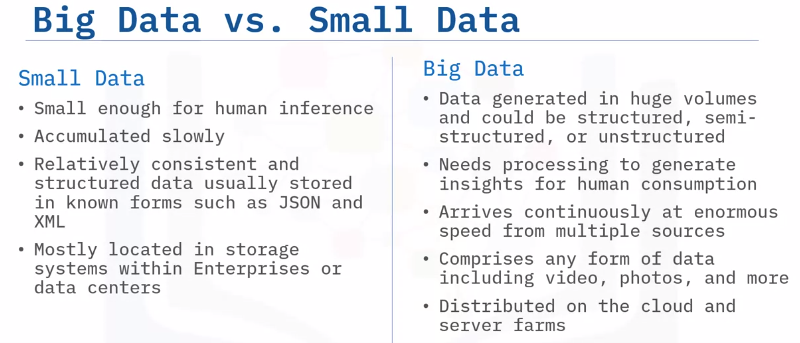
# Introduction to Big Data

**Learning Objectives**

* Define what Big Data is and identify key Big Data characteristics.
* Provide examples of Big Data related technologies, describe the impact of Big Data on businesses and people, and define the relationship between the Internet of Things (IoT) and Big Data.
* Compare linear to parallel processing and explain why Big Data requires parallel processing.
* Describe scalability, including horizontal scaling, embarrassing parallel calculations, and fault tolerance as they relate to Big Data.
* List the key Big Data ecosystem tooling categories and their associated major tools and vendors.
* Explain the role of open-source in Big Data and list related platforms and open-source frameworks.
* Describe the key sources and different types of Big Data.
* Provide examples of real-world Big Data use cases.

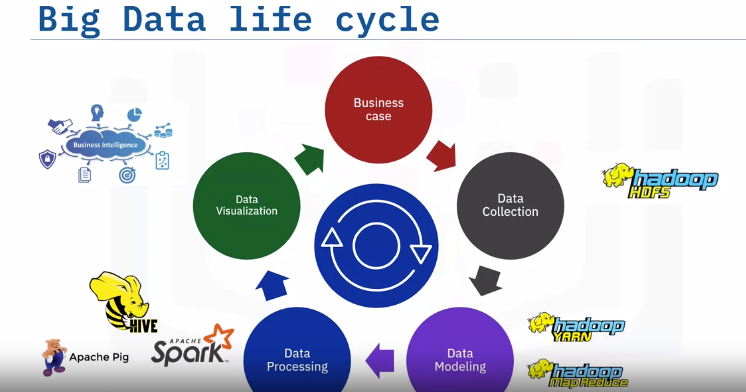
Introduction:

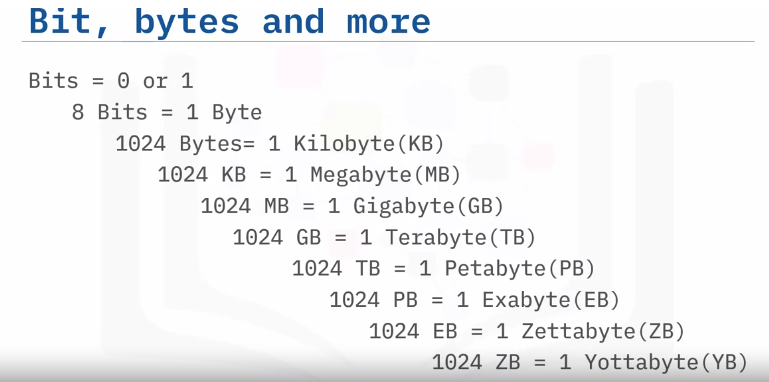
The latest statistics report that the accumulated world’s data will grow from 4.4 zettabytes to 44 zettabytes, with much of that data classified as Big Data. Revenues based on Big Data analytics are projected to increase to $103 billion by 2027. Understandably, organizations across industries want to harness the competitive advantages of Big Data analytics. This course provides us with the foundational knowledge and hands-on lab experience we need to understand what Big Data is and learn how organizations use Apache Hadoop, Apache Spark, including Apache Spark SQL, and Kubernetes to expedite and optimize Big Data processing.



It is a common misconception that Big Data refers to just large volumes of data. In reality, Big Data is the entire life cycle of working with large volumes of data. Let’s take a look at each phase in the Big Data life cycle. Big Data collection is initiated as a result of a business problem or requirement. As data is collected, it gets stored using a framework for distributed storage such as Hadoop HDFS.

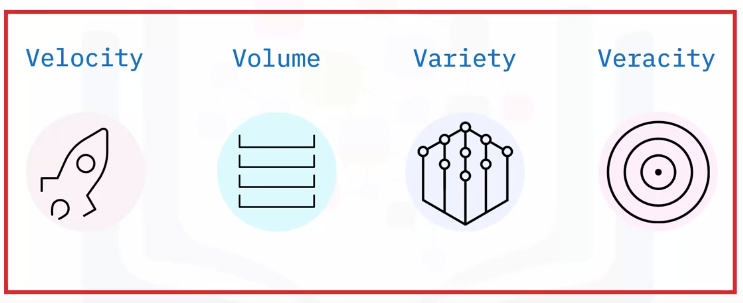
To make sense of all the data collected, Map and Reduce tasks and scripts create a data model to store it in a database. This data model includes the various data entities (or objects), and the relationship and rules between these entities. After modeling, data is ready to be processed. Tools such as Apache Spark are used to produce meaningful information from the modeled data. Finally, the processed data is visualized and presented in a graphical format such as charts and graphs. This visualized data is then used for making meaningful business decisions and lead to new business cases, thereby creating a continuous life cycle.





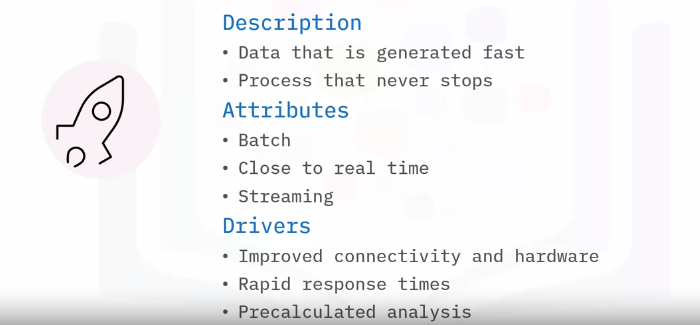
# Four V’s:

When we talk about Big Data, we traditionally talk about the four V’s of Big Data. These are: Velocity, Volume, Variety, and Veracity. Velocity is the speed at which data arrives. Volume is the increase in the amount of data stored over time. Variety is the diversity of data. Many forms of data exist and need to be stored. Veracity is the certainty of data. With a large amount of data available, how will we know if the data collected is accurate or inaccurate? These four main components are used to describe the dimensions of Big Data.



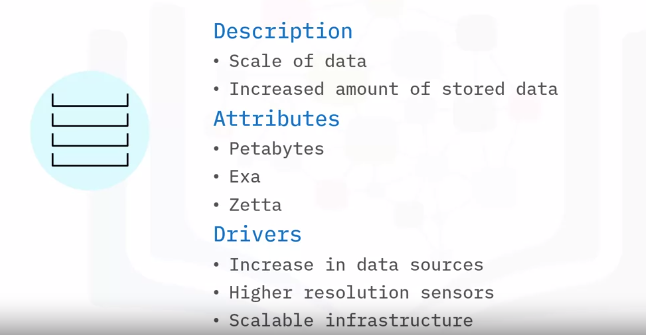
## Velocity:

Velocity signifies that data is being generated extremely fast, and the process never stops. Data must be processed quickly so that decisions can be made at the speed with which the data arrives. Velocity’s main attributes are: batch close to real-time, and streaming. What are the drivers? Definitely improving connectivity and hardware. Just think about all the devices that are connected through the Internet today and all the super-fast response times. Big Data also supports upscaling of pre-calculated analysis.



## Volume:

Volume is the increase in the amount of data stored. The amount of Big Data generated is vast compared to traditional data sources. Volume attributes of Big Data are: Petabytes, Exa, and Zetta, to name just a few. Typical drivers of volume in Big Data are: the increase in data sources, higher resolution sensors, and scalable hardware infrastructure.



## Variety:

Variety is the diversity of the data. Data is generated by people and processes through the use of machines, from both inside and outside an organization. Some of the data is structured and semi-structured, but most is unstructured. The main attributes are structure, complexity, and origin. Drivers of Variety in Big Data can be: mobile technologies, scalable hardware infrastructure, resilience, fault recovery, and efficient storage and retrieval.

## Veracity:

Veracity is the quality, origin, and conformity to facts and accuracy of the data. This is because data comes from both within and outside an organization. Attributes include consistency and completeness, integrity, and ambiguity. Drivers of Veracity in Big Data are: cost and the need for traceability, robust ingestion, and extract, transform, load (ETL) mechanisms.



Big Data has another V that must be considered. The fifth V of Big Data is Value. It is the outcome of making intelligent business decisions from leveraging the previous four V’s. The ultimate goal of an organization is to: produce value in the form of faster and smarter business decisions, increase efficient use of resources, and discover of new market opportunities. Big Data supports innovation and thus creates value.

**Summary:**

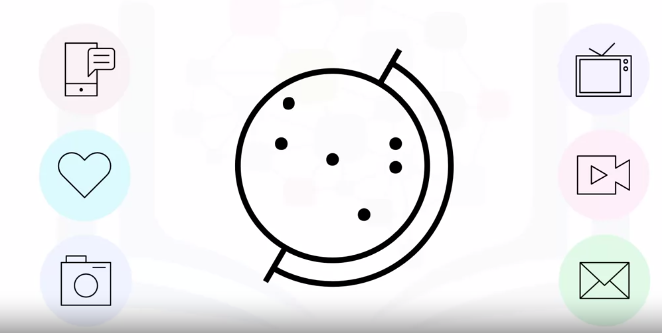
Big Data is the digital trace that gets generated in this digital era. Big Data is a high-volume, high-velocity, and/or high-variety information asset that demands cost-effective and innovative tools for processing. The core features of Big Data are the 4 V’s: Velocity, Volume, Variety, and Veracity. Big Data creates a fifth V, Value, when collected, stored, and processed correctly.

# Impact of Big Data:

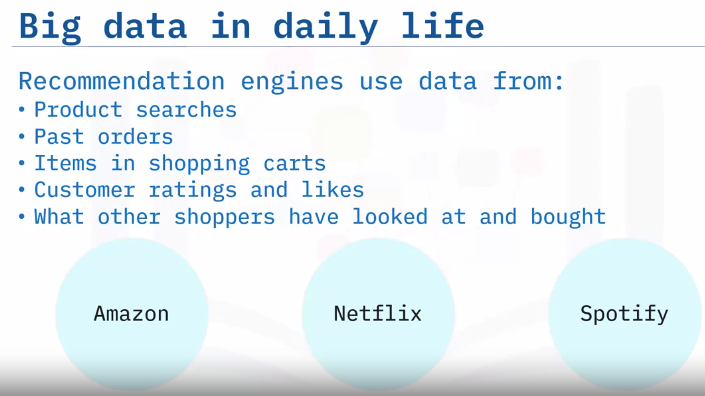
## Learning Objectives:

* List examples of Big Data related technologies
* Explain the impact of Big Data on businesses and people
* Describe the Internet of Things (IoT) and its impact on Big Data

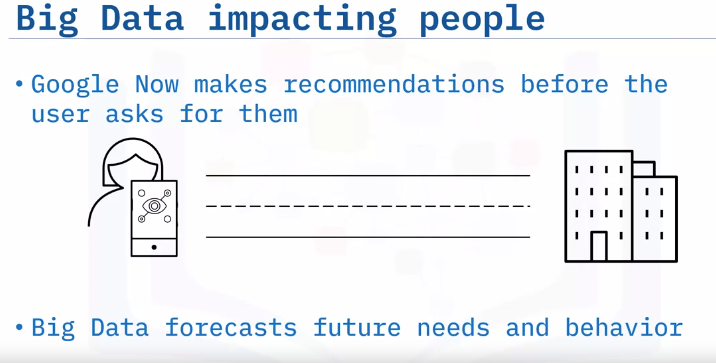
## Generating and using Big Data:



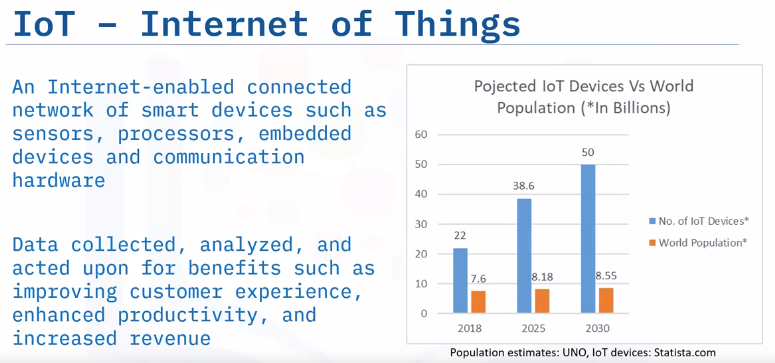
Big Data is one of the most important subjects of this century. Yet globally, millions of people are generating and using Big Data without necessarily being aware of it. This personal data in the form of photos, videos, and text that people send to each other forms the bulk of data collected by consumer goods companies. So, it is interesting to consider how Big Data is impacting businesses and people.



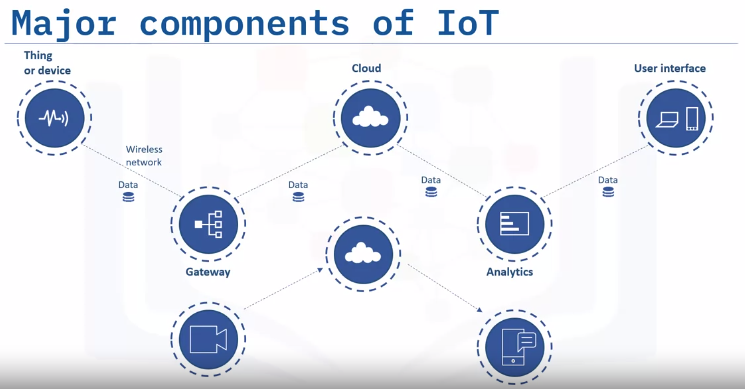
Have you ever searched for or bought a product on Amazon? Did you notice that Amazon’s recommendations are based on what the users have searched for, have bought in the past, the items they have in the virtual shopping cart, items they have rated and liked, and what other customers have viewed and purchased. Companies like Amazon, Netflix and Spotify use algorithms based on big data to make specific recommendations based on customer preferences and historical behavior.



## IoT:



The Internet of Things (IoT) refers to a system of physical objects that are connected through the Internet. A “thing or device” can include a smart device in our homes, or a personal communication device such as a smartphone or computer. These collect and transfer massive amounts of data over the Internet without any manual intervention by using embedded technologies All of these devices gather, analyze, share, and transmit data in real time. Without Big Data, IoT devices would not hold the functionalities and capabilities which have gained them so much attention globally. Following Statista’s forecast, we will see four times more IoT devices than the total number of the entire world population by 2030.

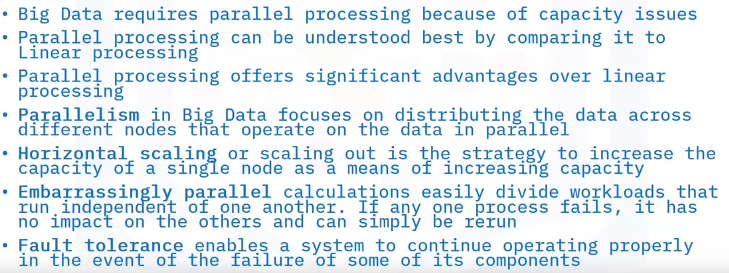


Let’s look at major components of the IoT ecosystem. An Internet-enabled connected network of systems, that comprise smart devices and sensors, constantly produce and transmit a variety of data. This Big Data is passed through a Gateway that routes it to its destined storage in the Cloud. Data Engineers use sophisticated algorithms to make sense of this collected Big Data and compile it into an analytical form that can be interpreted by humans. The analyzed data is then used to make business and consumer decisions to enhance the customer experience and to enhance productivity. For example, a security camera could continually upload footage, and a cloud-based big data algorithm may alert the user if any activity is detected.

**Summary:**

Big Data is everywhere and is being collected and used to drive business decisions and influence people’s lives. Virtual personal assistants like Siri on Apple devices, or Alexa on Amazon devices, use Big Data to devise answers to the infinite number of questions end-users may ask. Google Now impacts people through using Big Data to forecast future needs and behavior. Internet of Things (IoT) devices continually generate massive volumes of data; and Big Data analytics helps companies gain insights from the data collected by IoT devices.

# Parallel Processing Scaling and Data Parallelism:

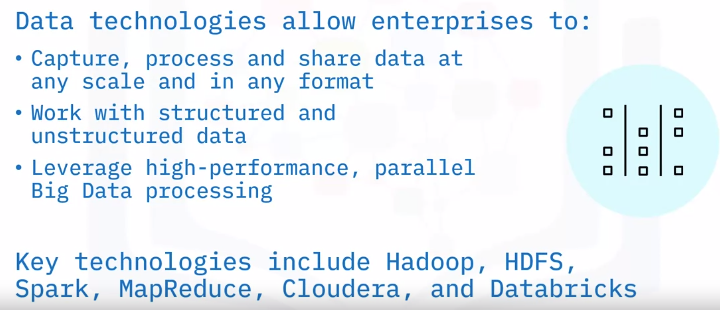


# Big Data Tools and Ecosystem:

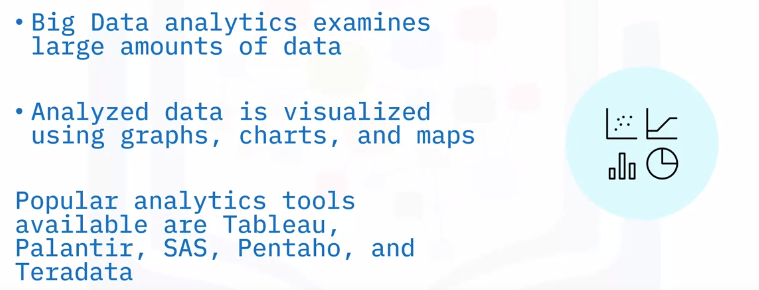
Big Data tools have six different categories.

* Data Technologies
* Analytics and Visualization
* Business Intelligence
* Cloud service providers
* NoSQL Databases; and
* Programming Tools.

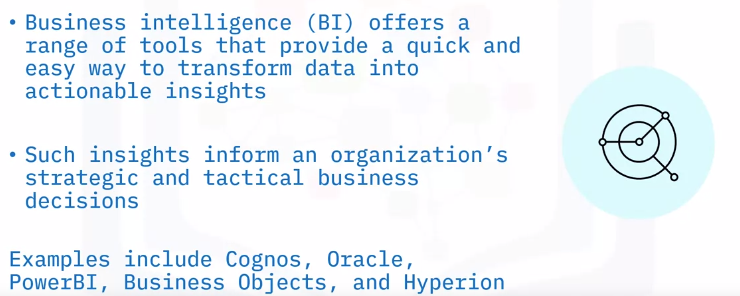
## Data Technologies:



## Analytics and Visualization:



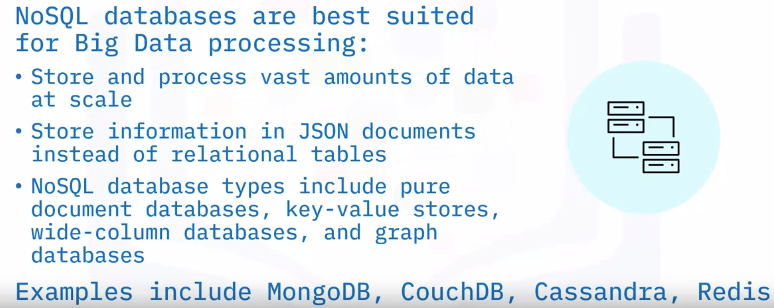
## Business Intelligence:



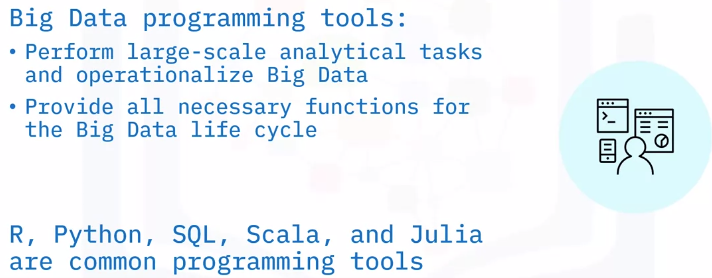
## Cloud Providers:

Big Data presents challenges to storing, transporting, processing, mining, and serving enormous amounts of data. Cloud providers offer fundamental infrastructure and support with shared computing resources including computing, storage, networking, and analytical software. Cloud providers also provide “software as a service” models with point solutions to allow enterprises to easily aggregate, process, and visualize data. Examples of Cloud service providers include AWS, IBM, GCP, and Oracle.

## NoSQL Databases:

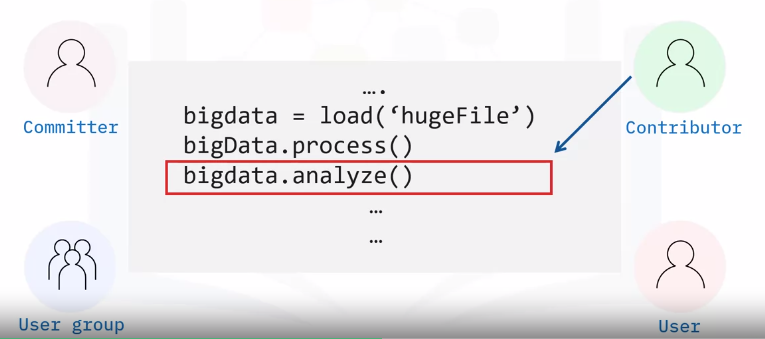


## Programming Tool:



# Open Source and Big Data:

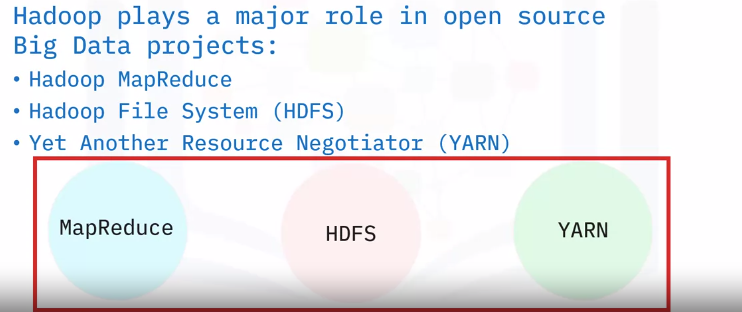
Most open-source projects have formal processes for contributing to code, and also include various levels of influence and obligation to the project. Committer, contributor, user, user group. Typically, committers have the ability to modify the code directly, while contributors submit their code for review (by a committer) before the code is modified. Many more people are simply users of the code.



## Open Source Platforms:

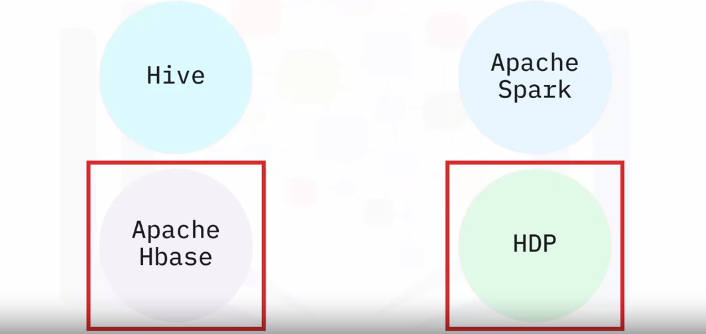


## Open Source in Big Data:



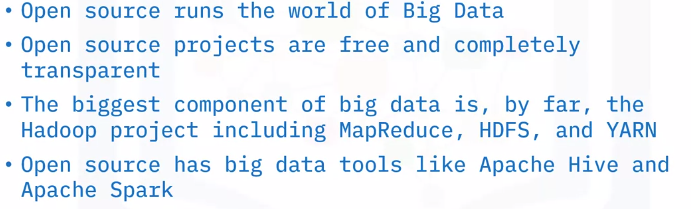
The biggest component of big data is, by far, the Hadoop project and its three main components: MapReduce, The Hadoop File System (HDFS), and The resource manager (YARN). MapReduce is a framework that allows code to be written to run at scale on a Hadoop cluster. It is still used, but not as much as more modern Big Data computation frameworks like Apache Spark. HDFS is the file system that stores and manages Big Data files. It manages all of the issues around large and distributed datasets, including resilience and partitioning. It is still a mainstay of the industry. 70% of the world’s Big Data resides on HDFS. More modern approaches to distributed storage, such as S3 and object storage, are coming into use, but they are based on the design principles of HDFS. YARN is the resource manager that comes with Hadoop, and it is the default resource manager for many Big Data applications, including HIVE and Spark. It is one of the most robust resource managers in use today, but more modern container-based resource managers (like Kubernetes) are slowly becoming the new de facto standards. Concluding, these are the main components of the Hadoop ecosystem, and most big data applications are built on top of them.

## Open Source tools in Big Data:

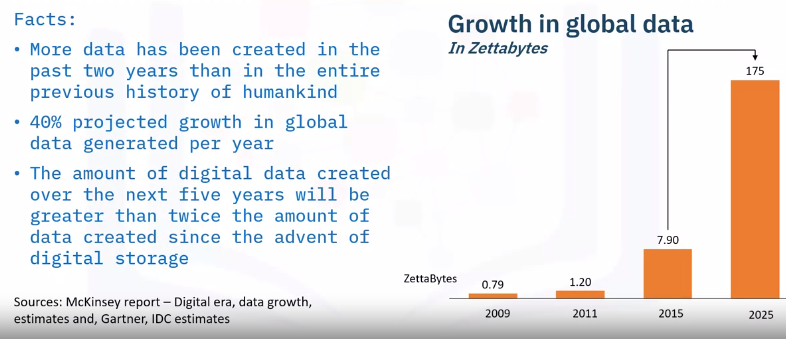


The array of big data applications available to the user is dizzying. They all build upon the basic Hadoop framework or interact with it in some way, however. Frameworks like Hive and Spark support lots of ETL (Extract, Transform, Load) and computation tasks on Hadoop systems. Some systems that integrate tightly with the Hadoop ecosystem are: Apache Hbase, which is a large NoSQL datastore. It manages storage and computation resources outside of the Hadoop ecosystem but often resides on the same cluster. Open-source packages like the Hortonworks Data Platform (HDP) provide a set of big data tools that are already configured to work together, and include most of the important open source packages (Hadoop, Spark, Hive, Hbase, and others).

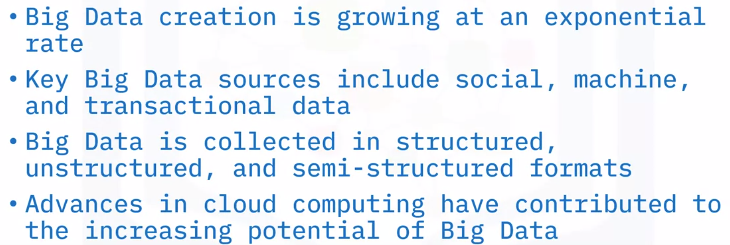
**Summary:**



# The hype about Big Data:



The chart on the right shows the growth in global data in zettabytes.



# Big Data Use Cases:

* Big data is quickly becoming the differentiator for success across industries
* Industries such as retail, insurance, telecom, manufacturing, automotive, and finance are leveraging Big Data to drive customer satisfaction and make competitive business decisions.

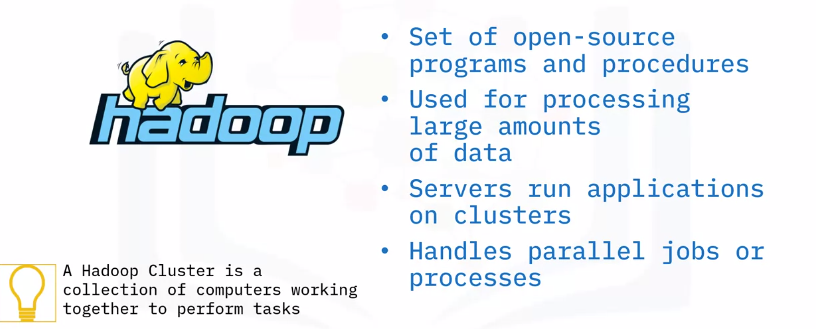
# Introduction to Big Data with Spark and Hadoop:

## **Learning Objectives**

* Define Hadoop, review its history, including its relationship with Big Data processing and some of its challenges.
* Explain MapReduce, including its applications, components, and use cases.
* List the stages of the Hadoop ecosystem and identify some of tools used in each stage.
* Differentiate between Hadoop’s core and extended components.
* Explain the Hadoop Distributed File System (HDFS), identify its key features, and describe its architecture.
* Describe Hive’s uses, features, and architecture.
* Compare and contrast HBase to HDFS.
* Describe HBase and its features, uses, architecture, and concepts.
* Compare and contrast Hive to traditional relational database management systems.
* Run a single-node Hadoop instance.
* Perform a word count by using Hadoop Map Reduce.

## I**ntroduction to Hadoop:**

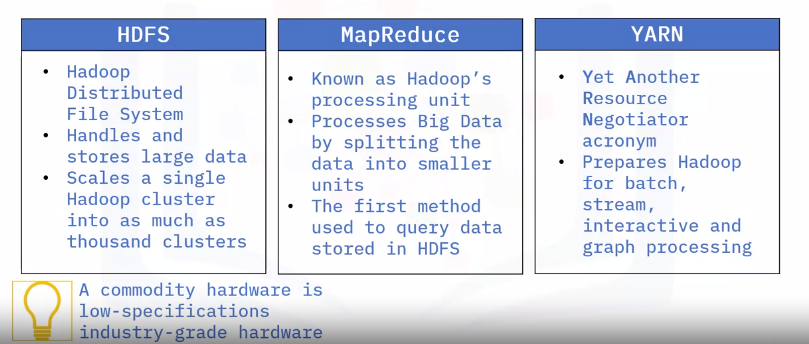
Hadoop is a set of open-source programs and procedures which can be used as the framework for Big Data operations. It is used for processing massive data in distributed file systems that are linked together. It allows for running applications on clusters. A cluster is a collection of computers working together at the same to time to perform tasks. It should be noted that Hadoop is not a database but an ecosystem that can handle processes and jobs in parallel or concurrently.



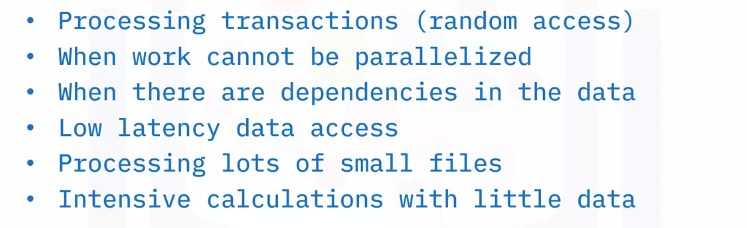
Hadoop is optimized to handle massive quantities of data which could be: Structured, tabular data, Unstructured data, such as images and videos, or Semi-structured data, using relatively inexpensive computers.

## **How does Hadoop work?**

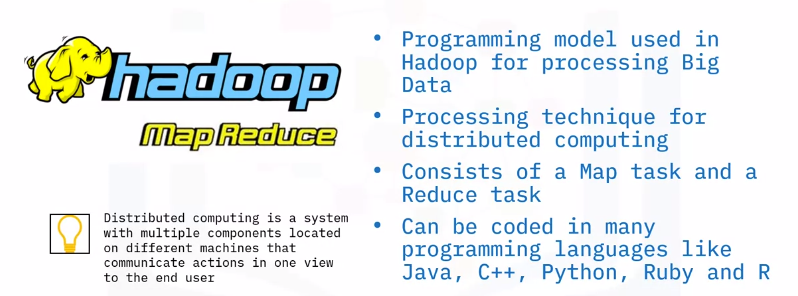
Hadoop has individual components for storing and processing data. The term Hadoop is often used to refer to both the core components of Hadoop as well as the ecosystem of related projects. The core components of Hadoop include: Hadoop Common, which is an essential part of the Apache Hadoop Framework that refers to the collection of common utilities and libraries that support other Hadoop modules. There is a storage component called Hadoop Distributed File System, or HDFS. It handles large data sets running on commodity hardware. A commodity hardware is low-specifications industry-grade hardware and scales a single Hadoop cluster to hundreds and even thousands. The next component is MapReduce which is a processing unit of Hadoop and an important core component to the Hadoop framework. MapReduce processes data by splitting large amounts of data into smaller units and processes them simultaneously. For a while, MapReduce was the only way to access the data stored in the HDFS. There are now other systems like Hive and Pig. And the last component is YARN, which is short for “Yet Another Resource Negotiator.” YARN is a very important component because it prepares the RAM and CPU for Hadoop to run data in batch processing, stream processing, interactive processing, and graph processing, with are stored in HDFS.



## Drawbacks / challenges of Hadoop:

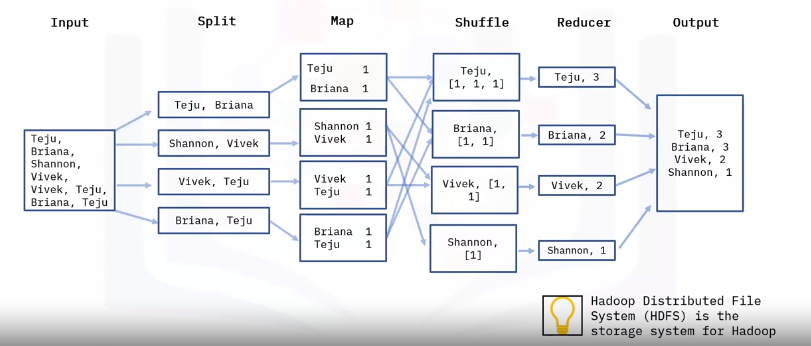


## MapReduce:



## Map and Reduce:

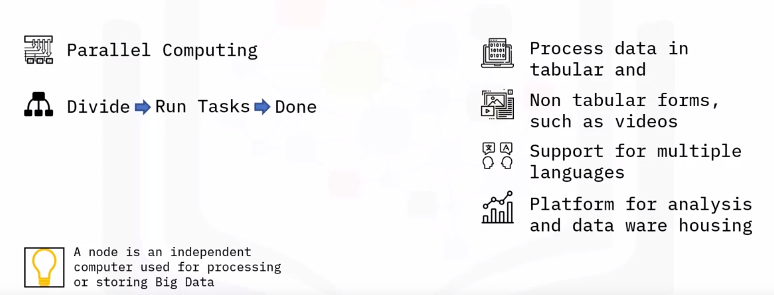
As the name suggests, the MapReduce framework contains two tasks, Map and Reduce. Map takes in an input file and performs some mapping tasks by processing and extracting important data information into a key value pairs and these are the preliminary output list. Some more reorganization goes on before the preliminary output is sent to the Reducer. The Reducer works with multiple map functions and aggregates the pairs using their keys to produce a final output. MapReduce keeps track of its tasks by creating unique keys to ensure that all the processes are solving the same problem.



Let us look at the framework visually. First, we have the Map step, which takes a set of data and converts it into another set of data, where individual elements are broken down into key/value pairs. The key is the name, and the value is the content. The input data is a file that is saved in the Hadoop file system called HDFS. Now let’s assume we have an input file that contains names of people, and we would like to do a word count on the unique name occurrences. First, the data is split into the following four files, each of them in key value pair and are worked on separately. For example, for the first split line for Teju and Briana, we have two key value pairs with one occurrence in each file, and it will do the same for all of key value pairs. We then have the reducer; the reducer processes the data that comes from the map. After processing, it produces a new set of outputs, which will be stored in the HDFS. The Reducer starts with shuffling. Shuffling sorts the key and a list of values in a list, for example, you will see the Key Teju and the corresponding list of values from the previous step. We will have Teju [1, 1, 1], This is because the name Teju occurred 3 times in the “Map” step. It does the same for the rest of the names, counting how many times they appeared in the “Map” step. The Reducer layer then aggregates the values in the list and saves them, then the final output is saved in an output file.

## Why use MapReduce?

The advantages of MapReduce is its ability to allow for a high level of parallel jobs across multiple nodes. A node is an independent computer used for processing and storing big volumes of data. In Hadoop we have two types of nodes, the name node and the data node. Map reduce allows for splitting and running independent tasks in parallel by dividing each task which in turn saves time. MapReduce is very flexible and can process data that come in tabular and non-tabular forms. Therefore, MapReduce provides business value to organizations regardless of how their data is structured. It also offers support for different languages and provides a platform for analysis, data warehousing and more.



## Common use cases:

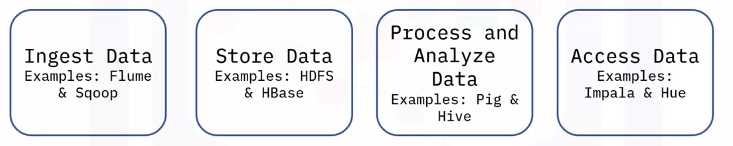
MapReduce has a couple of use cases and here we have some of them displayed. MapReduce can be used for social media platforms like LinkedIn and Instagram to analyze who visited, viewed, and interacted with your profile posts. Map reduce is used by Netflix to recommend movies based on what you have watched in the past by using the user's interests. It is also used in financial institutions like banks and credit card companies to flag and detect anomalies in user transactions. It can also be used in the advertisement industry to understand a user’s behavior by how they engage with ads. Google ads work by using MapReduce to understand the engagement of users with an ad.

**Summary**

MapReduce is a framework used in Parallel Computing, it contains two major tasks: Map and Reduce. Map processes data into Key Value pairs, sorts and organizes the data. The Reducer aggregates and computes a set of results and produces a final output. It is flexible for all data types like structured and unstructured data and can be applied to multiple industries such as social media, entertainment, and many more.

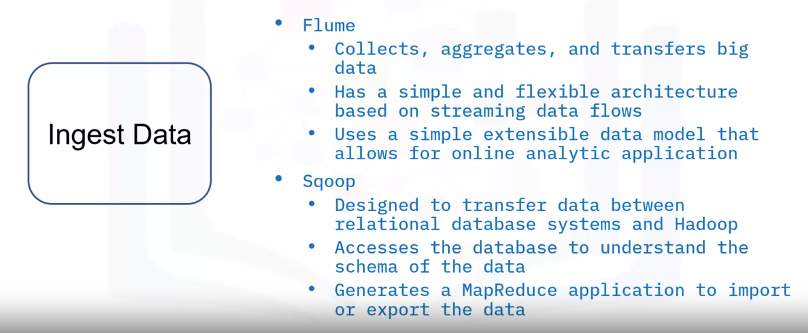
## Hadoop Ecosystem:

Hadoop Common refers to the common utilities and libraries that support the other Hadoop modules. Hadoop Distributed File System (HDFS) stores the data collected from the ingestion and distributes the data across multiple nodes. MapReduce is used for making Big Data manageable by processing them in clusters. Yet Another Resource Negotiator (YARN) is the resource manager across clusters. The extended Hadoop Ecosystem consists of libraries or software packages that are commonly used with or installed on top of the Hadoop core. The Hadoop ecosystem is made up of components that support one another for Big Data processing. We can examine the Hadoop ecosystem based on the various stages. When data is received from multiple sources, Flume and Sqoop are responsible for ingesting the data and transferring them to the Storage component, HDFS and HBase. Then, the data is distributed to a MapReduce framework like Pig and Hive to process and analyze the data, and the processing is done by parallel computing. After all that is done, tools like Hue are used to access the refined data.

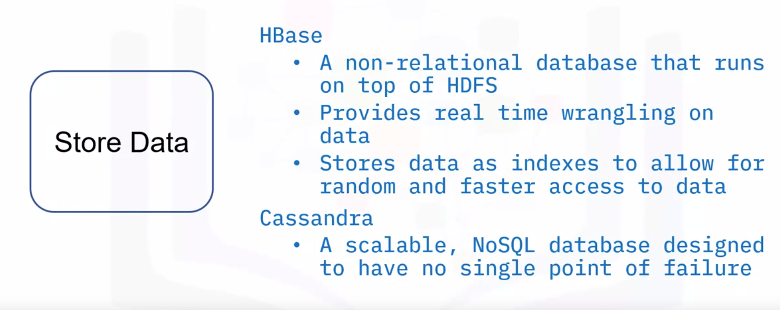


### Ingest data:

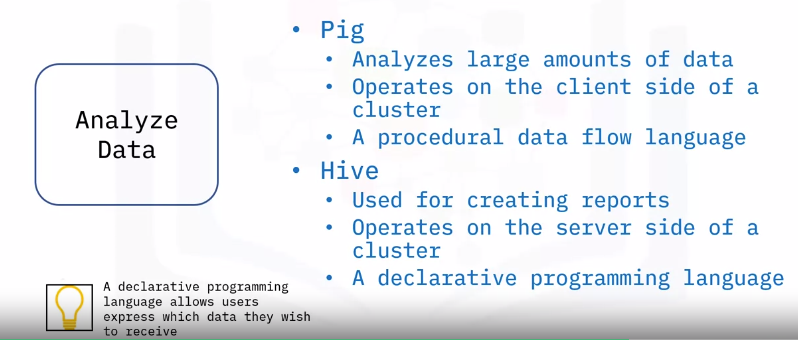
Ingesting is the first stage of Big Data processing. Whenever you deal with big data, you get data from different sources. You then use tools like Flume and Sqoop. Flume is a distributed service that collects, aggregates, and transfers Big Data to the storage system. Flume has a simple and flexible architecture based on streaming data flows and uses a simple extensible data model that allows for online analytic application. Sqoop is an open-source product designed to transfer bulk data between relational database systems and Hadoop. Sqoop looks in the relational database and summarizes the schema. It then generates MapReduce code to import and export data as needed. Sqoop allows you to quickly develop any other MapReduce applications that use the records that Sqoop stored into HDFS.



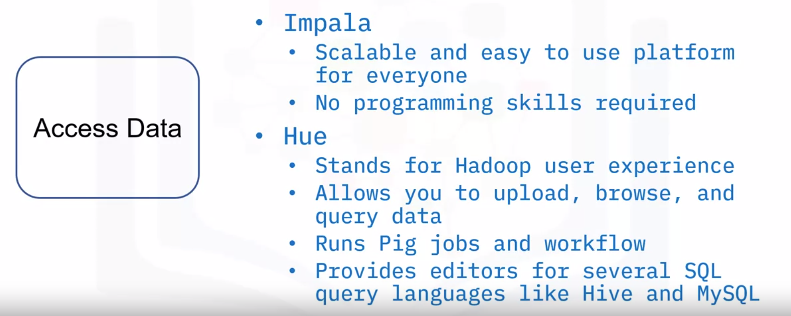
### Store Data:



### Analyze data:



### Access data:



## HDFS:

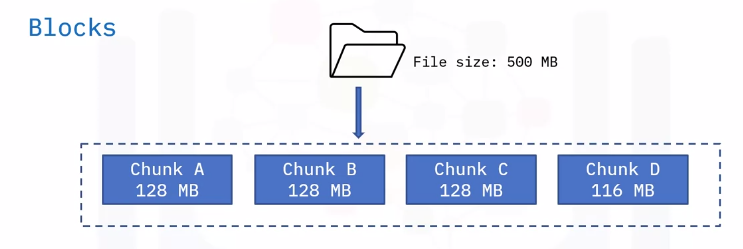
HDFS stands for Hadoop Distributed File System. A distributed file system is a file system that is distributed on multiple file servers and allows programmers to access or store files from any network or computer. HDFS is the storage layer of Hadoop. HDFS works by splitting the files into blocks, then creating replicas of the blocks, and storing them on different machines. HDFS is built to access streaming data seamlessly. Streaming means that HDFS provides a constant bitrate when transferring data rather than having the data being transferred in waves. HDFS uses a command line interface to interact with Hadoop.

### Key features:

The key features of HDFS are: The commodity hardware that stores the data is not expensive, and therefore reduces storage costs. HDFS can store very large amounts of data, as large as petabytes— in any format, tabular and non-tabular. It splits these large amounts of data into small chunks called blocks. One of the great features of HDFS is its ability to replicate and minimize the costs associated with data losses when there is failure with one of the hardware units. That capability makes HDFS fault tolerant. In the event of a data loss of one of the computers, the data can be found on another computer and work continues. HDFS is also highly scalable. A single cluster can scale into hundreds of nodes. Portability is also one of the key features, as HDFS is designed to easily move from one platform to another.

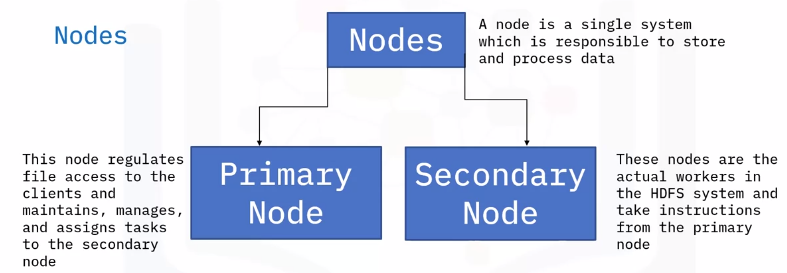
### HDFS Concepts:

To get familiar with Hadoop, there are a few concepts to know. When HDFS receives files, files are broken into smaller chunks called blocks. A block is the minimum amount of data that can be read or written and provides fault tolerance. Depending on your system configuration, the default block size could be 64 or 128 megabytes. For example, if we had a 500-megabyte file, with a default block chunk size of 128 megabytes, the file will be divided into 3 blocks of 128 megabytes and one block of 116 megabytes. The only time you will have equal splits is if the file size is a multiple of the default block size. Therefore, you can see that each file stored doesn’t have to take up the storage of the pre-configured block size.



### Nodes:

Let’s start with nodes. A node is a single system which is responsible for storing and processing data. Think about it as one machine or computer in which data is stored. Remember that HDFS follows the primary/secondary concept. HDFS has two types of nodes: The Primary node, known as the name node, regulates file access to the clients and maintains, manages, and assigns tasks to the secondary node, also known as a data node. There can be hundreds of data nodes in the HDFS that manage the storage system. They perform read and write requests at the instruction of the name node.



### Rack Awareness:

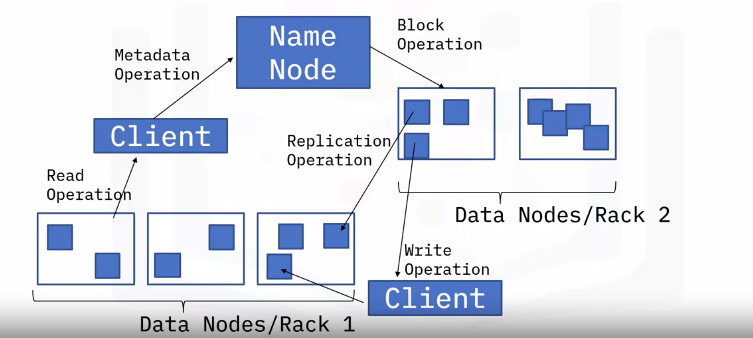
When performing operations like read and write, it is important that the name node maximizes performance by choosing the data nodes closest to themselves. This could be by choosing data nodes on the same rack or in nearby racks. This is called rack awareness. A rack is the collection of about forty to fifty data nodes using the same network switch. Rack awareness is used to reduce the network traffic and improve cluster performance. To achieve rack awareness, the name node keeps the rack ID information.

### Replication:

Replication is done by rack awareness as well. It is done by making sure replicas of a data node are in different racks. So, if a rack is down, you can still obtain the data from another rack. HDFS is known for optimizing replication. HDFS uses the rack awareness concept to create replicas to make sure that the data is reliable and available, and that the network bandwidth is properly utilized. Replication is creating a copy of the data block. When crashes happen, replication provides backup of the data blocks. Replication factor is defined as the number of times you make a copy of the data block.

### HDFS Architecture:

Hadoop follows the concept of a primary/secondary node architecture. The primary node is the name node. The architecture is such that per cluster, there is one name node and multiple data nodes, which are the secondary nodes. Internally, a file is split into one or more blocks and these blocks are stored in a set of data nodes. The name node oversees opening, closing, renaming file operations, and mapping file blocks to the data node. The data nodes are responsible for read and write requests from the client and perform the creation, replication, and deletion of file blocks based on instructions from the name node.



Summary:

