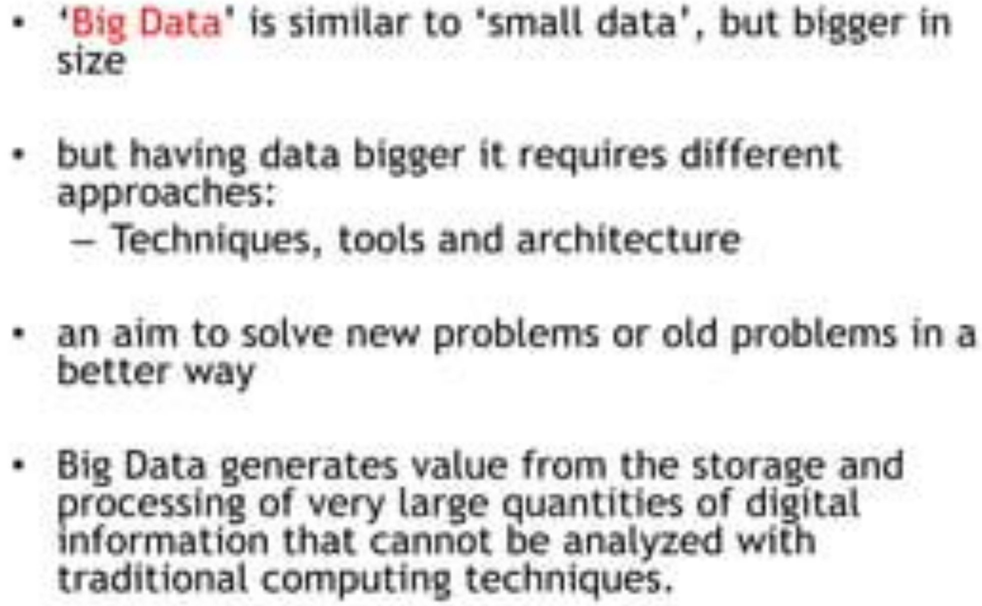
Content

1. Introduction
2. What is Big Data
3. Characteristic of Big Data
4. Storing,selecting and processing of Big Data
5. Why Big Data
6. How it is Different
7. Big Data sources
8. Tools used in Big Data
9. Application of Big Data
10. Risks of Big Data
11. Benefits of Big Data
12. How Big Data Impact on IT
13. Future of Big Data

What is BIG DATA?



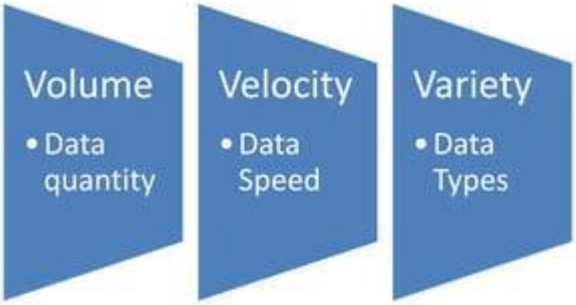
- 'Big Data' is similar to 'small data', but bigger in size

- but having data bigger it requires different approaches:
  - Techniques, tools and architecture

- an aim to solve new problems or old problems in a better way

- Big Data generates value from the storage and processing of very large quantities of digital information that cannot be analyzed with traditional computing techniques.

Three Characteristics of Big Data V3s



Volume
- Data quantity

Velocity
- Data Speed

Variety
- Data Types

1st Character of Big Data
Volume

- A typical PC might have had 10 gigabytes of storage in 2000.

- Today, Facebook ingests 500 terabytes of new data every day.

- Boeing 737 will generate 240 terabytes of flight data during a single flight across the US.

- The smart phones, the data they create and consume; sensors embedded into everyday objects will soon result in billions of new, constantly-updated data feeds containing environmental, location, and other information, including video.

2nd Character of Big Data
Velocity

- Clickstreams and ad impressions capture user behavior at millions of events per second

- high-frequency stock trading algorithms reflect market changes within microseconds

- machine to machine processes exchange data between billions of devices

- infrastructure and sensors generate massive log data in real-time

- on-line gaming systems support millions of concurrent users, each producing multiple inputs per second.

3rd Character of Big Data
Variety

- Big Data isn't just numbers, dates, and strings. Big Data is also geospatial data, 3D data, audio and video, and unstructured text, including log files and social media.

- Traditional database systems were designed to address smaller volumes of structured data, fewer updates or a predictable, consistent data structure.

- Big Data analysis includes different types of data

Storing Big Data

❖ **Analyzing your data characteristics**

- Selecting data sources for analysis
- Eliminating redundant data
- Establishing the role of NoSQL

❖ **Overview of Big Data stores**

- Data models: key value, graph, document, column-family
- Hadoop Distributed File System
- HBase
- Hive

Selecting Big Data stores

- Choosing the correct data stores based on your data characteristics
- Moving code to data
- Implementing polyglot data store solutions
- Aligning business goals to the appropriate data store

Processing Big Data

❖ **Integrating disparate data stores**

- Mapping data to the programming framework
- Connecting and extracting data from storage
- Transforming data for processing
- Subdividing data in preparation for Hadoop MapReduce

❖ **Employing Hadoop MapReduce**

- Creating the components of Hadoop MapReduce jobs
- Distributing data processing across server farms
- Executing Hadoop MapReduce jobs
- Monitoring the progress of job flows

The Structure of Big Data

❖Structured

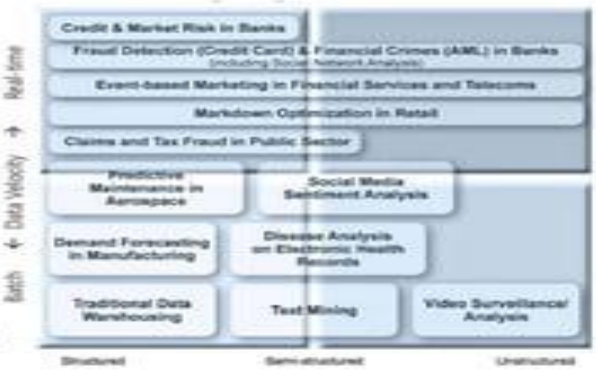
- Most traditional data sources

❖Semi-structured

- Many sources of big data

❖Unstructured

- Video data, audio data


Credit & Market Risk in Banks
Fraud Detection (Credit Card) & Financial Crimes (AML) in Banks (including Social Network Analysis)
Event-based Marketing in Financial Services and Telecoms
Markdown Optimization in Retail
Claims and Tax Fraud in Public Sector
Predictive Maintenance in Aerospace
Social Media Sentiment Analysis
Demand Forecasting in Manufacturing
Disease Analysis on Electronic Health Records
Traditional Data Warehousing
Text Mining
Video Surveillance/ Analysis
Batch ← Data Velocity → Real-time
Structured
Semi-structured
Unstructured

Hadoop

1. What is Hadoop?

Hadoop, as a Big Data framework, provides businesses with the ability to distribute data storage, parallel processing, and process data at higher volume, higher velocity, variety, value, and veracity. HDFS, MapReduce, and YARN are the three major components for this Hadoop tutorial.

Hadoop HDFS uses name nodes and data nodes to store extensive data. MapReduce manages these nodes for processing, and YARN acts as an Operating system for Hadoop in managing cluster resources.

2. Hadoop Ecosystem

Hadoop is a collection of multiple tools and frameworks to manage, store, the process effectively, and analyze broad data. HDFS (Hadoop distributed file system) acts as a distributed file system to store large datasets across commodity hardware. YARN is the Hadoop resource manager to handle a cluster of nodes, allocate RAM, memory, and other resources depending on the application requirements.

MapReduce handles the data processing, Sqoop for transferring data from the current Hadoop database, and other external databases, Flume for data collection and indigestion tool, Pig as script framework, Hive for querying through distributed storage, Spark for real-time data processing and analyzing, Mahout for algorithms, and Apache Ambari for real-time tracking.

Hadoop Installation on Ubuntu

Hadoop cluster setup on Ubuntu requires a lot of software to work together. First of all, you need to download the Oracle VM box and the Linux disc image to start with a virtual software setting up a cluster

4. Hadoop Architecture

Hadoop architecture has four essential components that offer support for parallel processing in storing humongous data with a node system. Hadoop HDFS for storing data in multiple slave machines, Hadoop YARN in managing resources across a cluster of machines, Hadoop MapReduce to process and analyze distributed data, and Zookeeper to sync the system across multiple hardware. Hadoop architecture is the basis for understanding this Big Data framework and generating actionable insights to help businesses scale in the right direction.

5. HDFS

Hadoop Distributed File System (HDFS) offers comprehensive support for huge files. HDFS can manage data in the size of petabytes and zettabytes data. HDFS comes packed with the ability to write or read terabytes of data per second, distribute data across multiple nodes in a single seek operation,

YARN infrastructure provides resources for executing applications. The MapReduce framework runs on YARN to divide functionalities with resource management and job scheduling for comprehensive monitoring.

MapReduce programming model is based on two phases as Mapping and Reducing. Mapping classifies data into nodes, and the Reducer class generates the final product by aggregating and reducing the output. It can process and compute significantly large volumes of data.

Pig is the leading scripting platform to process and analyze Big Datasets. It can use structured and unstructured data to get actionable insights and then stores the result in HDFS. Pig has two essential components; first, a Pig Latin script language along with a runtime engine to process and analyze MapReduce programs.

Hive Acting as a Data warehouse software, Hive uses SQL like language, HiveQL, for querying through distributed databases. There are mainly two Hive data types; first, as Primitive data types with numeric, string, date/time, and miscellaneous data types, and secondary Complex data types include arrays, maps, structs, and units. Similarly, the Hive has two differences with Local Mode and MapReduce Mode.

HBase is a complete storage system built with the primary aim of managing billions of rows and millions of columns across community hardware. HBase enables data to store in tabular form, thus making it exceptionally easy for fast reads and writes.

Sqoop acts as a tool or medium to load data from any external relational database management system (RDBMS) to the Hadoop system and then further to export to RDBMS, respectively.

ETL Process

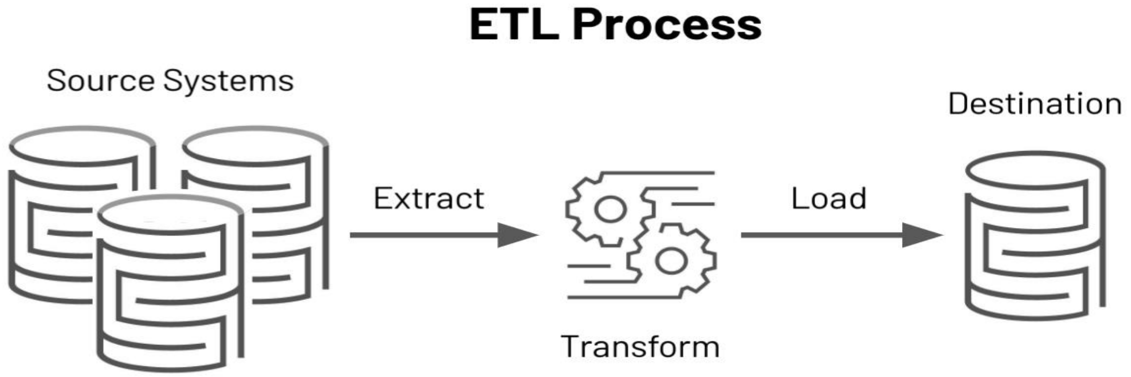
ETL stands for *Extract-Transform-Load*, it usually involves moving data from one or more sources, making some changes, and then loading it into a new single destination.

What is an ETL pipeline

An ETL pipeline (or data pipeline) is the mechanism by which ETL processes occur. Data pipelines are a set of tools and activities for moving data from one system with its method of data storage and processing to another system in which it can be stored and managed differently.

Automate reliable ETL on Delta Lake

Delta Live Tables (DLT) makes it easy to build and manage reliable data pipelines that deliver high quality data on Delta Lake. DLT helps data engineering teams simplify ETL development and management with declarative pipeline development, automatic testing, and deep visibility for monitoring and recovery.



What is Relational Model?

Relational Model (RM) represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship. The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

Relational Model Concepts in DBMS

1. **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student_Rollno, NAME, etc.
2. **Tables** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
3. **Tuple** – It is nothing but a single row of a table, which contains a single record.
4. **Relation Schema:** A relation schema represents the name of the relation with its attributes.
5. **Degree:** The total number of attributes which in the relation is called the degree of the relation.
6. **Cardinality:** Total number of rows present in the Table.
7. **Column:** The column represents the set of values for a specific attribute.
8. **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
9. **Relation key** – Every row has one, two or multiple attributes, which is called relation key.
10. **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain.

Relational Integrity Constraints

1. Domain Constraints
2. Key Constraints
3. Referential Integrity Constraints

Domain Constraints

Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.

Best Practices for creating a Relational Model

- Data need to be represented as a collection of relations
- Each relation should be depicted clearly in the table
- Rows should contain data about instances of an entity
- Columns must contain data about attributes of the entity
- Cells of the table should hold a single value
- Each column should be given a unique name
- No two rows can be identical
- The values of an attribute should be from the same domain