power and flexibility to collect, process, and analyze data
  - Hadoop handles different types of data such as structured, unstructured and semi-structured data
  - Hadoop is not a database but simply a data warehouse tool



# Key Components of Hadoop

MapReduce

Data Processing

YARN

Resource Management

HDFS

Storage, File System

Hadoop Common

- Map Reduce
- Yet Another Resource Negotiator (YARN)
- Hadoop Distributed File System (HDFS)
- Hadoop Common module is a Hadoop Base API- a jar file- for all Hadoop components. All other components work on top of this module

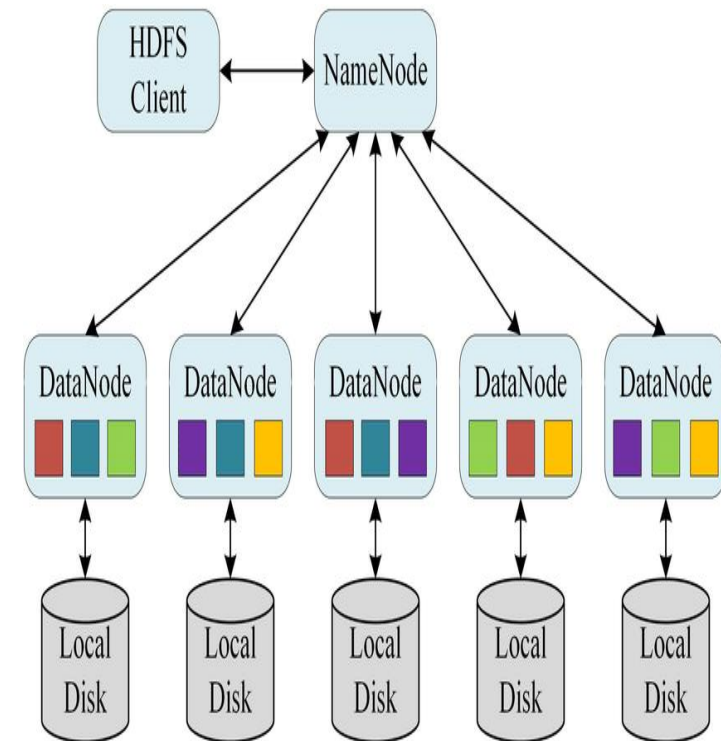


# Hadoop Distributed File System

- Distributed file system designed to run on commodity hardware for storing large files of data with streaming data access patterns
- Highly fault tolerant
- **Default storage for the Hadoop cluster**
- File system namespaces. Similar to most other existing file systems; one can create and remove files, move a file from one directory to another, or rename a file.
- Data/File on HDFS is stored in chunks (128 MB default) called blocks

# HDFS Architecture

- HDFS has a master/slave architecture.
- An HDFS cluster consists of a multiple NameNode that manages the file system namespace and regulates access to files by clients.
- Further, some DataNodes, usually one per node in the cluster, manage storage attached to the nodes that they run on. The DataNodes are used as common storage for blocks by all the NameNodes.
- Each DataNode registers with all the NameNodes in the cluster. DataNodes send periodic heartbeats and block reports. They also handle commands from the NameNodes. HDFS exposes a file system namespace and allows user data to be stored in files.



# Name Node

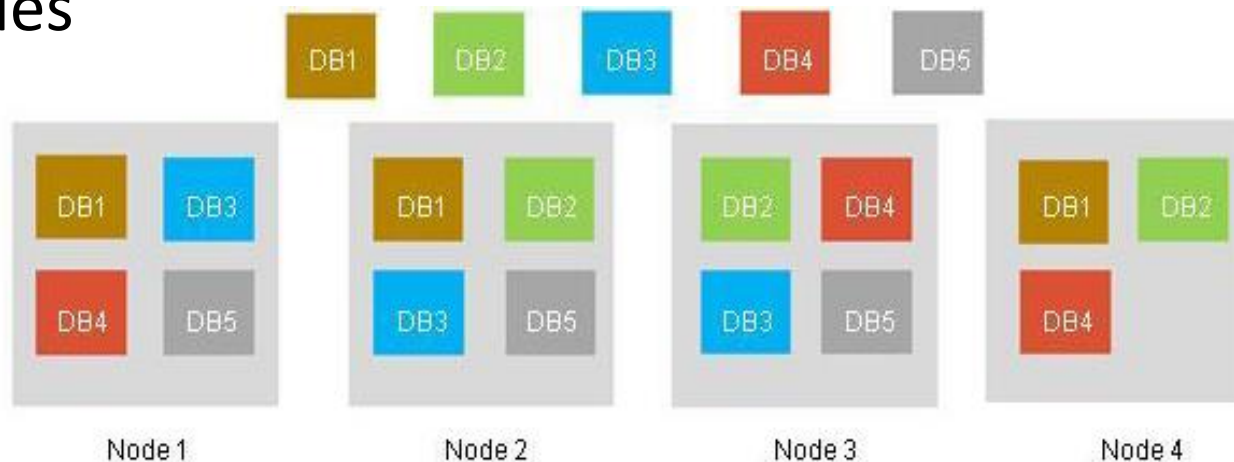
- A master server that manages the file system namespace and regulates access to clients.
- **Tasks of HDFS Name Node:**
  - Manage the file system namespace.
  - Regulate the client's access to files.
  - Execute the file system execution such as naming, closing, and opening files and directories.
- Information is stored persistently on the disk in the form of two files: **namespace image** and **edit log**.
  - Namespace image file contains the Inodes and the list of blocks which define the metadata.
  - Edit log contains any modifications that have been performed on the content of the image file.

# Data Node

- A file is split into one or more blocks, and these blocks are stored in multiple Data Nodes
- **Tasks of HDFS Data Node:**
  - Responsible for serving read and write requests from the file system clients
  - Performs operations such as block replica creation, deletion, and replication according to the instruction of NameNode.
  - Manage data storage of the system.
  - Perform CPU-intensive jobs such as semantic and language analysis, statistics and machine learning tasks, as well as I/O intensive jobs including clustering, data import, data export, search, decompression, and indexing.
  - They report back to NameNode with the list of blocks they are storing.
  - Bringing computation to data is often more efficient than the reverse.

# Block replication

- Data is replicated more than once in a Hadoop cluster for fault tolerance and availability
- Every block of data is replicated on more than one node so, even if a node fails, the data is available on another node
- The replication factor is the number of times a block is replicated. The default is 3 for HDFS, which means every block is replicated three times on three different nodes



# YARN

- Yet Another Resource Negotiator (YARN) is a Hadoop ecosystem component that provides the resource management.
- YARN is called as the operating system of Hadoop, as it is responsible for managing and monitoring workloads
- It allows multiple data processing engines such as real-time streaming and batch processing to handle data stored on a single platform.

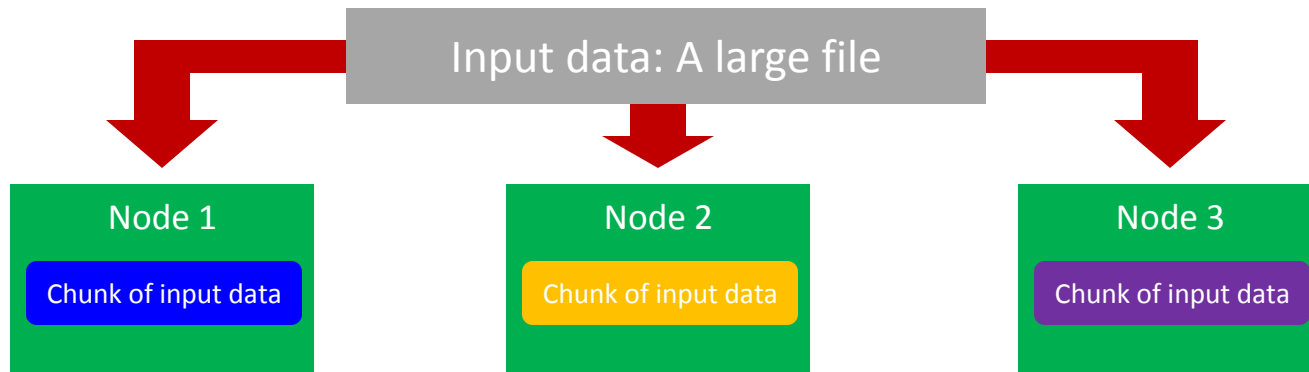
# Map Reduce

- A software paradigm for writing applications that process vast amounts of data, multi-terabyte datasets, in-parallel on large clusters- thousands of nodes- of commodity hardware in a reliable, fault-tolerant manner.
- Java-based programming paradigm
- A combination of the Map and Reduce models that can be applied to wide variety of business cases.
- Handles scheduling and fault tolerance
- Used in problems that are “embarrassingly parallel”

Example: Word Count

# Data Distribution

- In a MapReduce cluster, data is distributed to all the nodes of the cluster as it is being loaded in
- An underlying distributed file systems (e.g., GFS) splits large data files into chunks which are managed by different nodes in the cluster

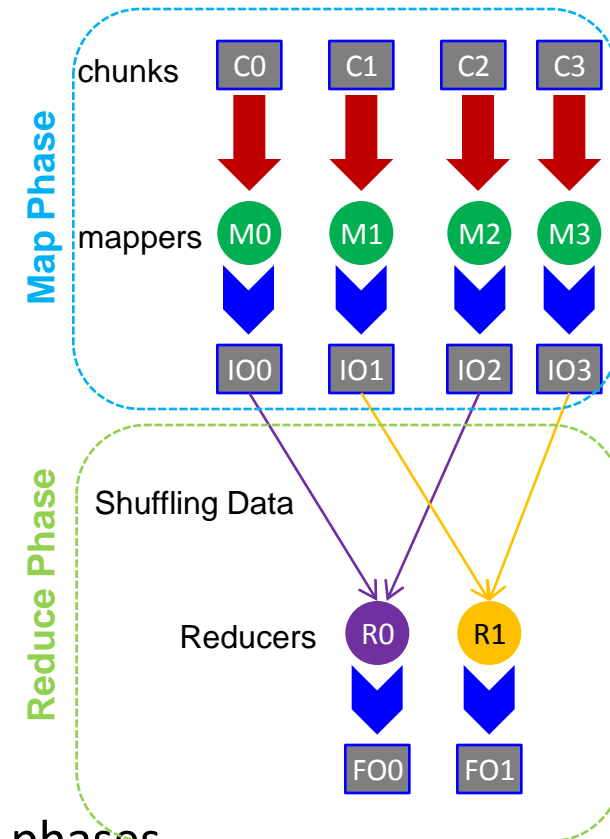


- Even though the file chunks are distributed across several machines, they form *a single namespace*



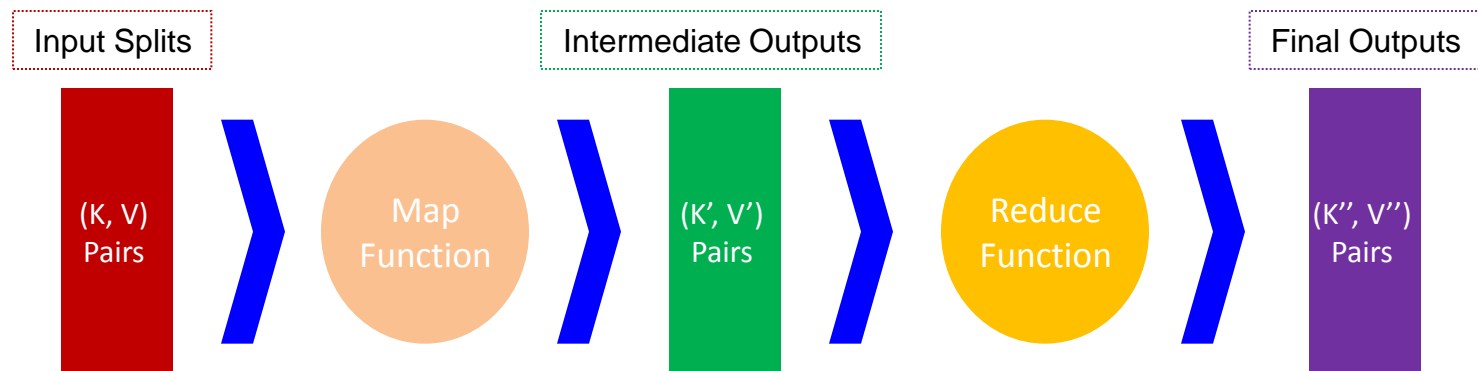
# MapReduce Steps

- In MapReduce, chunks are processed in isolation by tasks called *Mappers*
- The outputs from the mappers are denoted as intermediate outputs (IOs) and are brought into a second set of tasks called *Reducers*
- The process of bringing together IOs into a set of Reducers is known as *shuffling process*
- The Reducers produce the final outputs (FOs)
- Overall, MapReduce breaks the data flow into two phases, *map phase* and *reduce phase*



# Keys and Values

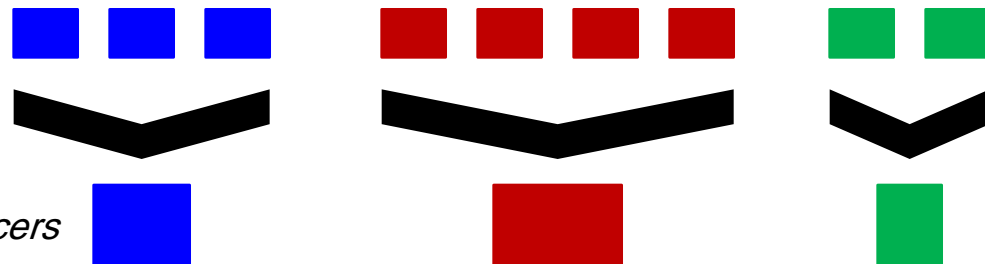
- The programmer in MapReduce has to specify two functions, the *map function* and the *reduce function* that implement the Mapper and the Reducer in a MapReduce program
- In MapReduce data elements are always structured as key-value (i.e., (K, V)) pairs
- The map and reduce functions receive and *emit* (K, V) pairs



# Partitions

- In MapReduce, intermediate output values are not usually reduced together
- *All values with the same key are presented to a single Reducer together*
- More specifically, a different subset of intermediate key space is assigned to each Reducer
- These subsets are known as *partitions*

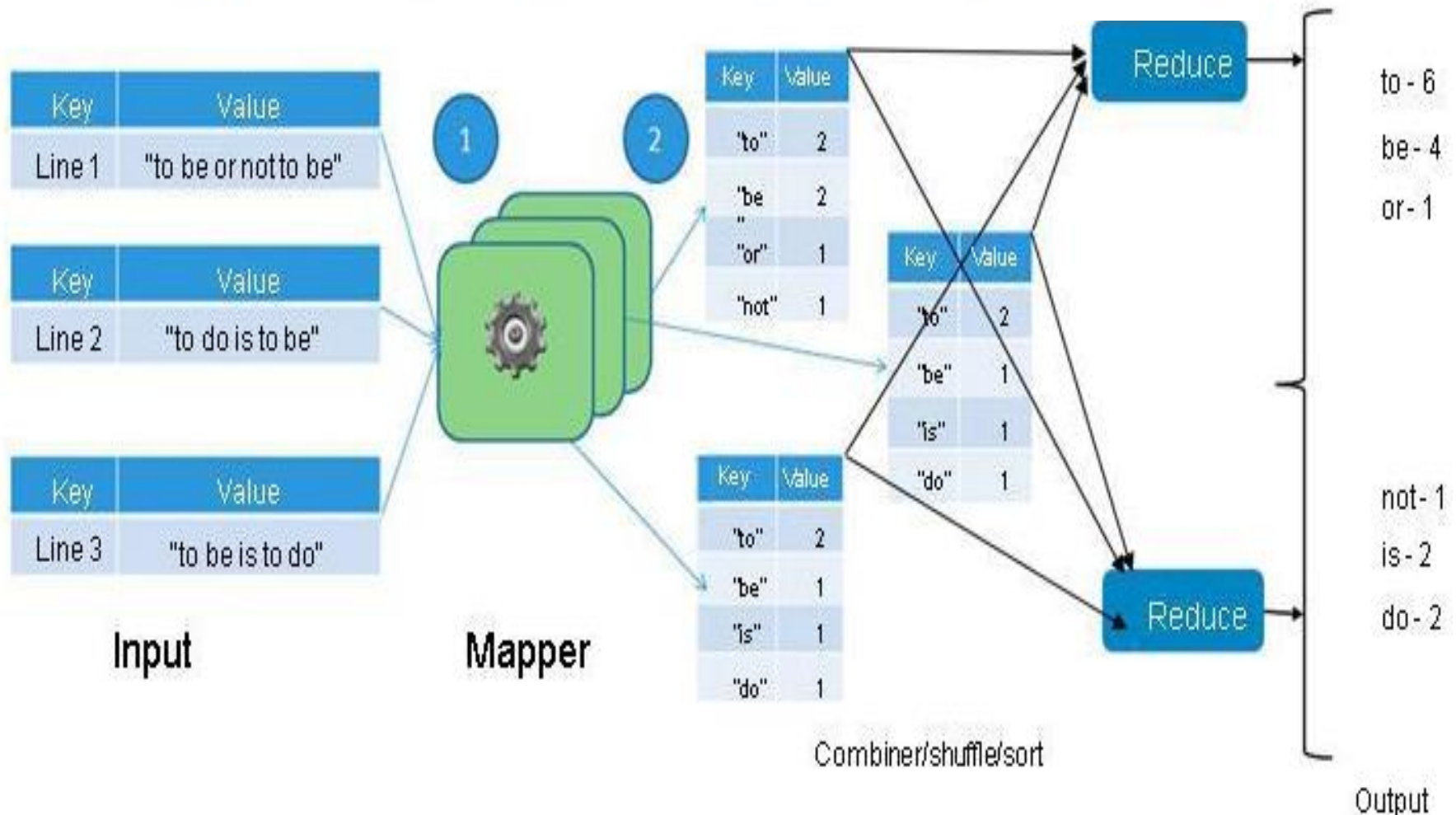
*Different colors represent different keys (potentially) from different Mappers*



*Partitions are the input to Reducers*

# MapReduce-Count Words Example

MapReduce—count words in document



# Questions