Data Engineering Episode 1

Subscribe My YouTube channel - 11 Data Street

Big Data - The Wider Lens

Q- What is Big Data?

There are many definitions of Bid Data- > Data that cannot be stored/processed on a single machine.

Ex- let's say we have one big video file of 2 TB and machine storage is only 1 TB so now its not possible to store this file into the 1 TB storage machine so this can be considered as Big data.

formal definition of big data ->

"Data that is identified by 3V's / 5V's is considered to be Big Data"

5 V – Volume, Variety, Velocity, Veracity, Value

1. **Volume** - Data that is so huge that it cannot be stored / processed on a single conventional system is considered as Big Data.
   Ex- one big file which can not fit into a single machine.

2. **Variety** - Data can be coming in different formats and we must handle it.
   Mainly there are 3 types of variety of data –
   A. **Structured** - Data in such formats have some schema associated with it.
      Ex- Database tables

Sample table: salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14

   B. **Semi-Structured** - Data in such formats have some partial schema associated with it and are not stored in the form of rows & columns. Ex- JSON, XML

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

C. Unstructured - Data in such formats have no schema associated with it. Ex- Image, Log files, Videos.

Sample log file-

```
[2010-04-24 07:51:54,393] DEBUG - [main] BulkOpsClient.main(): List of all
configurations loaded: {numofthreads=1, impstatchkinterval=30, maxloginattemp
manifestfiledir=.\Manifest\, sessionkeepchkinterval=300, routingurl=https://
sso.crmondemand.com, hosturl=http://sdchs20n263.us.oracle.com, testmode=debug
maxthreadfailure=1, logintimeoutms=180000, csvblocksize=1000, maxsoapsize=102

[2010-04-24 07:51:54,393] DEBUG - [main] BulkOpsClient.main(): List of all op
loaded: {password=*********, clientloglevel=detailed, waitforcompletion=False
datetimeformat=usa, importloglevel=errors, datafilepath=.\\data\\account1.csv
operation=insert, help=False, mapfilepath=.\\data\\account1.map,
clientlogfiledir=., recordtype=account, duplicatecheckoption=externalid,
username=oracle/oracle, csvdelimiter=,}
```

3. **Velocity** -The speed at which new data is coming in. for example if it's a live stream of camera then it's a high speed of data, temperature sensor data for every millisecond.

4. **Veracity** Is the quality or correctness of Data. Data cannot always be in the right format, and we should be able to handle this not so high-quality data.
   Data can have some null or missing values in it so need to handle such data.

5. **Value** Should be able to generate some value out of the data collected. The data must generate some meaning or business insights from it.

Q- why we need to learn a new technology to handle the big data why not use existing technologies?

The traditional / conventional technologies are not capable of handling Big Data.

To handle data, we need to have 3 things, we also call them resources:

1. Storage – Hard disk, let's say 1 TB storage.
2. Compute – CPU cores ex- 4 cores or 8 core CPU
3. Memory – RAM ex- 8 GB RAM or 32 GB RAM

All together Storage, compute and memory are called resources.

To understand why, lets first know what is –

Monolithic Vs Distributed System:

Meaning of Mono is single.
In **monolithic system** all the functions or services are integrated into a single unit. now if one single component gets fail then the whole system will get fail.

Ex- mobile phone or laptop are monolithic systems as all the services are packed in single unit only.

Now consider we have a laptop with 1 TB storage, and we need to handle 2 TB data now in this situation we need to increase the storage to 2 TB storage.

If tomorrow we need to handle 4 TB data, then again we need to increase the storage to 4 Tb storage for laptop and so on.
If means whenever we need to handle double of data, then we need to increase the resources also double like storage from 2 TB to 4 TB storage.

So here to increase the resource the problem is like whenever we need to handle double data then we add the double resources and expected is like performance also will get doubled every time we add the double resources.

But every time we double the resource till some extend the performance gets double but at a point performance will not get double in this case, we just cannot keep adding the double resource.

This is a big challenge in case of monolithic system.

Monolithic systems are not scalable systems as the performance does not increase in the same ratio as resources.

Monolithic follows Vertical Scaling of increasing the computing power of one Machine only by increasing the resources of that system.

distributed systems all the functions and services are integrated in a single unit. different services keep run in as a separate unit so if one service will get fail then generally it will not affect other services and as a whole system will not get fails.
Ex- consider a movie streaming platform there 3 functions are there like one function is to send movies, another function will handle the users and another function handling the payment services.

So, consider if payment service is down then still sending movies and handling users function will keep working only payment service will get effect

Distributed systems Is a cluster of systems grouped together with each holding its own resources.
Computing power of such a system is a sum of computing power of all the nodes in the cluster

Ex - 5 Node cluster, with each Node/ System holding 4CPU core, 8GB RAM and 1TB Hard Disk.

In case you need to scale up such a system, you just need to bring in more Nodes/ Systems.

Distributed System follows Horizontal Scaling that involves increasing the number of Nodes/Systems. Such a scaling is True Scaling because if you double the resources, the performance will also double proportionately.

All the Big data Systems are based on Distributed architecture.

To design a good Big Data System, the following 3 things need to be considered.

1. **Storage** - Where will such huge data be stored as traditional systems cannot handle such data. That is, we need to have Distributed Storage.

2. **Processing / Computation** -
Traditional systems will work when the data resides on a single machine where it must be processed.
However, Big Data is stored / distributed across the cluster on multiple nodes. Therefore, the traditional programming approach doesn't work, and we need Distributed Processing.

3. **Scalability** - The system should accommodate the growing demands. Therefore, a scalable model is required. A Distributed Storage and Processing System by default supports Scalability.

3 factors for designing a Big Data System

1.Where is the data going to be stored?
2.How is the data going to be processed?
3.Is the system Scalable?

Understanding Hadoop:

Hadoop is collection of tools and technologies to solve big data problems.
Hadoop is a framework designed to handle the big data problems.

2003 – The Birth of the Idea: Google Papers

- Google published two influential papers:
  - The **Google File System (GFS)** paper in 2003, which described a scalable distributed file system.
  - The **MapReduce paper** in 2004, which outlined a programming model for processing large data sets in parallel.
- These papers inspired developers to create similar tools in the open-source ecosystem.

2008 – Hadoop 1.0 Release

- Yahoo! adopted Hadoop and contributed to its development. Yahoo! built one of the largest Hadoop clusters with 10,000 nodes.

2011 – Hadoop 2.0 and YARN

- Hadoop 2.0 introduced **YARN (Yet Another Resource Negotiator)**, which separated resource management and job scheduling from the MapReduce engine.

Current version Hadoop 3.0 was officially released in **2017**.

Now new tools like Spark got popularity for faster processing but Hadoop's HDFS and YARN remain foundational in big data ecosystems.