Program : **B.E**

Subject Name: **Data Science**

Subject Code: **IT-8003**

Semester: **8th**

Data Science IT 8003 (2)

Unit 1

Introduction & Getting the fundamentals of Big Data:

In the previous era a computer have a small amount of memory and as the technology upgraded this demand is increasing in nature. Now a days the demand is so huge that it is not easy to manage that data in an efficient manner with the traditional approaches.

In current era on daily basis enormous amount of data is generated in terms of social media, purchasing sites, weather forecasting, airline, hospital data sets etc. To process this data without thinking of Big Data is beyond the imagination so manage this data and process this data Big Data comes into picture.

The term Big Data is misleading as an impression that after certain size the data is big and upto that certain size the data is small.

Just to refresh the memory, table is given that share the size of data:

Name	Symbol	Data
Kilobyte	KB	10^3
Megabyte	MB	10^6
Gigabyte	GB	10^9
Terabyte	TB	10^12
Petabyte	PB	10^15
Exabyte	EB	10^18
Zettabyte	ZB	10^21
Yottabyte	YB	10^24

Table 1.1: Different Symbol & Memory Size

So the question arise here that from which point Big Data start so if we analyse this it totally depend on the situation, the Bog Data could start from any point there is no fix definition however it is mostly defined this way that Big Data is a data that become difficult to process because of its size using traditional system.

Example:

- If we want to send a 100 MB file via Gmail it is not possible with the traditional system because in Gmail maximum size of an attachment should be less than 25 MB.
- If we want to view a 100 GB image on our normal computer it is not possible to view due to capability of system.
- If we want to edit a 100 TB video via some particular editing software and it is not possible to edit due to limitation of the software.

So the term Big Data is related to the capabilities of the system and at higher level the term is related to Organization.

Evolution of Data Management:

The evolution of data management can be easily understood by the below figure:

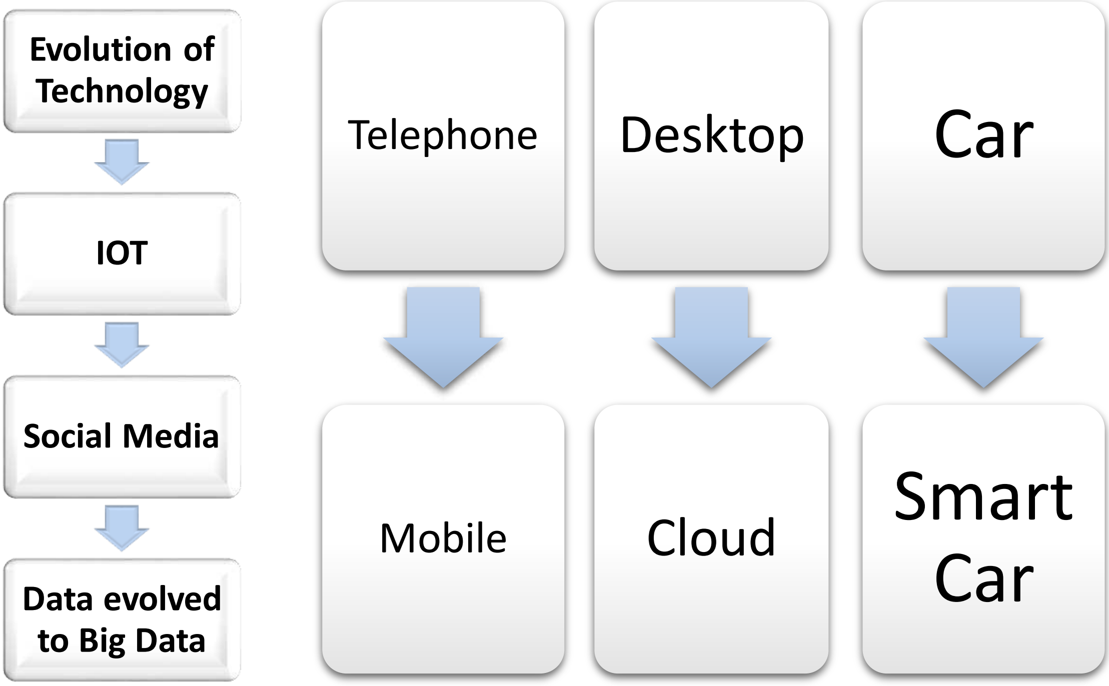

Figure 1.1: Evolution of Data Management

From the figure we can see that how technology is evolved, from telephone to mobile phone with IOS that are making human being life smarter; apart form that we are using bulky desktop where we uses floppy to process the data & now we can store our data to cloud and in the similar way now the car is also converted into smart car that sense the data in the form of environment, weather condition, traffic volume etc. and these advancement in technology numerous amount of data on daily basis.

IOT connects your physical device with internet and makes it smarter. For example Smart Air Conditioner which reads room temperature, outside temperature & body temperature and accordingly decide the temperature of room. In order to do it AC collect data from internet and process the same to function.

Social Media is actually one of the important factor in the evolution of data. Almost every single user use different social media platform and they generate huge amount of data and most challenging task is that it generated unstructured data that is difficult to manage. So here it is not only generating data but also generates different forms of data.

So these are few major examples for evolution of data, there are so many other reason for evolution of data.

Defining Big Data:

Big data is the term for collection of data sets so large and complex that it become difficult to process using on-hand database system tools or traditional data processing application.

'Big Data' is a term used to describe collection of data that is huge in size and yet growing exponentially with time. “Big Data” is the data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it.

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

Big Data Management Architecture:

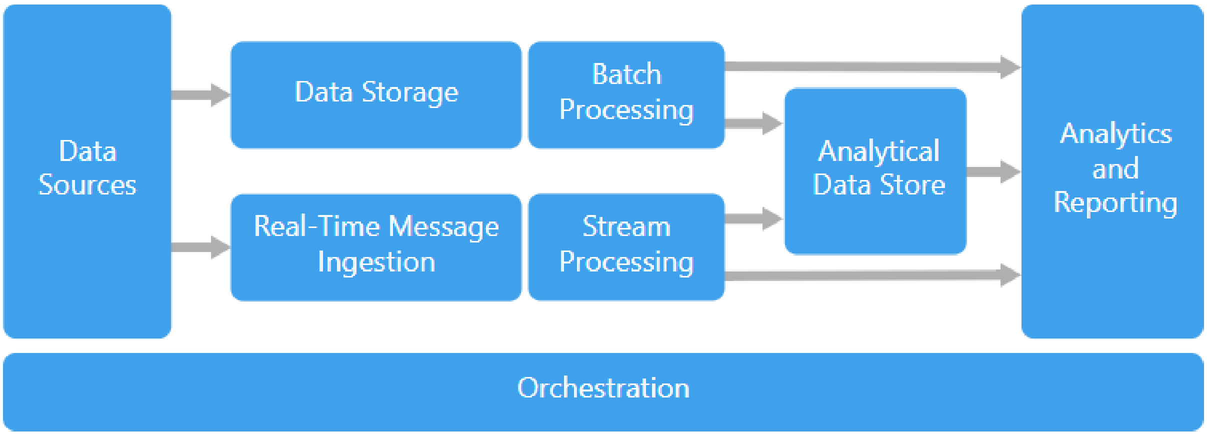

Figure 1.2: Big Data Management Architecture

A big data management architecture is designed to grip the absorption, processing and investigation of data that is too large or complex for traditional database systems.

Big data solutions typically involve one or more of the following types of workload:

- Batch processing of big data sources at rest
- Real-time processing of big data in motion
- Interactive exploration of big data
- Predictive analytics and machine learning

Big Data architecture consists of following components:

Data Sources: All big data solutions start with one or more data sources like Application data stores, such as relational databases; Static files produced by applications, such as web server log files; Real-time data sources, such as IoT devices.

Data Storage: Data for batch processing operations is typically stored in a distributed file store that can hold high volumes of large files in various formats. This kind of store is often called a data lake. Options for implementing this storage include Azure Data Lake Store or blob containers in Azure Storage.

Batch Processing: Because the data sets are so large, often a big data solution must process data files using long-running batch jobs to filter, aggregate, and otherwise prepare the data for analysis. Usually these jobs involve reading source files, processing them, and writing the output to new files.

Real-time Message Ingestion: If the solution includes real-time sources, the architecture must include a way to capture and store real-time messages for stream processing. However, many solutions need a message ingestion store to act as a buffer for messages, and to support scale-out processing, reliable delivery, and other message queuing semantics.

Stream Processing: After capturing real-time messages, the solution must process them by filtering, aggregating and otherwise preparing the data for analysis. The processed stream data is then written to an output sink. Azure Stream Analytics provides a managed stream processing service based on perpetually running SQL queries that operate on unbounded streams.

Analytical Data Store: Many big data solutions prepare data for analysis and then serve the processed data in a structured format that can be queried using analytical tools. Azure SQL Data Warehouse provides a managed service for large-scale, cloud-based data warehousing.

Analysis and Reporting: The goal of most big data solutions is to provide insights into the data through analysis and reporting. To empower users to analyse the data, the architecture may include a data modelling layer, such as a multidimensional OLAP cube or tabular data model in Azure Analysis Services. Analysis and reporting can also take the form of interactive data exploration by data scientists or data analysts.

Orchestration: Most big data solutions consist of repeated data processing operations, encapsulated in workflows that transform source data, move data between multiple sources and sinks, load the processed data into an analytical data store, or push the results straight to a report or dashboard.

Capturing, Organizing, Integrate, Analyse and Act

Capturing the Big Data: There are several sources of Big Data which generate a lot of new data. The unstructured data, now mostly generated through the internet and social media. Text messages, tweets, (blog) posts are increasingly becoming an important source of data relevant for any organization. Capturing unstructured data, traditionally documents and email messages, has been the territory of Enterprise Content Management. Document capture software has been along for some decennia. But can those systems cope with the increasing amounts and number of sources in a Big Data environment?

When we focus on the first of the Big Data challenges, the capture of data, the level of challenge is mixed. It's no real problem to suck in large amounts of data. Well, that been said, this can still pose some major technical issues.

What data do we keep and what data do we discard as redundant, obsolete, trivial and irrelevant. It's all about keeping your collection of Big Data fit for purpose, now and in the future. This also implies that you know what you store. So one of the tasks of capturing is filtering the data to only keep the relevant information.

Organize: A big data research platform needs to process massive quantities of data—filtering, transforming and sorting it before loading it into a data warehouse. Oracle offers a choice of products for organizing data. In addition, Oracle enables end-to-end control of structured and unstructured content, allowing you to manage all your data from application-to-archive efficiently, securely, and cost effectively with the Oracle content management and tiered storage solution designed specifically for research organizations.

Integration: Big data integration is discovering information, profiling them, understanding the data, the value of that data, tracking through metadata, improving the quality of data and then transforming it into the form that is required for big data. Every big data use case needs integration. There are several challenges one can face during this integration such as analysis, data curation, capture, sharing, search, visualization, information privacy and storage.

Analyse: The infrastructure required for analysing big data must be able to support deeper analytics such as statistical analysis and data mining on a wider variety of data types stored in diverse systems; scale to extreme data volumes; deliver faster response times; and automate decisions based on analytical models. Oracle offers a portfolio of tools for statistical and advanced analysis.

Setting the Architecture Foundation:

Big data architecture is the foundation for big data analytics. Architects begin by understanding the goals and objectives of the project, and the insights of different approaches. Thus the architecture foundation needs the right planning and effective tools.

System architects go through a same pattern to plan big data architecture. Meeting with stakeholders to understand organization objectives for its big data, and plan the framework with appropriate hardware and software, data sources and formats, analytics tools, data storage decisions, and results consumption.

The need of big data architecture depends on the pattern & size of data. Single computing tasks rarely top more than 100GB of data, does not require a big data architecture. Unless you are analysing terabytes and petabytes of data on a constant basis to a scalable server instead of a massively scale-out architecture like Hadoop. I

An Individual probably do need big data architecture under the following circumstances:

- To extract information from extensive networking or web logs.
- Process massive datasets over 100GB in size.
- Invest in a big data project, including third-party products to optimize your environment.
- Store large amounts of unstructured data & to summarize or transform into a structured format.
- Have multiple large data sources to analyse, including structured and unstructured.
- Want to proactively analyse big data for business needs.

Performance Matters in Big Data:

Performance testing for big data application involves testing of huge volumes of data, and it requires a specific testing approach.

Big data architecture also needs to perform in concert with the organization’s supporting infrastructure. For example, you might be interested in running models to determine whether it is safe to drill for oil in an offshore area given real-time data of temperature, salinity, sediment resuspension, and a host of other biological, chemical, and physical properties of the water column.

It might take days to run this model using a traditional server configuration. However, using a distributed computing model, what took days might now take minutes.

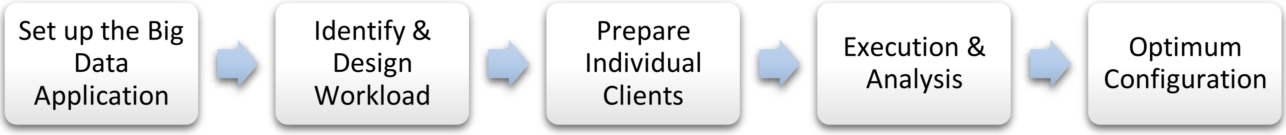
Set up the Big Data Application
Identify & Design Workload
Prepare Individual Clients
Execution & Analysis
Optimum Configuration

Figure 1.3: Performance Testing Approach

Hadoop is involved with storage and maintenance of a large set of data including both structured as well as unstructured data. A large and long flow of testing procedure is included here:

- First of all do the setup of the application prior to the testing procedure begins
- Find out the required workloads and make the design accordingly
- Make ready each and every client separately
- Perform the testing procedure and also check the output carefully
- Do the best possible organization

Big Data Types:

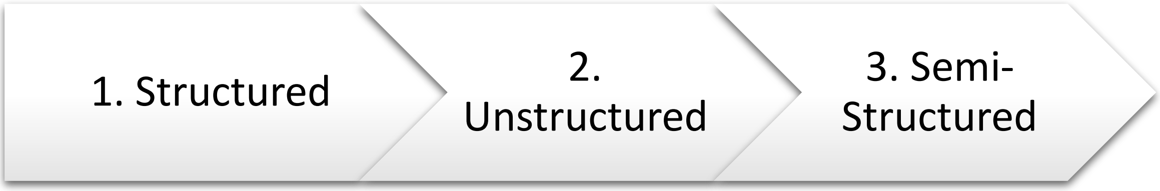

Figure 1.4: Big Data Types

Structured: Structured data, that can be processed, stored, and retrieved in a fixed format. It refers to highly organized information that can be readily and seamlessly stored and accessed from a database by simple search engine algorithms. For instance, the student table in a database will be structured as the student details, their achievements, their semester grades, etc., in an organized manner.

Unstructured: Unstructured data refers to the data that lacks any specific form or structure whatsoever. This makes it very difficult and time-consuming to process and analyse unstructured data.

Example: Output returned by 'Google Search'

Semi-structured: Semi-structured data pertains to the data containing both the formats mentioned above, that is, structured and unstructured data. To be precise, it refers to the data that although has not been classified under a particular repository (database), yet contains vital information or tags that segregate individual elements within the data.

Example: Personal data stored in a XML file

Big Data Characteristics:

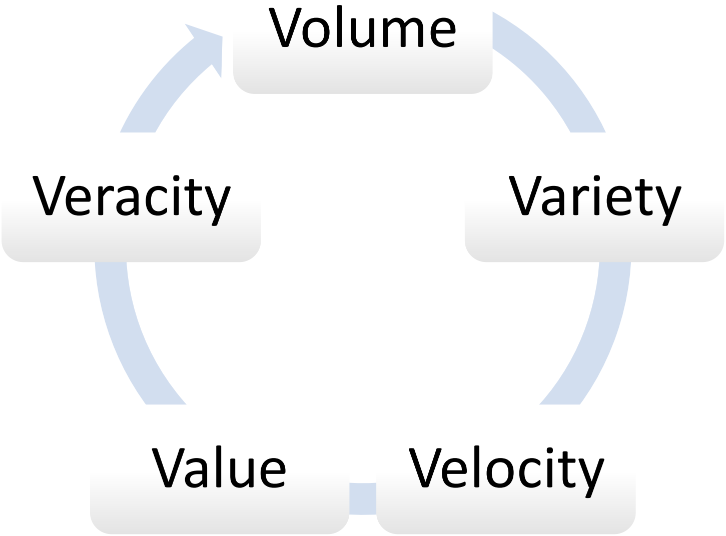

Figure 1.5: Big Data Characteristics

Volume: The data that results into large files. For example social media that is increasingly day by day.

Variety: Processing different kind of data that can be of structured, semi-structured or unstructured in nature.

Velocity: Data is being generated at an alarming rate.

Value: Finding correct meaning of the data.

Veracity: Uncertainty and inconsistencies in the data.

Defining Structured Data:

The term structured data generally refers to data that has a defined length and format for big data. Examples of structured data include numbers, dates, and groups of words and numbers called strings. Most experts agree that this kind of data accounts for about 20 percent of the data that is out there.

Sources of Big Structured Data:

Although this might seem like business as usual, in reality, structured data is taking on a new role in the world of big data. The evolution of technology provides newer sources of structured data being produced often in real time and in large volumes. The sources of data are divided into two categories:

Computer or Machine generated: Machine-generated data generally refers to data that is created by a machine without human intervention.

Example: Sensor data, Web log data, Point-of-sale data, financial data etc.

Human Generated: This is data that humans, in interaction with computers, supply.

Example: Input data, Click-stream data, gaming related data etc.

Role of RDBMS in Big Data:

Big data is becoming an important element in the way organizations are leveraging high-volume data at the right speed to solve specific data problems. Relational Database Management Systems are important for this high volume. Big data does not live in isolation. To be effective, companies often need to be able to combine the results of big data analysis with the data that exists within the business.

One of the most important services provided by operational databases is persistence. Persistence guarantees that the data stored in a database won't be changed without permissions and that it will available as long as it is important to the business.

In companies both small and large, most of their important operational information is probably stored in RDBMSs. Many companies have different RDBMSs for different areas of their business. Transactional data might be stored in one vendor's database, while customer information could be stored in another.

It is not likely you will use RDBMSs for the core of the implementation, but you will need to rely on the data stored in RDBMSs to create the highest level of value to the business with big data.

During your big data implementation, you'll likely come across PostgreSQL, a widely used, open source relational database. Several factors contribute to the popularity of PostgreSQL. It is also available on just about every variety of operating system, from PCs to mainframes.

Defining Unstructured Data:

Unstructured data is essentially everything else. Unstructured data has internal structure but is not structured via pre-defined data models or schema. It may be textual or non-textual, and human or machine-generated. It may also be stored within a non-relational database like NoSQL.

Sources of Unstructured Data:

Unstructured data sources deal with data such as email messages, word-processing documents, audio or video files, collaboration software, or instant messages.

Typical human-generated unstructured data includes:

Text files, Emails, Social Media, Mobile Data, Business Applications, Satellite Imagery, Scientific Data etc.

Integrating Data Types into a Big Data Environment:

Data Integration is the process of transferring the data from source to destination format. Many data warehousing and data management approaches has been supported by integration tools for data migration and transportation by using Extract-Transform-Load (ETL) approach. These tools are widely fit for handling large volumes of data and not flexible to handle semi or unstructured data. To overcome these challenges in big data world, programmatically driven parallel techniques such as map-reduce models were introduced.

Data Integration as a process is highly cumbersome and iterative especially to add new data sources. Traditionally waterfall approach is used in EDW (Enterprise Data Warehouse), where one cannot move to the next phase before completing the earlier one. This approach has its merits to ensure the right data sources are picked and right data integration processes are developed to sustain the usefulness of EDW.

In big data environment, the situation is completely different. Therefore the traditional approaches of integration are inefficient in handling the current situation. So people are expected to do something regarding this issue.

Extract, Transform and Load Process:

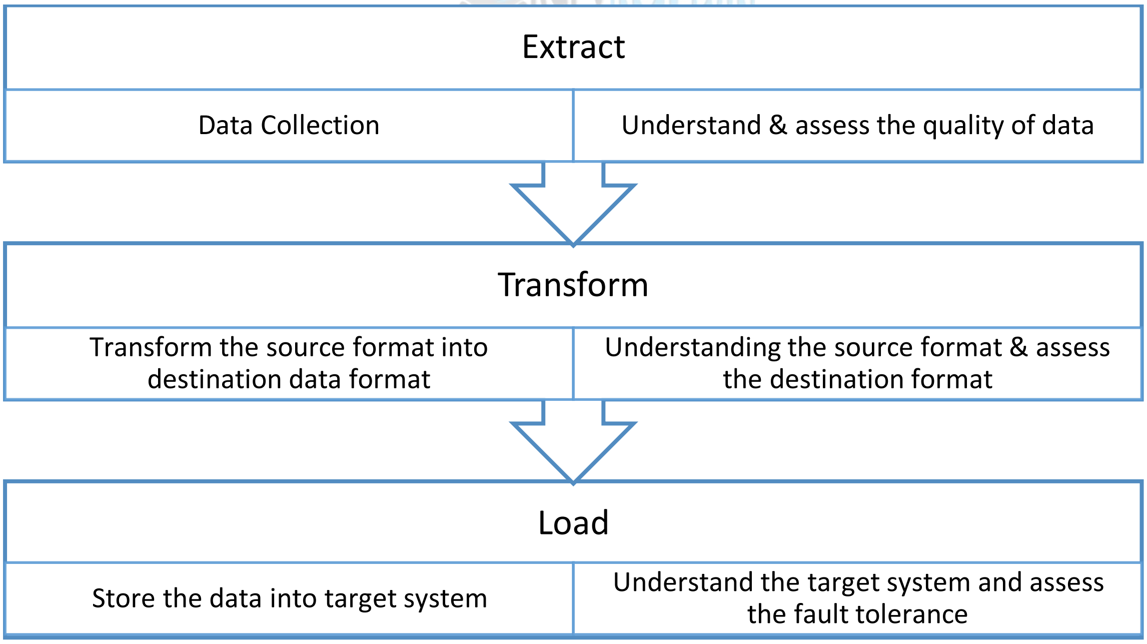
Extract
Data Collection
Understand & assess the quality of data
Transform
Transform the source format into destination data format
Understanding the source format & assess the destination format
Load
Store the data into target system
Understand the target system and assess the fault tolerance

Figure 1.6: ETL Process

ETL comes from Data Warehousing and stands for Extract-Transform-Load. ETL performs the process of loading data from the source system to target system. It is converting data of same type or different types to centralized Data Warehouse which is having standard format for all data.

Extract: The “Extract” task involves gathering data from external sources that needs to be brought to the required systems and databases. The goal of this task is to understand the format of data, assess the overall quality of the data and to extract the data from its source so that it can be manipulated in next task.

Transform: In the “transform” step a variety of software tools and even custom programming are used to manipulate the data so that it integrates with data that is already present.

Load: After the successfully transformation of the source data it is required to physically load it into the target system or database. Before loading the data, it is required to make sure that there is a backup of the current system so that roll back or undo can be initiated in case of failure of the Load process. After loading the data, it is common to run audit reports so that there can be review of the results of the merged databases and systems to make sure the new data has not caused any errors.