

# Introduction to Data Engineering

Data is growing at an unprecedented rate. And it is being tapped by different industries or businesses in different ways. Data Engineering is the backbone that enables Analysts, Data Scientists to understand data.

Let's figure out the thin gap to avoid confusion between Data Science roles and Data Engineering roles.

<b>Data Scientists</b>	<b>Data Engineers</b>
Data scientists are the people who analyze data, create algorithms and make predictions based on that data	Data engineers build the pipelines that collect and deliver data for data scientists
<p>Responsibilities:</p> <ul style="list-style-type: none"><li>• Data scientists will usually already get data that has passed the first round of cleaning and manipulation, which they can use for analysis</li><li>• Building sophisticated analytics programs and statistical methods to prepare data for use in predictive and prescriptive modeling</li><li>• Develop models that can operate on Big Data</li><li>• Deliver results that have an impact on business outcomes</li></ul>	<p>Responsibilities:</p> <ul style="list-style-type: none"><li>• Extract, organize, and integrate data from disparate sources</li><li>• Prepare data for analysis and reporting by transforming and cleansing it</li><li>• Design and manage data pipelines that encompass the journey of data from source to destination systems</li><li>• Setup and manage the infrastructure required for the ingestion, processing and storage of data</li></ul>

## What do Data Engineers do?

Data engineering is at the intersection of software engineering and data science. Having a working knowledge of comparable technologies can help you evaluate the trade-offs between different tools and make appropriate recommendations.

## Skillset

- Infrastructure components
  - Virtual machines
  - Networking
  - Application services such as load balancing
  - application performance monitoring
- Cloud-based services
  - AWS/GCP/Azure/IBM
- Databases and data warehouses
  - RDBMS such - MySQL, Oracle Database, and PostgreSQL.
  - NoSQL databases - Redis, MongoDB, Cassandra, and Neo4J.
  - Data warehouses - Oracle Exadata, AWS RedShift, Snowflake
- Proficiency working with data pipelines
  - Apache Beam, AirFlow, or GCP DataFlow
- ETL tools - IBM Infosphere, AWS Glue, Informatica.
- Languages
  - Query languages (SQL/NoSQL)
  - Programming languages - Python or Java
  - Shell Scripting languages - Linux Shell, Bash
- Big Data processing tools such as Hadoop, Hive, Spark, Kafka

## Where does all the data come from?

### Disparate Sources of Data

The data sources are diverse and dynamic. Data resides in -

- text, images, videos, clickstreams, user conversations
- social media posts, IoT devices' sensor data
- GPS (location) data
- different graphics
- real-time events that stream data
- legacy databases etc.

Okay, Data – Data everywhere! So, what?

### Is it a problem or a good thing?

It's both actually. Precisely, a good thing but with a good price of managing Big Data!

How do we manage different sources?

- When working with so many different sources, the first step is to pull a copy of the data from the original sources into a data repository.  
Challenges: Reliability, Security, and Integrity of the data
- After we pull the data in a commonplace, the next steps would be to get it organized, cleaned up, and optimized for access by end-users.
- We also need to conform to compliances and standards enforced in the organization. **e.g. every fintech company in India has to abide by the rules/regulations of RBI. Recently, RBI published a notification (link) saying that fintech companies should only store the last 4 digits of the card numbers (credit card, debit card) and then companies have to clean their data.**

### Big Data Examples

Today we're dealing with datasets that are so massive and so varied that traditional tools and analysis methods are no longer adequate.

Let me take a few examples to get a hang of it,

#### 1. Aircraft/Flights

Let us say you get hired as a Data Scientist at a leading aviation company. **Do you have any idea how much data you are going to deal with?**

Fact - Data collected by sensors on an aircraft covers more than 300,000 parameters with engine data being one of the most important data points captured. **The average commercial aircraft creates around 20 terabytes of engine information every hour.**

For simplicity, let's take an example of a flight from Lucknow to the Maldives. Multiply by an average **8 hours** flight of flight duration and you're now looking at **20x8x2 TB** of data (don't be silly, aircraft doesn't operate with a single-engine, hence multiplied by 2)

Big Data is keeping the aircraft safe and enables a good flying experience.

## 2. E-commerce

Giants like **Amazon** analyze

- a. demographic, geographic and economic factors;
- b. age, gender, income levels, consumer and lifestyle habits.
- c. every single trace of the customer's digital activity. While we browse through the Internet, buying goods, writing reviews or following people

The models keep learning about our profiles anonymously. Just imagine the scale of data they need to store and then efficiently process it.

## 3. Music Industry

How does **Spotify** recommend a playlist in which all the songs seems so personalized and so well-curated as per our mood?

Again, It's Big Data!

Spotify gathers and builds intelligence about factors like songs' play time, where they are being streamed, what kind of device is streaming, and when they are being played.







All that information provide the music-tech industry with mind-blowing insights to impact listeners' experience.

What are the main challenges involved in handling Big Data?

6 V's of Big Data

# The six Vs of Big Data

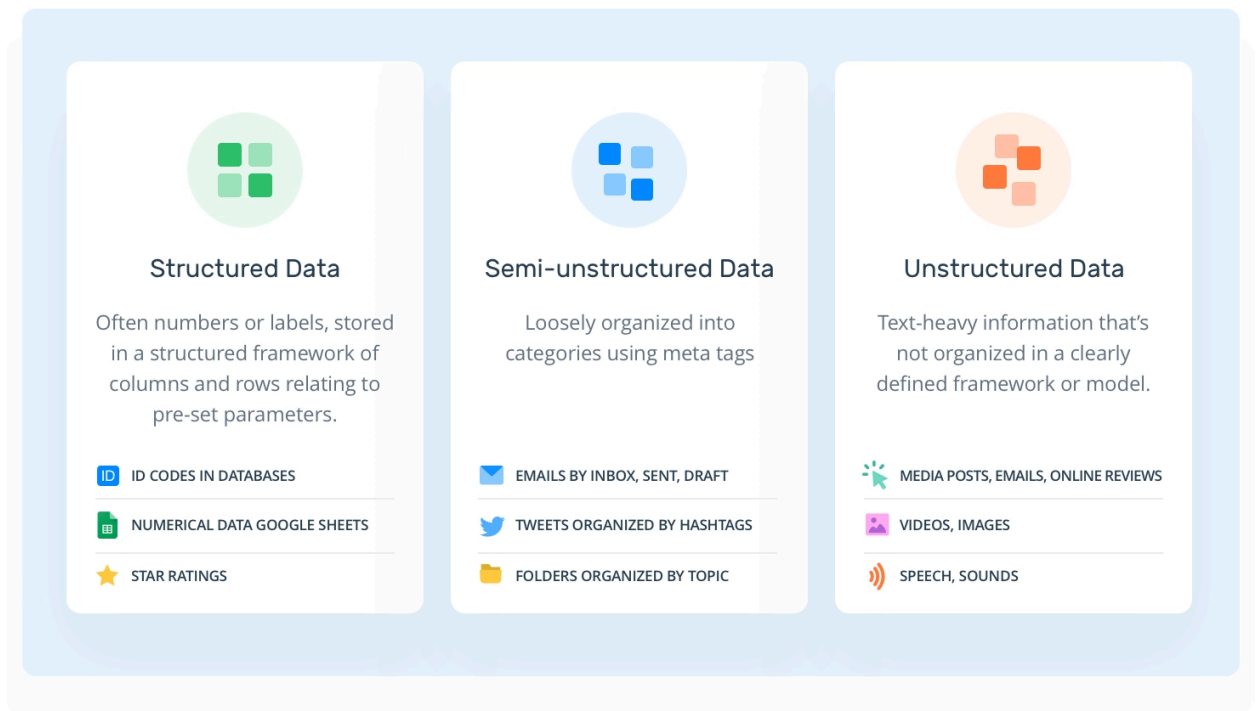
Big data is a collection of data from various sources, often characterized by what's become known as the 3Vs: volume and velocity. Over time, other Vs have been added to descriptions of big data:

VOLUME	VARIETY	VELOCITY	VERACITY	VALUE	VARIABILITY
The amount of data from myriad sources	The types of data: structured, semi-structured, unstructured	The speed at which big data is generated	The degree to which big data can be trusted	The business value of the data collected	The ways in which the big data can be used and formatted
					

Let us try to understand a few key terms with the help of the Aircraft example stated above,

1. **Volume:** In our example, when we said, **320 TB** of engine data, that is precisely the volume of data.  
Our Big Data solutions should be able to ingest, process, and store even large data volumes. This aspect changes rapidly as data collection continues to increase, thus the solution should be scalable for growing business needs.
2. **Variety:** Apart from the engine data as stated, we gather so much more data hence so much variety.

# Unstructured vs Structured Data



For example, data around the passengers and flight,

```
passengers (id, name, gender, age, mobile_num, email_id)
flight (id, source, destination, num_of_seats, onboard_cnt)
```

This is an example of **structured** data!

**Definition:** Data that follows a rigid format can be organized neatly into rows and columns as structured data. It has a well-defined structure and can be stored in well-defined schemas such as relational databases.

Let's take another use case of sending an email when you booked the flight tickets, primary attributes are like,

```
email (sender_email, receiver_emails, subject, body, attachments)
```

Again seems like well-structured data, but is it?

No, actually it's **semi-structured**, because of the different nature of parameters in the schema (it's not that rigid!)

e.g, The 'receiver\_emails' column is very much possibly a JSON, like,

```
{
  "direct": "harshit@scaler.com",
  "cc": "mudit@scaler.com",
  "bcc": [
    "anshuman@scaler.com",
    "abhimanyu@scaler.com"
  ]
}
```

Apart from this, you would have **file attachments** as well. It can be a *.pdf*, *.png* etc.

**Definition: Semi-structured data is a mix of data that has consistent characteristics but data that doesn't conform to a rigid structure. It contains tags and elements, or metadata, which is used to group and organize it.**

- Other types of data in our example could be GPS (location) data, the current latitude and longitude along with other details in native or raw formats are going to be **unstructured** data.
- More examples of unstructured data could be content like
  - photos
  - videos
  - text files,
  - PDFs
  - social media posts

**Definition: Data that does not have an easily identifiable structure and, therefore, cannot be organized in a mainstream relational database in the form of rows and columns can be termed as unstructured data. It does not follow any particular format, sequence, semantics, or rules.**

Being able to leverage the potential of variable data sources and types, structured, semi-structured and unstructured. Integrating diverse data into a manageable structure is key to a robust big data opportunity

3. **Velocity:** Going back to our Aircraft example, we are now clear that we have a different variety of data and in humongous volume. To prevent any unwanted situation (e.g, crash) we should be able to ingest and process the generated data in a real-time manner, to understand it properly and act accordingly.

Another example could be, suppose you are driving but somehow missed a right turn suggested by **Google Maps**. It *instantly* suggests us a new optimal path. It is possible only because Google Maps is gathering and processing our location in real-time!

**How quickly we can ingest and process data is also known as velocity.**

### **Types of Data processing based on velocity**

And, the above 2 examples of Aircraft and Google Maps are examples of **Stream Processing** (a type of Velocity).

- **Definition:** Stream processing allows you to feed data into analytics tools as soon as they get generated and get instant analytics results.
- **Tech:** There are multiple open-source stream processing platforms such as Apache Kafka, Apache Flink, Apache Storm, etc.
- **Use Case**
  1. Social media sentiment analysis (are the people happy-sad about the news, budget etc.)
  2. Log monitoring (analyze the logs generated by a system and send an alert if it encounters words like FAIL, FAILURE, FAILED etc. It will help bring the issue to our attention, in real-time and may prevent system outage)
  3. Fraud detection

If you stream-process transaction data, you can detect anomalies that signal fraud in real-time, then stop fraudulent transactions before they are completed. Savior, isn't it?

The other type of velocity is **Batch Processing**.

**Example:** Let us say your task is to analyze transactional data of a major financial firm over a day, every day so that you can understand any pattern which can help in business/revenue.

This data contains millions of records for a day that can be stored as a file or record etc. This particular file will undergo processing at the end of the day for various analysis that the firm wants to do. It will take a large amount of time for that file to be processed. This is a common example of Batch Processing.



- **Definition:** Batch processing works well in situations where you don't need real-time analytics results, and when it is more important to process large volumes of data to get more detailed insights than it is to get fast analytics results.
- Tech: Prominently, Hadoop MapReduce is being utilized in the industry
- Use case: Payroll, Billing, Orders from customers

To take advantage of geolocation data, perceived hypes and trends, and real-time available market and customer information, we need to process it fast. The data platform system built for it should have such capabilities.

4. **Value:** There is data in abundance. But, not all data points may be relevant for all businesses.

For example,

- Your workout schedule might be useful for Spotify to recommend some Hip-Hop or Heavy Metal songs, but not useful for a stock brokerage company like Zerodha, Upstox.
- Likewise, your lifestyle or expense related data might be useful for E-commerce companies like Amazon. It might be useful for Investment related applications (not very much, though). But, it won't be that much beneficial for companies like Uber, Ola!

**Definition:** Value of data means understanding the potential to create revenue or unlock opportunities through your data. If it is not valuable, then questions should be raised about why and where to store it.

5. **Veracity:** Consider a data set of statistics on what people purchase at restaurants and these items' prices over the past five years. You might ask
  - a. Who created the source?
  - b. What methodology did they follow in collecting the data?
  - c. Were only certain cuisines or certain types of restaurants included?

Answers to these questions are necessary to determine the veracity of this information. Knowledge of the data's veracity in turn helps us better understand the risks associated with analysis and business decisions based on this particular data set.

**Definition:** Being able to identify the relevance, correctness or accuracy of data, and apply it to the appropriate purposes.

Understanding data relevance is key to value.

In short: the truth and authenticity of the data, and what can you do with it? In a sense, it is a hygiene factor. By showing the veracity of your data, you show that you have taken a critical look at it.

6. **Variability:** Consider a soda shop may offer 6 different blends of soda, but if you get the same blend of soda every day and it tastes different every day, that is variability.

The same is in the case for data as well. The context of data changes over time and if it is continuously changing, then it can have an impact on the quality of your data.

**Definition:**

Managing and contextualizing data in a way that provides structure, even in unpredictable and variable data environments.

The above 6 terms [**Volume, Velocity, Variety, Value, Veracity, Variability**] are known as the 6V's of Big Data. These are the certain challenges/problems that lead to the conclusion that the traditional tools and analysis methods may not be able to unlock the true potential of the data being collected hence we should better be using Big Data frameworks to manage it.

## Where do we actually store data? How do we separate data storage based on use cases?

### OLTP vs OLAP systems

- The first thing to do while gathering the data is to think of the place where we can store that data.
- Depending on what we are going to do with that data there are different ways to store it.
- The place where we store the data is called the Data **Repository**.

There are **2 main types of data repositories**,

## OLTP (Online transaction processing)

Let's take an example to understand it,

**Example:** Assume that a couple has a joint account with a bank. One day (after a fight :p) both simultaneously reach different ATM centers at precisely the same time and want to withdraw the total amount present in their bank account.

Clearly, the person that completes the transaction process first will be able to get all the money. The ATM system should be designed to reflect the amount as zero, as soon as any one of the above-mentioned transactions complete and should fail the other person's transaction.

The key to note here is that OLTP systems are optimized for transactional superiority instead of data analysis. Hence, the above example fits well in an OLTP system.

**Definition:** OLTP databases are read, written, and updated frequently, because here the emphasis is on fast processing, because. If a transaction fails, built-in system logic ensures data integrity. It holds ACID properties.

**Tech:** Normalized databases for efficiency (say MySQL)

**Use case:**

- Online banking
- Online airline ticket booking/ movie show booking
- Sending a text message
- Adding an item to the shopping cart

## OLAP (Online Analytical Processing)

Taking up another use case,

When Spotify analyses songs by users to come up with the personalized homepage of their songs and playlist or when Netflix does a movie recommendation analysis, they utilize the data stored in OLAP systems (and not OLTP systems). Because they are optimized for analysis by design.

**Definition:**

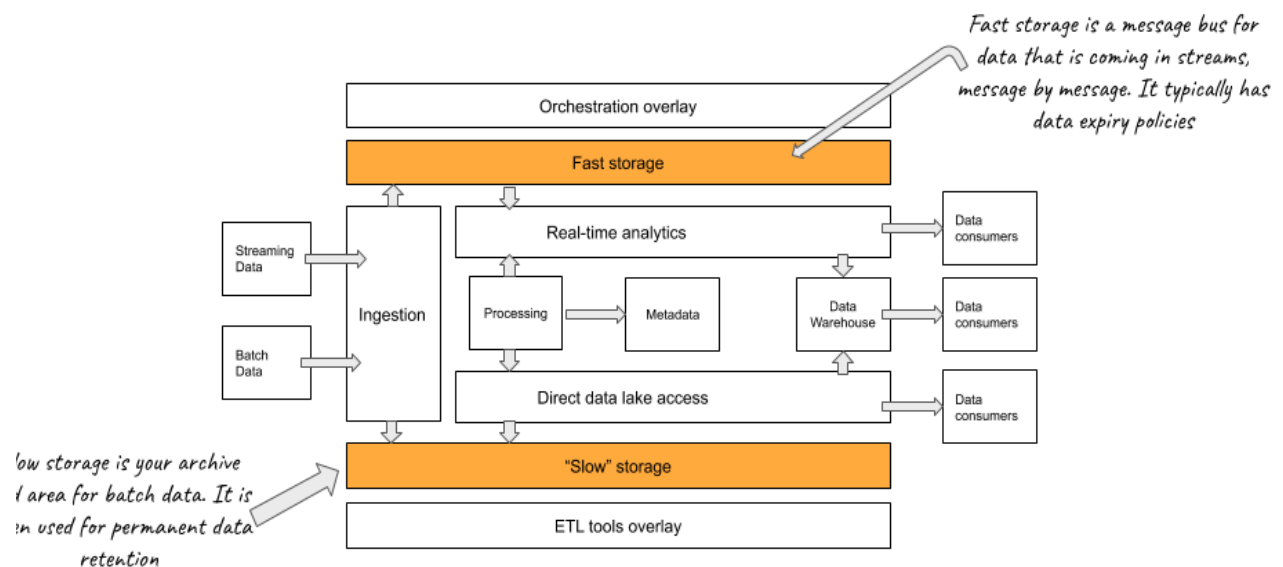
1. OLAP systems provide an environment to get insights from the database retrieved from multiple database systems at one time.
2. It simply pre-calculates and pre-aggregates data to make analysis faster
3. It is based on a multidimensional data model. It includes capabilities like prediction, report viewing, and complex analytical calculations and allows analysts to get the insights of the data in a fast and consistent way

Tech: Data Warehouse, Data Lakes, Delta Lake, Lakehouse, Data Marts etc.

Use case: Any kind of analysis

What are the different services that make up a minimal Data Platform?

## Data Platform Architecture



Let's try to get a birds-eye view of a new user onboarding process followed at CRED!  
Steps -

### 1. **CRED requests the credit details (from CRIF) when users sign up!**

We can say CRED connects to an external data source (CRIF - RBI approved credit bureau in India and brings some data to the internal platform.

This process is called **Ingestion**.

In a data platform design, we need to have a service (layer) for this.

Definition - Ingestion Layer:

- The main task of this layer is to connect to data sources. Transfer data from these data sources to the data platform in streaming mode or batch mode.
- The layer maintains information about the data collected in the metadata repository e.g. How much data has been ingested in how much time and other similar details

Tech: *Google Cloud DataFlow, AWS Kinesis, Apache Kafka* are some examples of the tools or services that can be leveraged for both batch and stream data ingestion.

## 2. **CRED stores the fetched credit history on its internal system**

Once data is getting ingested, it needs to be **stored** somewhere in the system. We need a service (layer) for this,

Definition - The Storage and Integration layer:

- It stores data for processing and long-term use.
- The layer should be designed to make it **reliable, scalable, high-performing**, and also cost-efficient

Tech: *AWS RDS, Google Cloud SQL, MongoDB, Cassandra*

## 3. **CRED processes that data and checks the CIBIL score**

```
if cibil_score > 750:
    print('Congratulations!')
    return 1 # success
else:
    print('Please check back after a few days')
    return 0 # failure
```

Once the data has been ingested, stored, and integrated, it needs to be processed. Thus a new service (layer) is needed here.

Definition - The Processing layer

- Data validations, transformations, and applying business logic to the data are some of the things that need to be done in this layer.
- Other characteristics –
- Ability to read data in both batch and stream modes
  - Applying transformations
  - Supporting popular querying tools and programming languages.

- Ability to scale to meet the processing demands of a growing dataset.

Tech: Examples of the transformation that happens here are – *Structuring of data, Normalization and denormalization and data cleaning*

Ponder over!

1. Do you think that Data Processing and Data Storage always have to be 2 separate layers or they can be combined into one layer as well?
2. Do you think, Data Processing layer can precede the data storage layer? If yes, in what cases. If not, why?

Ans -

It's important to note that storage and processing may not always be performed in separate layers. For example, in relational databases, storage and processing can occur in the same layer, while in Big Data systems, data can be first stored in the Hadoop File Distribution System or HDFS, and then processed in a data processing engine like Spark. And, the data processing layer can also precede the data storage layer, where transformations are applied before the data is loaded, or stored, in the database.

#### 4. **CRED analyses the processed data to derive some insights/patterns**

Although the processing is done, to tweak the user experience and leverage that data for some business decisions which can enable cash inflow, it needs to be analyzed by Data Science or Analytics folks. Thus another service (layer)

Definition - Analysis and User Interface Layer or Visualization Layer:

- This layer delivers processed data to data consumers.
- Data consumers can include Business Intelligence Analysts and business stakeholders who consume this data through interactive visual representations, such as dashboards and analytical reports.

Tech: This layer needs to support the querying languages or tools.

#### 5. **CRED automates the above-mentioned flow of 4 steps**

Since this is the common process CRED follows for all the new user onboarding. The flow is **uniform and repeats** in every signup. Hence, it should be automated. Ah, the last service (layer)

Definition - Data Pipeline Layer:

- Overlaying the Data Ingestion, Data Storage, Data Processing layers and the UI layer is the Data Pipeline layer with the Extract, Transform, and Load the data.
- This layer is responsible for implementing and maintaining a continuously flowing data pipeline.

Tech: There are several data pipeline solutions available, the most popular among them being Apache Airflow.

We just learned about the different layers in a Data Platform Architecture with a business use case. We are going to implement similar pipelines in our Data Labs.

What is an ETL pipeline?

The ETL (Extract, Transform, Load) process helps eliminate data errors, bottlenecks, and latency to provide a smooth flow of data from one system to the other.

### Extract

The first step of this process is extracting data from the target sources that are usually heterogeneous such as an SQL or NoSQL database, a cloud platform or an XML file.

### Transform

- Raw data that has been extracted from the sources need to be transformed into a format that can be used by different applications.
- In this stage, data gets cleansed, mapped and transformed, often to a specific schema, so it meets operational needs.
- Transformation is required to deal with the constraints of traditional data warehouses in which we will load.

### Load

- The process of writing converted data from a staging area to a target system typically a **data warehouse like AWS Redshift, Snowflake**
- Now it is ready to be analyzed by BI tools or data analytics tools

Why can't standalone systems suffice?

## Distributed Systems

Before we start building something like this, we need to have a hang around Distributed systems.

Alas, the **Standalone systems don't fit well** in the ecosystem because of these **limitations**:

1. [Availability](#): is limited to the availability of the piece of hardware it runs on. Continuously running the service leads to unavoidable toil and the system may go down
2. [Durability](#): If we put all our customer's data on a single disk, it's highly unlikely to still have it years after years. To avoid the loss we need to take backups frequently. Though it might prevent a few disk failures, but not against floods, fires, or explosions.
3. [Scalability](#): Distributing a system over many machines gives a lot of flexibility about how to scale it, which is not the case in standalone systems.
4. [Efficiency](#): Distributed systems are efficient in the sense that in case of an unexpected peak in the application usages, we can leverage features like automatic scaling.

## Understanding Instance, Node, Cluster

- **Instance**
  - An *instance* typically refers to a *virtual machine*
  - It represents a single machine in software, but in practice, it shares a *physical machine* (computer) with other instances
  - There can be multiple instances in a single node
- **Node**
  - A single independent computer that is responsible to process or store data.
  - You can configure nodes as per your requirements
    - **Master Nodes** - Govern the tasks among the worker nodes. Configured for reliability
    - **Worker Nodes (Slave nodes)** - Actually, perform the job. If they die, the Master node will assign its task to some other worker node. Data Losses are generally prevented with the help of replication
- **Cluster**
  - A group of nodes interconnected together to process a huge amount of data in a distributed manner
  - Cluster size means how many nodes you have in your cluster. You can add or remove nodes from the cluster.

## [Definition \(Distributed Systems\):](#)



- Also known as distributed computing is a collection of independent components located on different machines or computers at different locations that share messages to achieve common goals.
- As such, the distributed system will appear as if it is one interface or computer to the end-user. The hope is that together, the system can maximize resources and information while preventing failures, as if one system fails, it won't affect the availability of the service.

#### Characteristics:

1. **Resource sharing** - whether it's the hardware, software or data that can be shared
2. **Concurrency** - multiple machines can process the same function at the same time
3. **Scalability** - how do the computing and processing capabilities multiply when extended to many machines
4. **Fault tolerance** - how easy and quickly can failures in parts of the system be detected and recovered
5. **Parallel Processing** - Execution of multiple jobs at the same time, in a real parallel sense

Are all multitasking simultaneous?

Concurrency vs Parallelism

#### Concurrency

1. Concurrency is essentially applicable when we talk about a minimum of two tasks or more. When an application is capable of **executing two tasks virtually at the same time**, we call it a concurrent application.
2. Though, in this case, **tasks look like running simultaneously**, but essentially they MAY not. They take advantage of the **CPU time-slicing** feature of the operating system where each task runs part of its task and then goes to the waiting state.
3. When the first task is in the waiting state, the CPU is assigned to the second task to complete its part of the task.
4. Example - Operating system based on the priority of tasks, thus, assigns CPU and other computing resources e.g. memory; turn by turn to all tasks and give them a chance to complete. To the end-user, it seems that all tasks are running in parallel. This is called concurrency.

## Parallelism

1. Parallelism does not require two tasks to exist. It literally runs parts of tasks OR multiple tasks, at the same time using the multi-core infrastructure of the CPU, by assigning one core to each task or sub-task.
2. Parallelism requires hardware with multiple processing units

Ponder over!

True or False?

In a single-core CPU, we may get concurrency but NOT parallelism. [Ans- True]

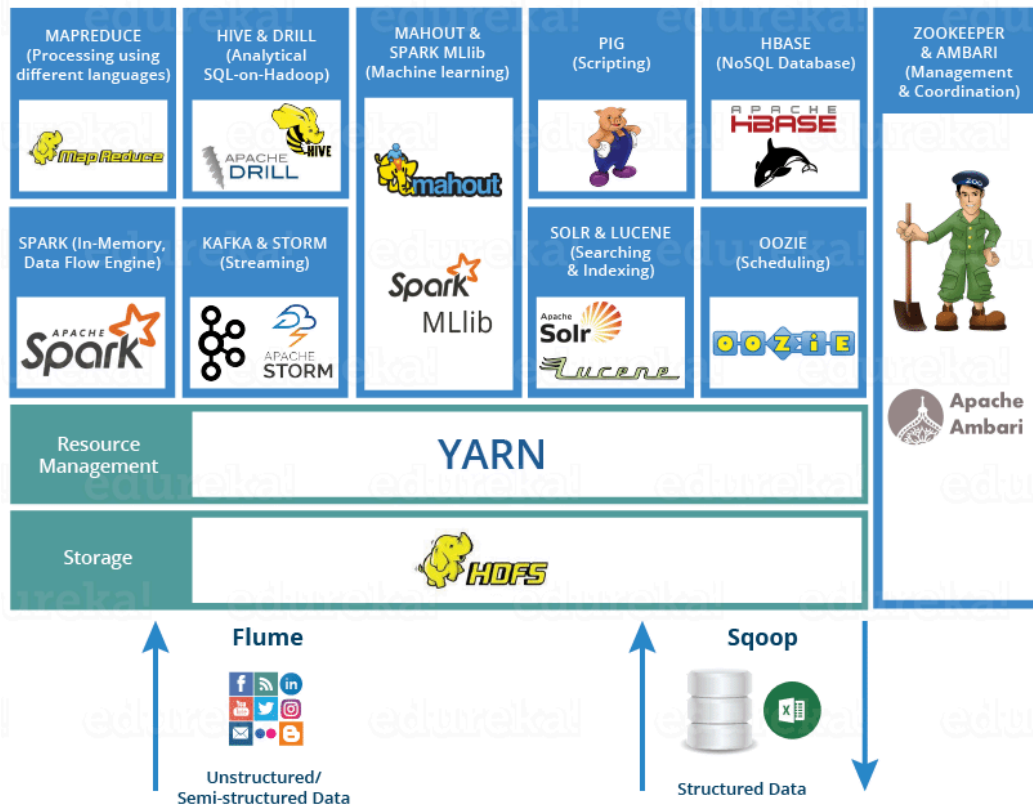
Parallel processing is only possible when we have distributed systems [Ans- True]

## Do we have multiple types of Distributed clusters?

1. Master - Worker
  - a. No communication among the worker nodes, only communication is between master and worker e.g *Hadoop MapReduce*, *Spark* etc.
  - b. Usually, the limitation is of a single point of failure in it, but it can be handled
2. Peer to Peer
  - a. Every node knows the state of every other node
  - b. e.g Cassandra database

How to leverage the Big Data technology into building our own platform?

Modern technologies for the different layers in the Data Platform Architecture!



How can we distribute data storage across systems?

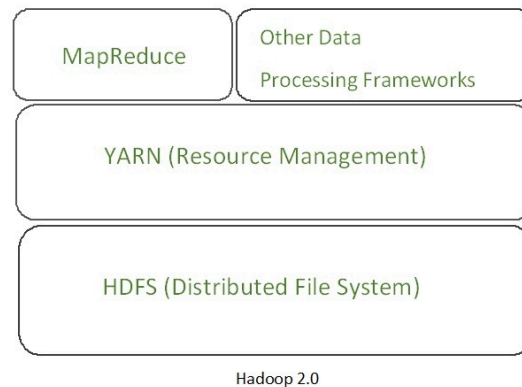
HDFS (Hadoop Distributed File System)

- HDFS is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems.
- However, the differences from other distributed file systems are significant.
  - i. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware.
  - ii. HDFS provides high throughput access to application data and is suitable for applications that have large data sets

Who decides how much resources will be required and their allocation/distribution whenever any task arrives?

YARN (Yet Another Resource Negotiator)

- The fundamental idea of YARN is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons.
- The main components of YARN architecture include
  - Client: It submits map-reduce jobs.
  - Resource Manager:
    - It is the master daemon of YARN and is responsible for resource assignment and management among all the applications.
    - Whenever it receives a processing request, it forwards it to the corresponding node manager and allocates resources for the completion of the request accordingly.
    - It has two major components:
      - **Scheduler:** It performs scheduling based on the allocated application and available resources. It is a pure scheduler, which means it does not perform other tasks such as monitoring or tracking and does not guarantee a restart if a task fails.
      - **Application Manager:** It is responsible for accepting the application and negotiating the first container from the resource manager. It also restarts the Application Manager container if a task fails.
  - Node Manager: It takes care of individual nodes on the Hadoop cluster and manages application and workflow and that particular node. It monitors resource usage, performs log management and also kills a container based on directions from the resource manager.



How to actually break down a giant problem and process them piece by piece?

## MapReduce

- Hadoop MapReduce is a software framework for easily writing applications that process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.
- A MapReduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner.
  - i. The framework sorts the outputs of the maps, which are then input to the *reduced tasks*.
  - ii. Typically both the input and the output of the job are stored in a file system.
  - iii. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Why do we need another processing engine besides MapReduce?

## Spark

- Apache Spark is a lightning-fast unified analytics engine used for cluster computing for large data sets to run programs parallelly across multiple nodes. It is a combination of multiple stack libraries such as SQL and Dataframes, GraphX, MLlib, and Spark Streaming.

- The most vital feature of Apache Spark is its processing speed. It permits the application to run on a Hadoop cluster, up to one hundred times quicker in memory, and ten times quicker on disk.
- Can process in both stream and batch processing modes.
- Spark Lazy Evaluation plays a key role in saving calculation overhead. Since only necessary values get computed.

## Curriculum To be Covered:

- SQL/NoSQL Database :** Playing with SQL/NoSQL DB and evaluating their use cases.
- Data Modeling and Data Warehousing :** Focus on creating fact,dimension tables, SCS with cons and pros of storage types
- Batch pipeline :** Showcasing how big giants perform their reporting, analyze data on dashboards and then making data driven decision
- Real time Streaming Pipeline:** Creating our own crypto dashboard tracking bitcoin fluctuations.
- Orchestration with Airflow:** once we have relevant workflow, coordinating the execution and monitoring of these workflows will be done via airflow
- Git and github :** Pushing our code to the version control system to keep track of code changes and to collaborate on code.
- Docker:** Eventually we will automate the deployment of applications in lightweight containers so that applications can work efficiently in different environments

## Question to be asked in class :

Qn 1: Under which category does AVRO,Parquet files lie?

- Structured Data
- Unstructured Data
- Semi Structured Data
- Hybrid Structured Data

Qn 2: What Do Data Engineers do?

- Extract, organize, and integrate data from disparate sources
- Prepare data for analysis and reporting by transforming and cleansing it
- Design and manage data pipelines that encompass the journey of data from source to destination systems
- All Of the above

Qn 3 : In Batch Processing Data Size is?

- Known and infinite
- Unknown and infinite
- Known and finite (right answer)
- Unknown and finite

Qn 4: "It is the level or the intensity by which the data is reliable, consistent, accurate, and trustworthy." Which V in big data identifies this?

- Value
- Veracity
- Variability
- Velocity

## Questions for Assignments:

Q1)The Processing which allows you to feed data into analytics tools as soon as they get generated and analyzes the data in real time is?

- Batch Processing
- Stream Processing (right answer)
- Distributed Data Processing
- Multi-Processing

Q2) OLAP based on?

- One Dimensional data model
- Two Dimensional data model
- Multidimensional data model (right answer)
- All of the above

Q3) Which layer is responsible for collecting data from various data sources and transferring it in streaming mode or batch mode?

- Ingestion Layer (right answer)
- Processing Layer
- Storage Layer
- Data Pipeline Layer

Q4) The nodes in the distributed systems can be arranged in the form of?

- master - worker systems
- peer to peer systems
- Both A and B (right answer)
- None of the above

Q5) If you're adding a comment in someone's instagram picture then which type of data repositories is involved?

- OLTP (right answer)
- OLAP
- Both
- None of the above

Q6) Suppose A retail company has been collecting data of sales, customer demographics, and inventory for several years. Now The company wants to use this data to generate reports and visualizations that will help them to make better decisions and increase their sales. Your team is tasked to implement a system that can handle large amounts of data and complex analytical queries. Which system you'll implement?

- OLTP systems
- OLAP Systems (right answer)
- HDFS
- None of the above

Q7) You are working as a data engineer at company 'xyz' and are provided to process verified data that is originally in multiple files and formats (like CSV and TSV).



Which V out of the 6 V's of Big data characteristics poses a challenge here?

**Note:** In this particular case, the organization has enough resources to handle the volume of big data, so it's not a challenge here.

- Velocity
- Volume
- Value
- Variety (right answer)

Q8) Suppose you are playing a multiplayer online game with your friends. The game has started. Now Everyone is interacting with the game as well as with each other simultaneously. So what type of application would be best suited for this?

- Parallel Applications
- Concurrent Applications (right answer)
- Sequential application
- Distributed application

Q9) You are working at a logistics company. Your manager has asked you to get insights from raw data about different transactions, customers feedback, orders, etc, and store this data in a structured format on the cloud.

Which option correctly represents the steps from the ETL pipeline to be followed here?

- Extract
- Transform
- Load
- All of the options (right answer)

Q10) Suppose you are working at a retail company and your manager wants you to analyze "customer purchase data" to identify patterns and trends. They have collected data on millions of customer transactions over the past year.

Which of the following best describes the characteristics of the data that your company has collected?

- Volume (right answer)
- Variety
- Velocity
- Veracity