SS01 - INTRODUCTION TO BIG DATA AND DISTRIBUTED SYSTEMS

THE HOANG

OBJECTIVES OF THE SESSION

- Definition and significance of Big Data
- Types of Big Data
- Basics of Distributed Systems
- Overview of Hadoop Ecosystem
- Apache Spark
- Spark vs Hadoop

WHAT IS BIG DATA?

- Definition of Big Data
- Characteristics of Big Data
- Importance in today’s data-driven world

WHAT IS BIG DATA?

How much data is created each day?

- According to the latest estimates, 402.74 million terabytes of data are created each day
- = 402,740,000,000 Gigabytes
- Requires 800M x 500G PC to store this amount

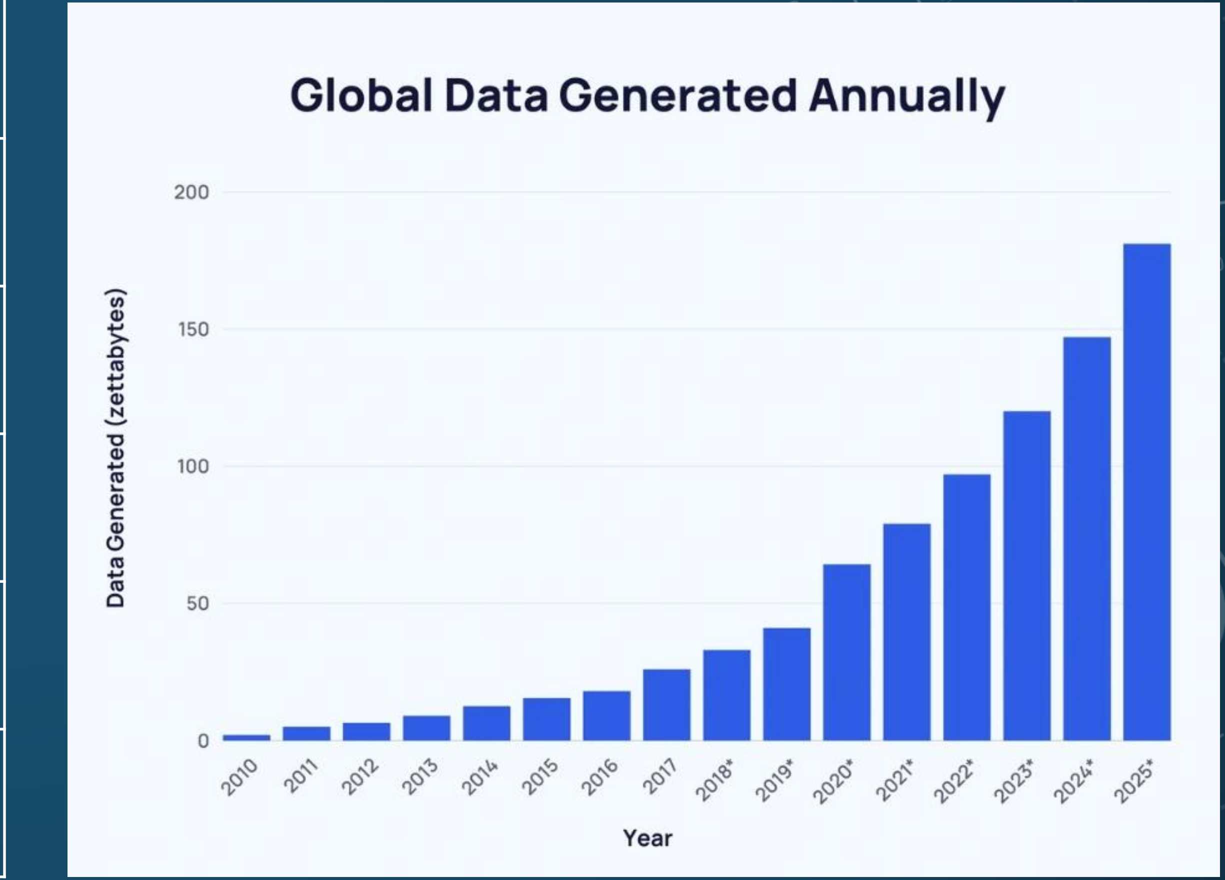
Unit of Measurement	Data Generated
Zettabytes	0.4
Exabytes	402.74
Petabytes	402,740
Terabytes	402.74 million
Gigabytes	402.74 billion
Megabytes	402.74 trillion

Source: explodingtopics

WHAT IS BIG DATA?

Data Creation Growth Projections

2020*	64.2 zettabytes	↑ 23.2 zettabytes	↑ 56.59%
2021*	79 zettabytes	↑ 14.8 zettabytes	↑ 23.05%
2022*	97 zettabytes	↑ 18 zettabytes	↑ 22.78%
2023*	120 zettabytes	↑ 23 zettabytes	↑ 23.71%
2024*	147 zettabytes	↑ 27 zettabytes	↑ 22.5%
2025*	181 zettabytes	↑ 34 zettabytes	↑ 23.13%



Source: explodingtopics

WHAT IS BIG DATA?

Data Creation by Category

Category	Proportion of Internet Data Traffic
Video	53.72%
Social	12.69%
Gaming	9.86%
Web browsing	5.67%
Messaging	5.35%
Marketplace	4.54%
File sharing	3.74%
Cloud	2.73%
VPN	1.39%
Audio	0.31%

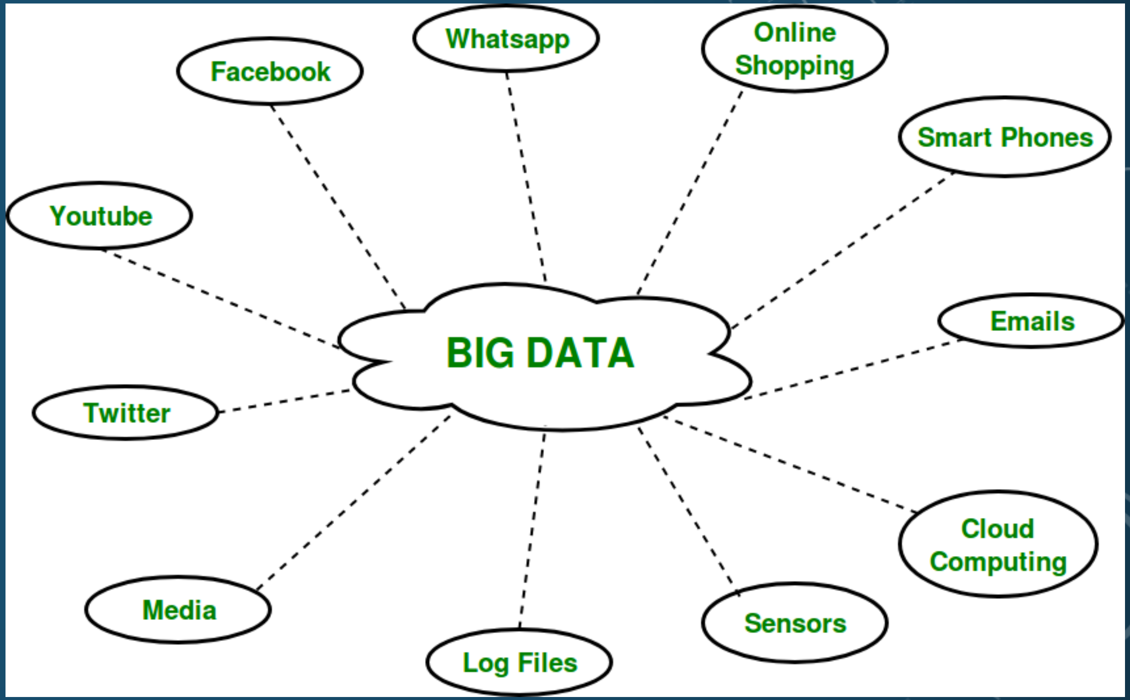
Source: explodingtopics

WHAT IS BIG DATA?

Data Creation by Category

- **Big Data** refers to the vast volumes of structured and unstructured data that are generated every second from various sources, such as social media, sensors, devices, transactions, and more.

- This data is too large, complex, and fast-moving for traditional data processing software to handle effectively. Big Data encompasses not just the size but also the challenges related to storing, processing, and analyzing this information to extract meaningful insights.

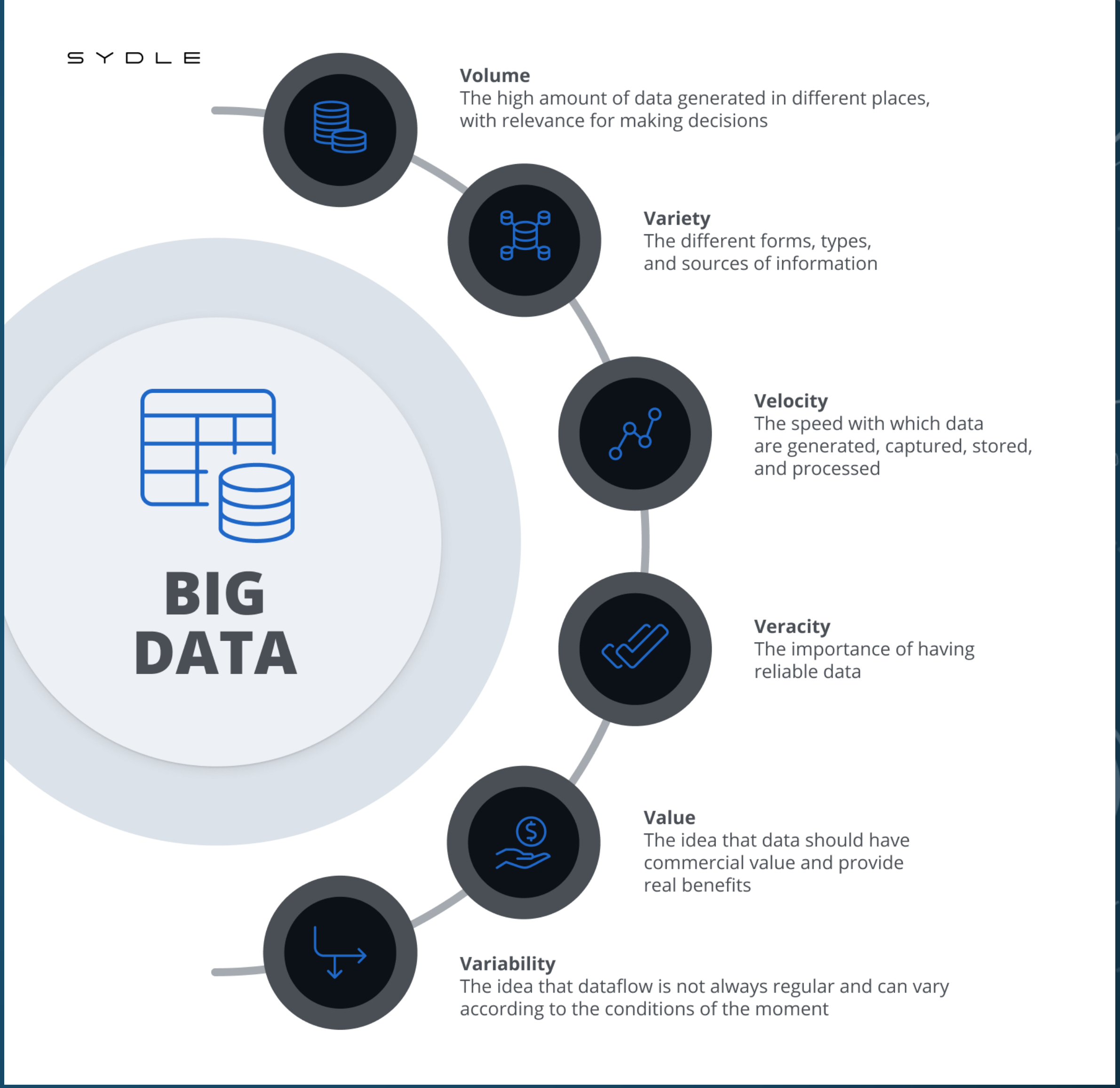


Source: explodingtopics

WHAT IS BIG DATA?

5 most critical V’s of Big Data

- **Volume**: The sheer amount of data generated.
- **Velocity**: The speed at which data is generated and processed.
- **Variety**: The different types of data, both structured and unstructured.
- **Veracity**: The quality and accuracy of the data.
- **Value**: The potential worth of the data for generating business value or insights.



Volume
The high amount of data generated in different places, with relevance for making decisions

Variety
The different forms, types, and sources of information

Velocity
The speed with which data are generated, captured, stored, and processed

Veracity
The importance of having reliable data

Value
The idea that data should have commercial value and provide real benefits

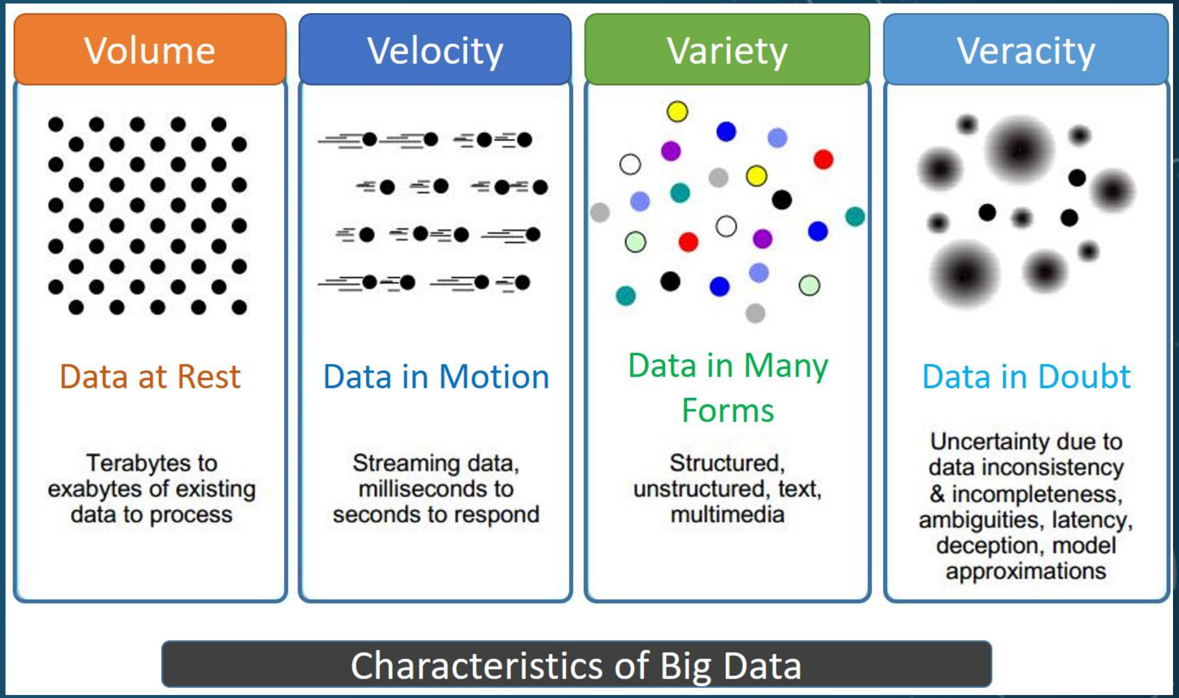
Variability
The idea that dataflow is not always regular and can vary according to the conditions of the moment

Source: SYDLE

WHAT IS BIG DATA?

Characteristics of Big Data

- **High volume**: Massive amounts of data generated continuously.
- **High velocity**: Data generated rapidly and needs to be processed quickly.
- **High variety**: Diverse data types (structured, unstructured, semi-structured).
- **Complexity**: Data is often messy and requires cleaning and preparation.
- **Scalability**: The ability to handle increasing data volumes.
- **Distribution**: Data is often spread across multiple sources and locations.



Volume
Velocity
Variety
Veracity
Data at Rest
Data in Motion
Data in Many Forms
Data in Doubt
Terabytes to exabytes of existing data to process
Streaming data, milliseconds to seconds to respond
Structured, unstructured, text, multimedia
Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations
Characteristics of Big Data

Source: Oreilly

WHAT IS BIG DATA?

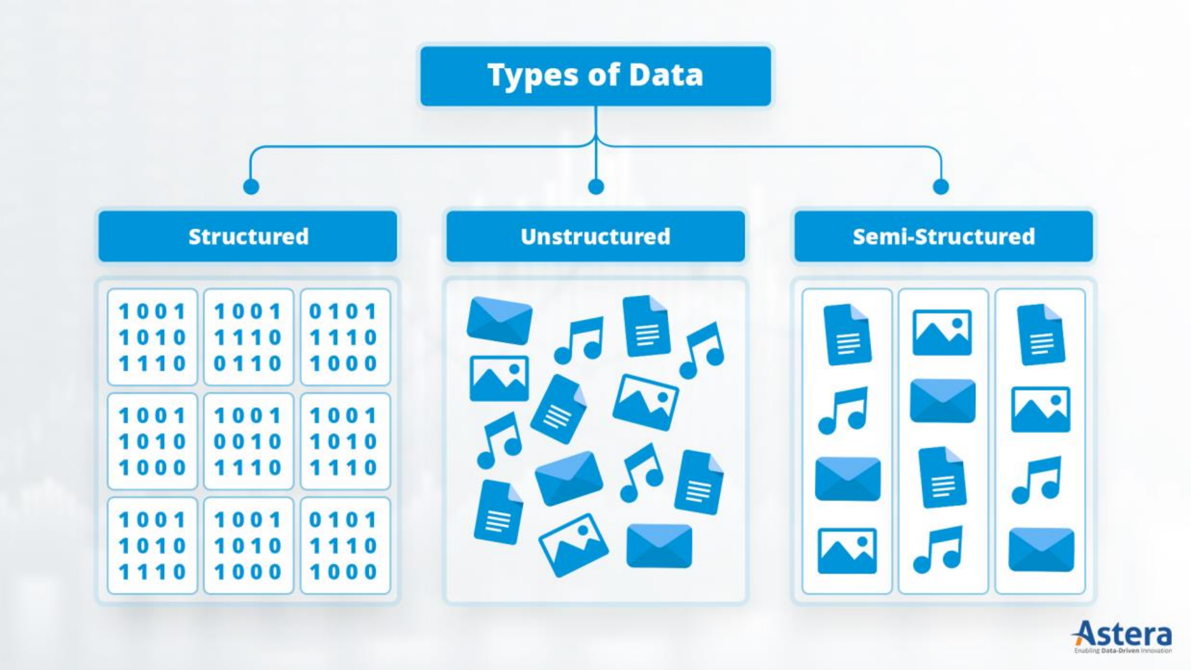
Importance in Today’s Data-Driven World

- **Informed decision-making**: Uncovers patterns and trends for better strategic choices.
- **Competitive advantage**: Gain insights that competitors may not have.
- **Customer understanding**: Analyze customer behavior for personalized experiences.
- **Operational efficiency**: Optimize processes and reduce costs.
- **Innovation**: Fuel new products, services, and business models.
- **Risk management**: Identify potential risks and threats.

Source: Internet

TYPES OF BIG DATA

- Structured Data: Databases, spreadsheets
- Semi-Structured Data: JSON, XML
- Unstructured Data: Text, images, videos
- Examples of each type and their relevance



Source: Internet

TYPES OF BIG DATA

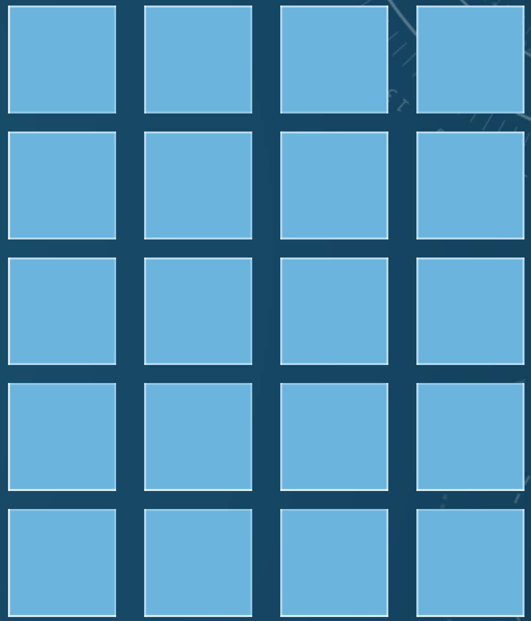
Structured Data

Definition:

Structured data is highly organized and fits neatly into traditional databases. It is typically stored in tables with rows and columns, and each piece of data has a predefined data type. This makes it easy to query and analyze.

Examples:

- Relational databases like MySQL, Oracle, and SQL Server
- Spreadsheets like Excel and Google Sheets



Source: Internet

TYPES OF BIG DATA

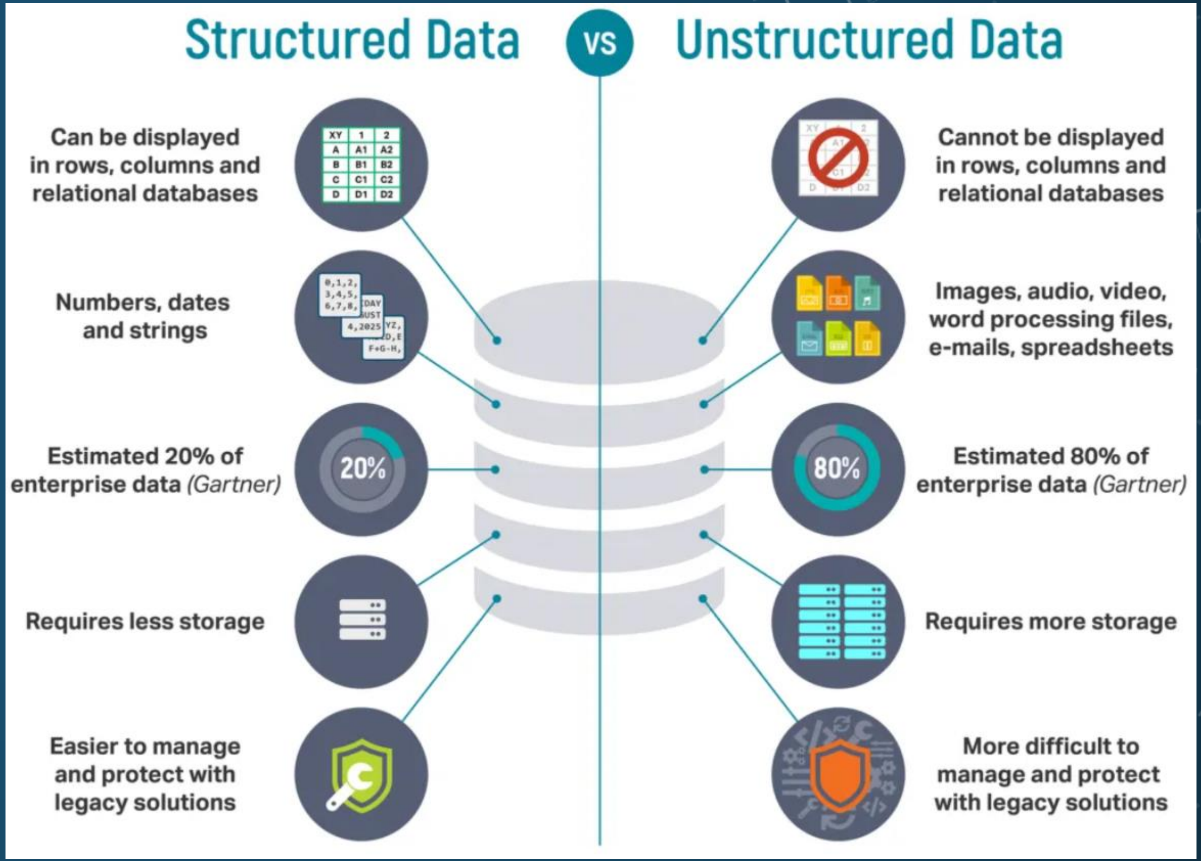
Un-structured Data

Definition:

Unstructured data is the most common type of data and does not have a predefined data model or organization. It is typically stored in a variety of formats, such as text, images, videos, and audio.

Examples:

- Text: Text documents, emails, social media posts, and news articles
- Images: Images, photos, and graphics
- Videos: Videos, movies, and TV shows
- Audio: Audio files, music, and podcasts



Structured Data VS Unstructured Data

Can be displayed in rows, columns and relational databases

Cannot be displayed in rows, columns and relational databases

Numbers, dates and strings

Images, audio, video, word processing files, e-mails, spreadsheets

Estimated 20% of enterprise data *(Gartner)*

Estimated 80% of enterprise data *(Gartner)*

Requires less storage

Requires more storage

Easier to manage and protect with legacy solutions

More difficult to manage and protect with legacy solutions

Source: Internet

TYPES OF BIG DATA

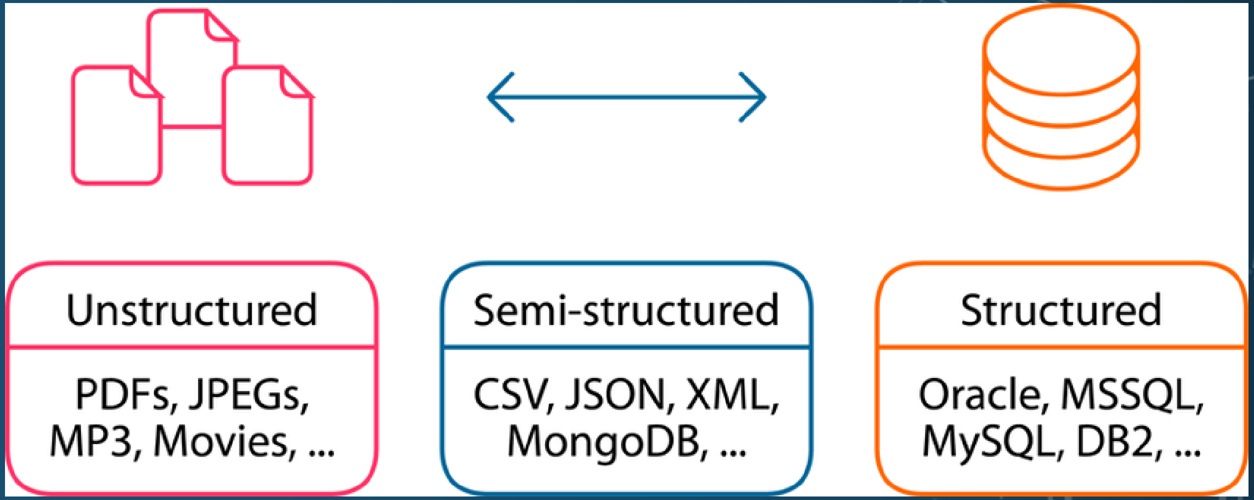
Semi-structured Data

Definition:

Semi-structured data is a hybrid of structured and unstructured data. It has some organization but does not conform to a strict schema like structured data. It is often stored in hierarchical formats like XML or JSON.

Examples:

- JSON: JSON (JavaScript Object Notation)
- XML: XML (Extensible Markup Language)



Unstructured

PDFs, JPEGs, MP3, Movies, ...

Semi-structured

CSV, JSON, XML, MongoDB, ...

Structured

Oracle, MSSQL, MySQL, DB2, ...

Source: Internet

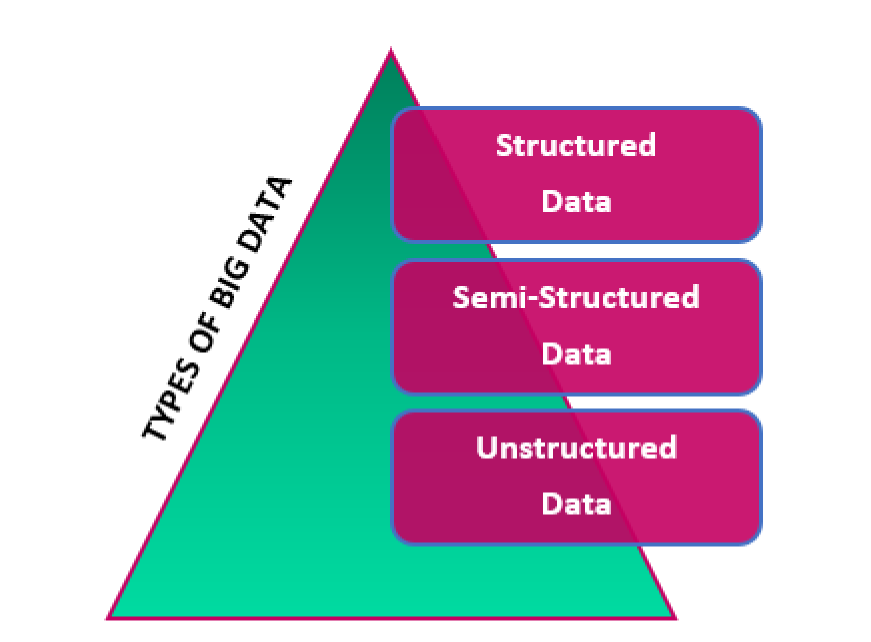
TYPES OF BIG DATA

Additional points

Structured data is the easiest type of data to analyze. It can be easily queried and processed using traditional database tools.

Semi-structured data is becoming increasingly popular due to its flexibility. It can be used for a variety of applications, and it is often easier to store and manage than unstructured data.

Unstructured data is the most challenging type of data to analyze. It requires specialized tools and techniques to extract value from it.



Source: Internet

CHALLENGES OF BIG DATA

- Storage and Management
- Data Quality and Cleansing
- Analysis and Processing Speed
- Privacy and Security Concerns

CHALLENGES OF BIG DATA

Storage and Management:

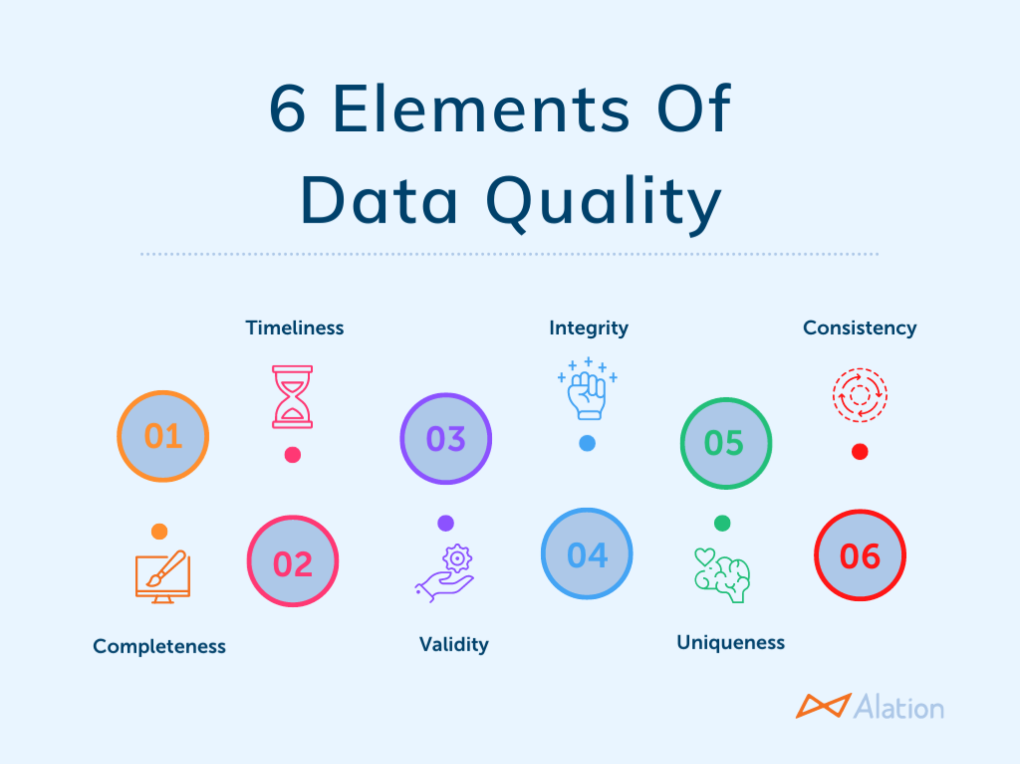
- **Cost**: Storing and managing large volumes of data can be expensive.
- **Scalability**: Storage systems must be able to scale to accommodate growing data volumes.
- **Data Loss**: The risk of data loss is higher with large datasets.
- **Data Accessibility**: Ensuring data is accessible and available when needed.

Source: Internet

CHALLENGES OF BIG DATA

Data Quality and Cleansing:

- **Inconsistent Data**: Data from different sources may be formatted differently, making it difficult to integrate and analyze.
- **Missing Data**: Data may be missing or incomplete, which can lead to inaccurate results.
- **Duplicate Data**: Duplicate data can lead to wasted storage space and processing time.
- **Data Errors**: Errors in data can lead to incorrect results.
- **Data Cleansing**: The process of cleaning and preparing data for analysis can be time-consuming and expensive.

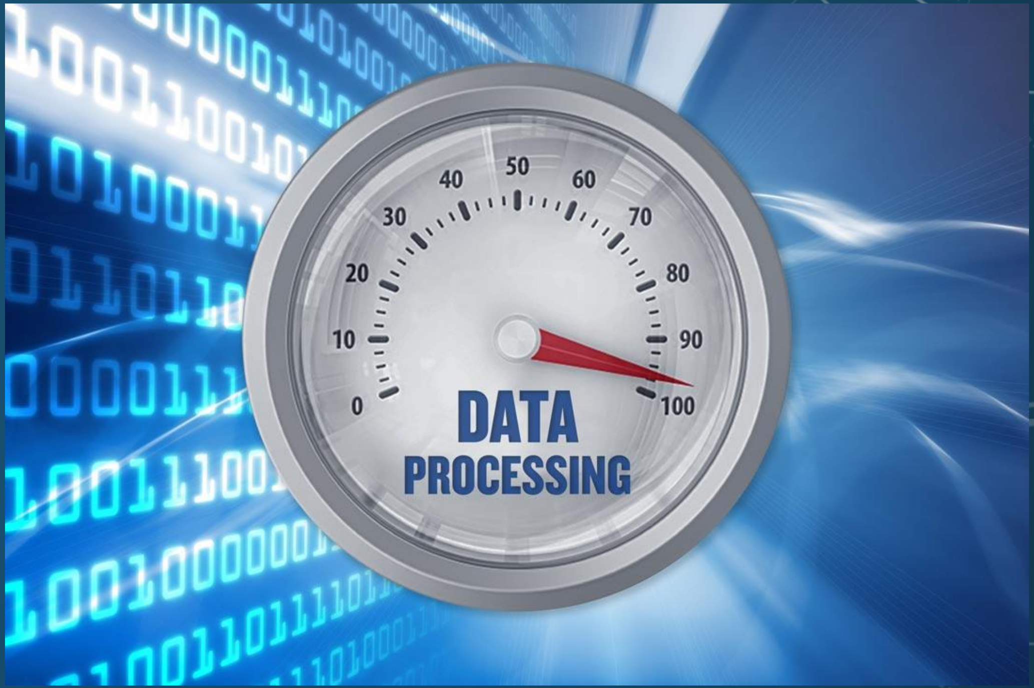


Source: Internet

CHALLENGES OF BIG DATA

Analysis and Processing Speed:

- **Data Volume**: Large datasets can take a long time to process.
- **Data Complexity**: Complex data structures can be difficult to analyze.
- **Data Velocity**: Data is often generated at high speeds, making it difficult to keep up with the analysis.
- **Real-time Processing**: The need to process data in real-time can be challenging.

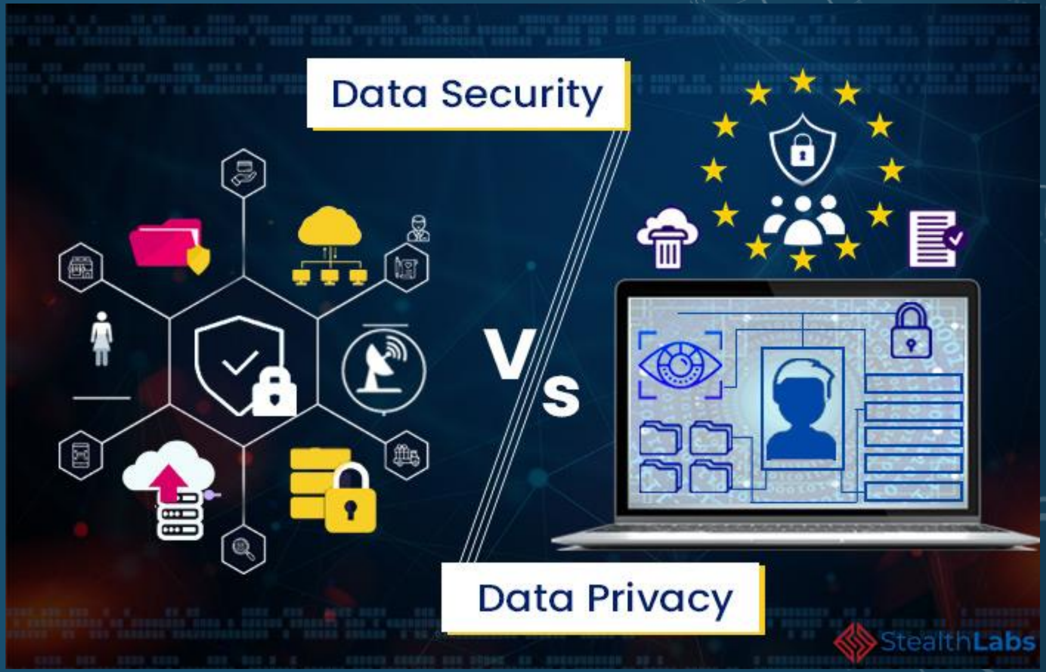


Source: Internet

CHALLENGES OF BIG DATA

Privacy and Security Concerns:

- **Data Breaches**: The risk of data breaches is high with large datasets.
- **Data Privacy**: Protecting sensitive data from unauthorized access.
- **Data Compliance**: Ensuring compliance with data privacy regulations.
- **Data Security**: Protecting data from loss, damage, or corruption.
- **Data Governance**: Managing data access and usage.



Source: Internet

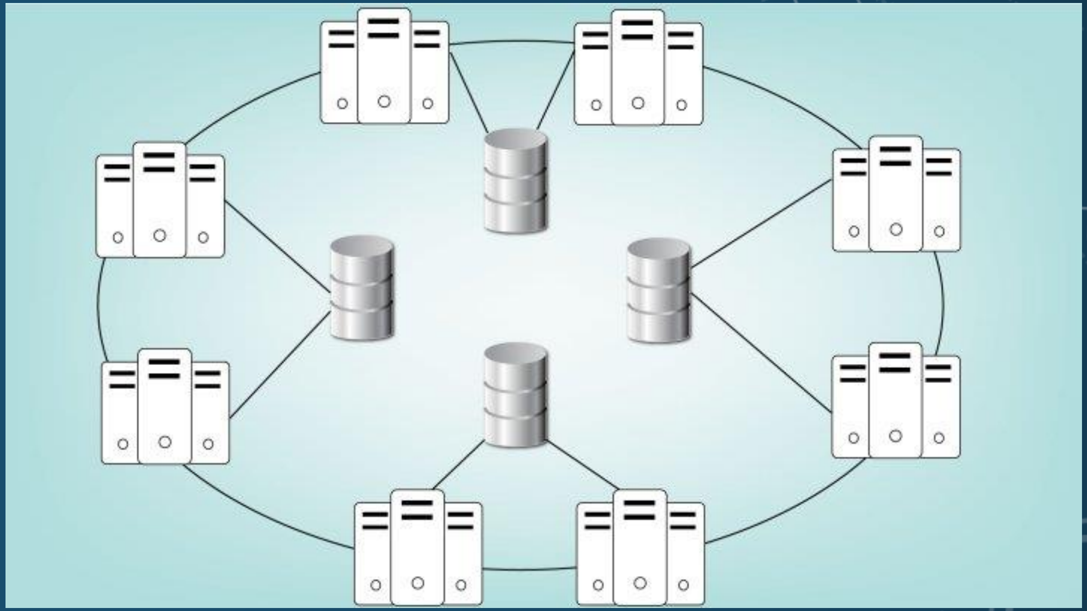
INTRODUCTION TO DISTRIBUTED SYSTEMS

- Definition of Distributed Systems
- Key Characteristics:
  - Scalability
  - Fault Tolerance
  - Resource Sharing
- Examples in real-world applications

INTRODUCTION TO DISTRIBUTED SYSTEMS

Definition of Distributed Systems

- A distributed system is a collection of independent computer systems, often geographically dispersed, that communicate and coordinate their actions to appear as a single coherent system to its users. These systems share resources, information, and processing power to achieve a common goal. Essentially, it's a software system where components located on networked computers communicate and coordinate their actions by passing messages



Source: Internet

INTRODUCTION TO DISTRIBUTED SYSTEMS

Key Characteristics

- **Independence**: Each component in a distributed system can fail independently without affecting the entire system.
- **Concurrency**: Multiple components can execute concurrently.
- **No shared clock**: Systems operate on different local times, requiring careful synchronization.
- **No shared memory**: Components access different physical memory locations.
- **Network communication**: Components interact through message passing over a network.
- **Heterogeneity**: Systems can consist of different hardware and software components.
- **Scalability**: Distributed systems can be easily scaled by adding or removing components.

Source: Internet

INTRODUCTION TO DISTRIBUTED SYSTEMS

Real-world Examples

- **E-commerce platforms**: Handling orders, payments, inventory management across multiple servers.
- **Cloud computing**: Distributing workloads across multiple data centers.
- **Social media platforms**: Managing user profiles, posts, and interactions on a global scale.
- **Online gaming**: Coordinating player actions, game state, and communication across different servers.
- **Financial systems**: Processing transactions, managing accounts, ensuring data consistency across branches.
- **Search engines**: Indexing and serving search results from multiple data centers.
- Etc …

Source: Internet

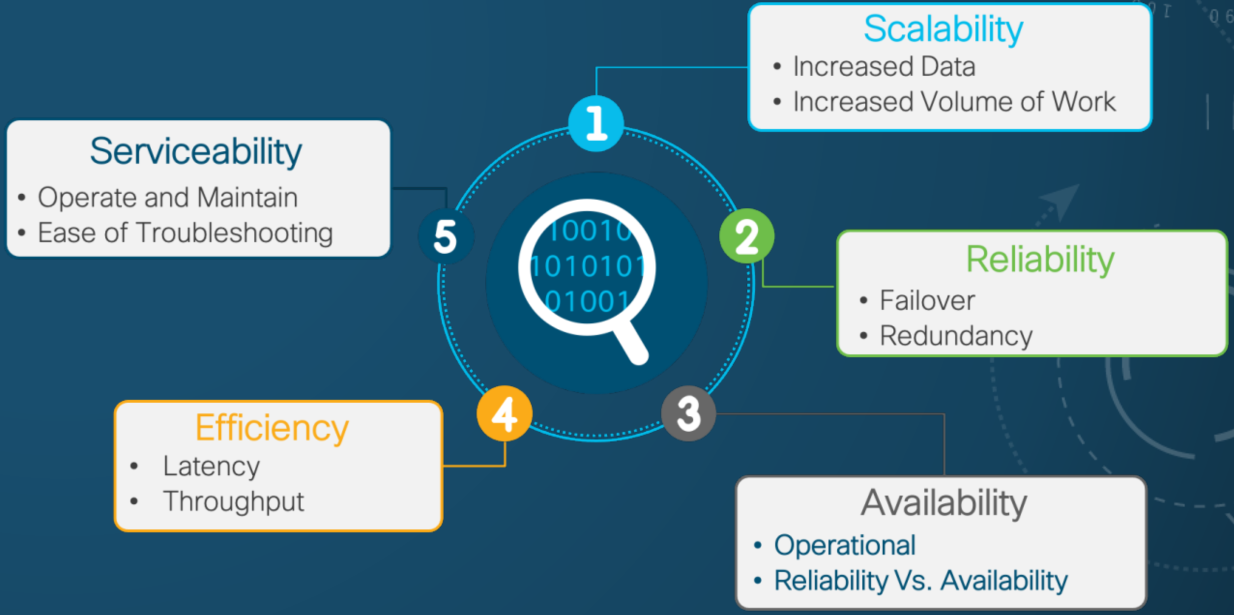
WHY USE DISTRIBUTED SYSTEMS FOR BIG DATA?

- Benefits of distributed processing
- How it handles large datasets efficiently
- Real-time processing capabilities

WHY DISTRIBUTED SYSTEMS?

Benefits of distributed processing

- **Scalability**: Grow with data by adding more computers.
- **Reliability**: System keeps working even if parts fail.
- **Availability**: Always accessible, even with problems.
- **Cost-effective**: Cheaper than one big computer.



Scalability
- Increased Data
- Increased Volume of Work

Reliability
- Failover
- Redundancy

Availability
- Operational
- Reliability Vs. Availability

Efficiency
- Latency
- Throughput

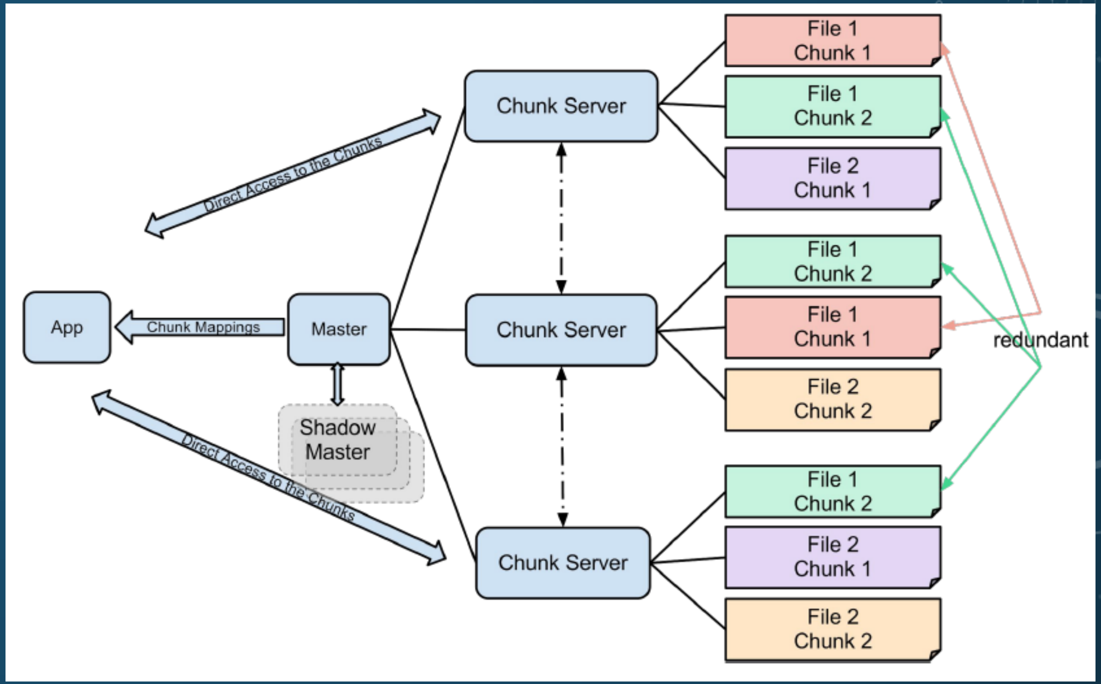
Serviceability
- Operate and Maintain
- Ease of Troubleshooting

Source: Internet

WHY DISTRIBUTED SYSTEMS?

How it handles large datasets efficiently

- **Partitioning / bucketing**: Break data into smaller parts and distribute it across multiple computers (nodes).
- **Parallelism:** Multiple nodes process data at once. Much faster than traditional systems
- **Data locality**: Store data near the computers processing it, reducing network traffic and improving performance.



Source: Internet

WHY DISTRIBUTED SYSTEMS?

Real-Time Processing

- **Low Latency**: process data with minimal delay, enabling real-time analysis and decision-making. This is crucial for applications like fraud detection, recommendation systems, and IoT data processing.
- **Stream Processing**: handle continuous data streams efficiently, allowing for real-time insights and updates. This is essential for applications that deal with rapidly changing data, such as social media analytics and financial market data.

Source: Internet

OVERVIEW OF THE HADOOP ECOSYSTEM

- Introduction to Hadoop:
  - HDFS (Hadoop Distributed File System)
  - MapReduce framework
- Components of the Hadoop Ecosystem:
  - Hive, Pig, HBase, Sqoop, Flume, etc.
- Brief explanation of each component's role

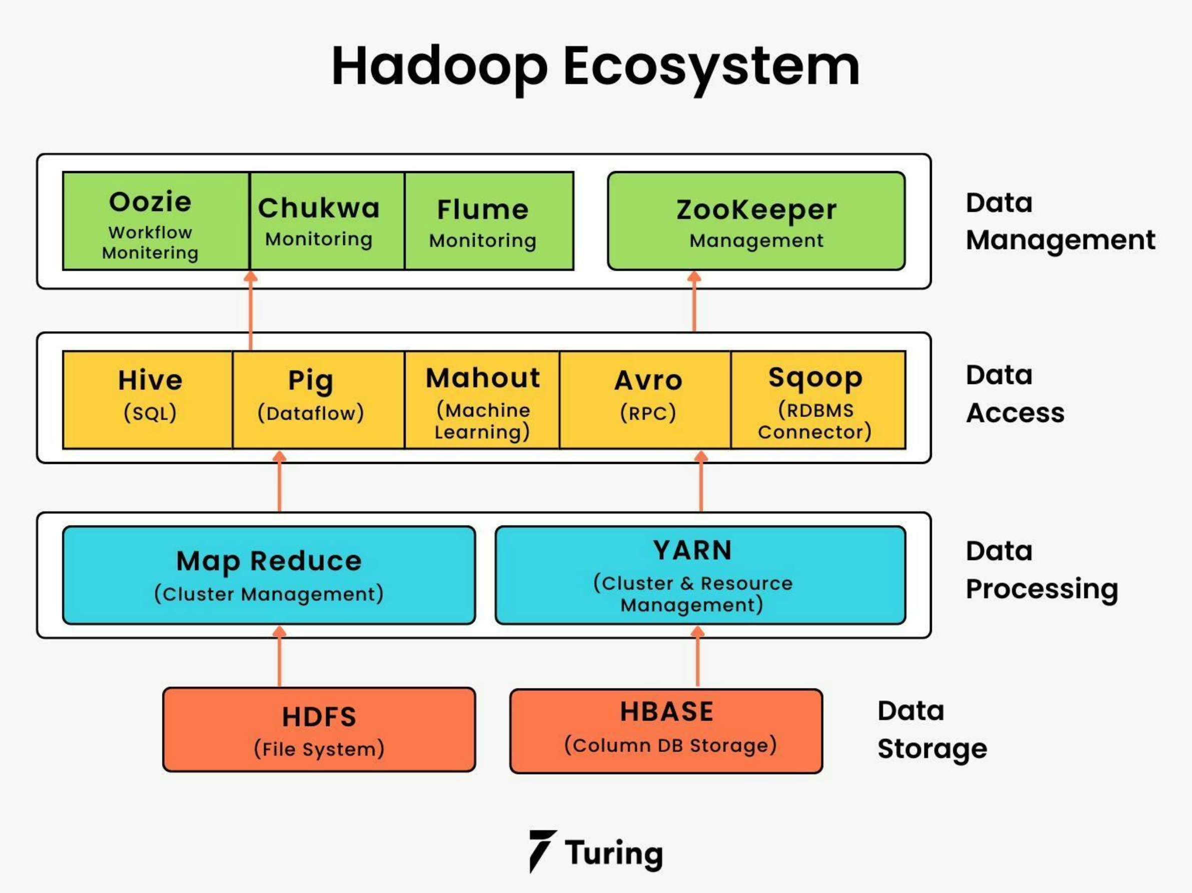
OVERVIEW OF THE HADOOP ECOSYSTEM

What is Hadoop Ecosystem?

The Hadoop ecosystem is a collection of open-source tools and frameworks designed to store, process, and analyze (access) and manage massive amounts of data.

It's built around the core components of HDFS (for storage) and MapReduce (for processing), but it encompasses a wide range of tools to address different Big Data challenges.

Other components: YARN; Hive; Pig; HBase; Spark; Zookeeper, … and many more

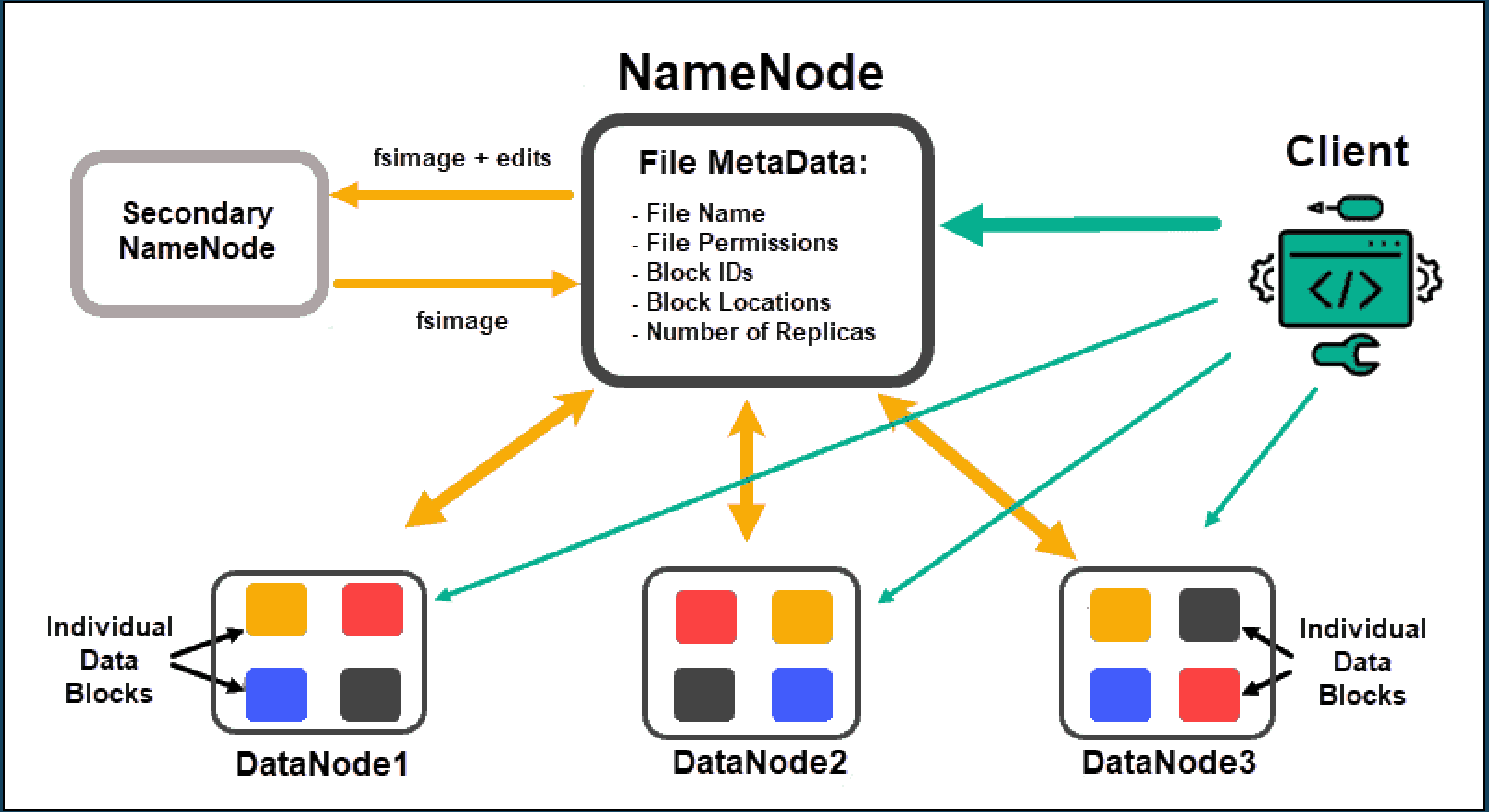


Source: Internet

OVERVIEW OF THE HADOOP ECOSYSTEM

HDFS (Hadoop Distributed File System)

HDFS is the storage component of Hadoop. It's designed to store massive amounts of data across multiple commodity servers.



Source: Internet

OVERVIEW OF THE HADOOP ECOSYSTEM

HDFS (Hadoop Distributed File System)

Key characteristics

- High fault tolerance: Data is replicated across multiple nodes for redundancy.
- Scalability: Easily add more nodes to store increasing amounts of data.
- Streaming access: Data is read in a stream, optimizing for large data processing.
- Write once, read many: Data is written once and read multiple times.

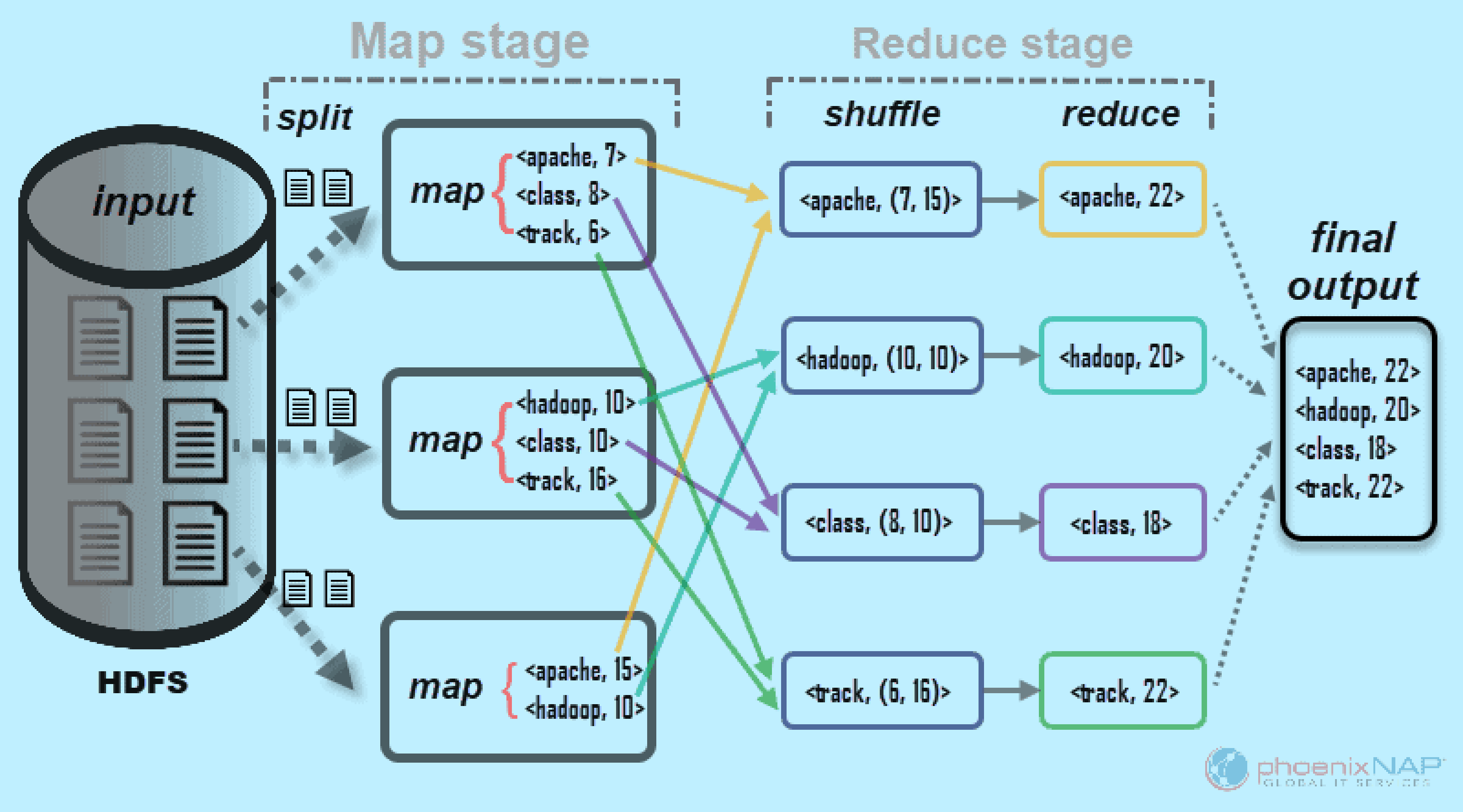
How it works:

- Data is split into blocks and stored across different nodes.
- Replication ensures data availability.
- NameNode manages file system metadata.
- DataNodes store data blocks.

OVERVIEW OF THE HADOOP ECOSYSTEM

MapReduce Framework

MapReduce is the processing engine of Hadoop. It's a programming model for processing large datasets across clusters of computers.



Source: Internet

OVERVIEW OF THE HADOOP ECOSYSTEM

MapReduce Framework

Key concepts:

- Map: Breaks down data into key-value pairs.
- Reduce: Combines key-value pairs to produce final output.
- Distributed processing: Distributes tasks across multiple nodes.
- Fault tolerance: Handles node failures by re-executing tasks.

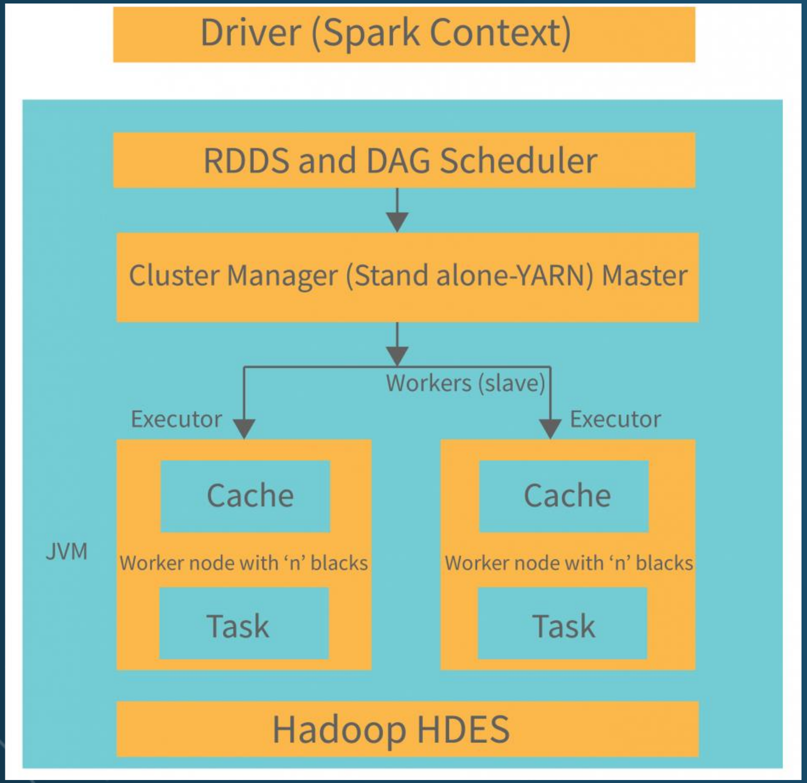
How it works:

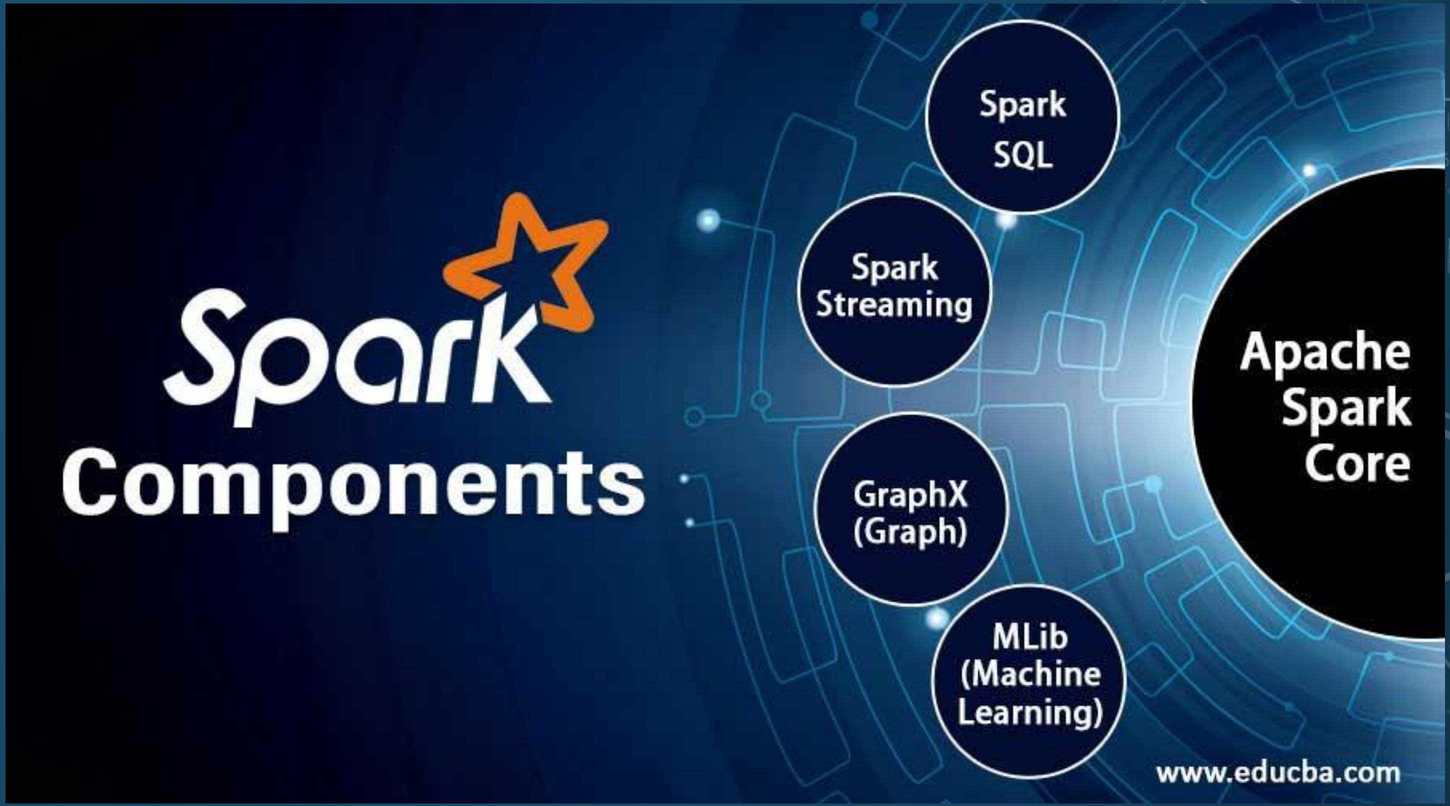
- Input data is split into chunks.
- Map tasks process each chunk independently.
- Map outputs are sorted and grouped by key.
- Reduce tasks process grouped data to produce final output.

APACHE SPARK

What is Apache Spark

Apache Spark is an open-source unified analytics engine for large-scale data processing. Spark provides an interface for programming clusters with implicit data parallelism and fault tolerance.





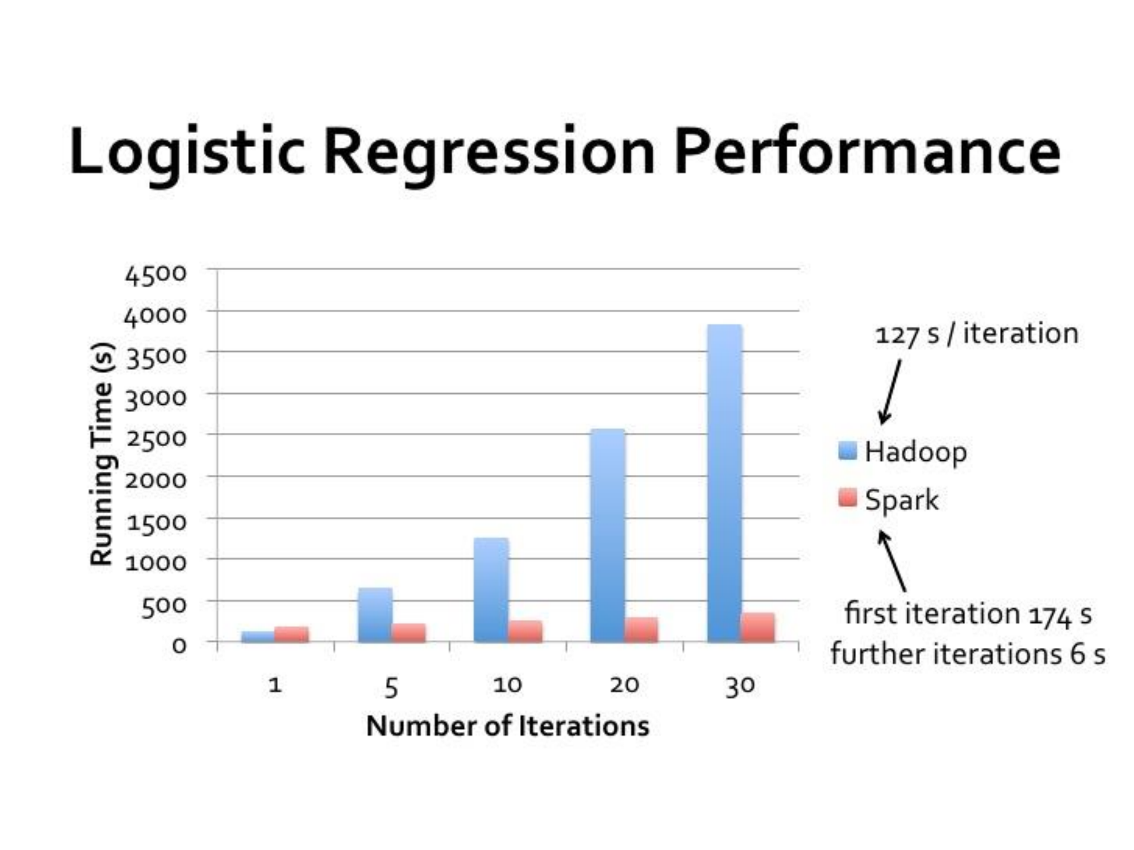
Source: Internet

APACHE SPARK VS. HADOOP

Speed?

- Hadoop: Slow, uses hard drives. => 3G
- Spark: Fast, uses computer memory. => 4G

- Spark is 100x faster than Hadoop thanks to in-memory processing, DAG execution model and many other sophisticated optimization techniques.



Source: Internet

APACHE SPARK VS. HADOOP

Ease of Use?

- Hadoop: Complex, harder to learn.
- Spark: Simpler, easier to use.

APACHE SPARK VS. HADOOP

Architecture

Feature	Spark	Hadoop
Data Structure	Resilient Distributed Datasets (RDDs)	Hadoop Distributed File System (HDFS)
Execution Model	Directed Acyclic Graph (DAG)	MapReduce
Fault Tolerance	RDD lineage for efficient recovery	Replication of data blocks

APACHE SPARK VS. HADOOP

How They Work?

- Hadoop: Stores data on disks and processes it slowly.
- Spark: Keeps data in computer memory for faster processing.

Feature	Spark	Hadoop
Storage	Primarily in-memory, also supports external storage	Disk-based
Processing Speed	Significantly faster due to in-memory computation	Slower due to disk I/O
Batch Processing	Efficient	Well-suited
Real-time Processing	Excellent with Spark Streaming	Not optimized
Iterative Processing	Highly efficient	Inefficient due to disk I/O

APACHE SPARK VS. HADOOP

When to Use?

- Hadoop: Huge datasets, simple calculations.
- Spark: Fast calculations, complex tasks, real-time data.

Use Case	Spark	Hadoop
Data Exploration and Analysis	Excellent due to interactive nature	Suitable for large-scale batch processing
Machine Learning	Well-suited for iterative algorithms	Can be used but less efficient
Real-time Analytics	Ideal for low latency processing	Limited capabilities
Batch Processing	Efficient for large datasets	Core strength
Graph Processing	Efficient with GraphX library	Can be used but less efficient

REAL-WORLD APPLICATIONS OF BIG DATA

- Case studies or examples:
  - Healthcare analytics
  - E-commerce personalization
  - Social media sentiment analysis

RECOMMENDED RESOURCES

- "Big Data: The Missing Manual" by Tim O'Reilly (Chapter 1)
- edX's "Big Data Essentials" course (Module 1)
- Additional online resources (blogs, tutorials)

Q&A SESSION

CONCLUSION AND KEY TAKEAWAYS

- Recap the key points discussed in the session.
  - Definition and significance of Big Data
  - Types of Big Data
  - Basics of Distributed Systems
  - Overview of Hadoop Ecosystem
  - Apache Spark
  - Spark vs Hadoop

- And, it’s always encouraged to explore further.