Lambda Architecture Assignment

1. Explain the factors leading to Big Data. List & explain major sources of Big Data.

→ The rise in technology has led to the production and storage of Large amounts of Data. & Affordibility of smartphones.

→ Earlier MetaBytes of data were used but nowadays Petabytes of data are used for analyzing, discovering new facts & Knowledge.

→ Traditional systems were not built to handle large amount of data so there is a need for new systems to analyze the fast growing complex and unstructured Data.

→ The Big Data is a high-volume, high-velocity, high variety & a high veracity im formation set.

The Major Sources of Big Data are:

→ Social Networks & Web Data: Such as Facebook, Twitter, emails blogs & Youtube. On an average Per day Billions of Bytes of data are produced by social Networking Sites making it a large contributor of Big Data.

→ Transaction data & Business Processes: Such as credit card, transactions, flight bookings etc & Medical records.

→ · Machine generated Data: From IoT sensors, Home automation, traffic sensors etc. Data from computer system like Logs, web logs etc

→ Human-generated Data: biometrics data, human Machine interaction data.

2. List and explain the charctristics of Big Data.

→ Volume: It represents the big size of the data. It comes from large piece of data or collection of small data over a peroid of time.

→ Velocity: The term velocity refers to the speed of generation of data. Velocity is so fast to meet the demands & the challenges of processing Big Data, the velocity of generation of data plays a crucial role.

→ Variety: Big Data comprises of variety of data. Data is generated from multiple sources in a system. This introduces variety in data & therefore introduces 'complexity'. It is caused to the availability of large number of hetrogenous platforms in the industry.

→ Veracity: It defines the quality of data captured, which can vary greatly, affecting its accurate analysis.

3. List & explain the major challenges of Big Data Systems.

→ Scaling up & down Big Data according to current Demand.
* Due to data verity it is difficult to create common storage & dificult to integrate.

→ Collecting and Integrating massive & Diverse Datasets.
* Collection of data might be difficult as a lot of institutions would refuse to share & available data consists of diverse datasets. Finding a common ground to integrate would be difficult.

→ Picking the Right NoSQL Tools:
→ Depending on the type of data choosing the right NoSQL tools can be difficult as it might take trail & error approach.

→ Maintaing Data Integrity, Security and Privacy:
→ Securing the Data for the analyzis can be tough as Some ~~has~~ data can be sensitive like patient information.

→ Overcoming Big Data talent & Resource constraint.

④ Due to increasing tools & technologies in big Data it is hard to find the Right talent as people are still workingon traditional methods.

④ To Maitain & process data of large volume it is near impossible due to hardware constraints

4. Discuss the problems faced by traditiond database Systems.

→ Big data is too big for traditional storage

④ Though in theory traditional database systems can handle large amounts of data. It simply can't Keepup with demands of mordon data to deliver the effiency & insights we need.

→ Scaling up vs scaling out.

→ traditional data storage Systems are centralized, So we are forced to Scale up which is less resource effiont than scaling out.

→ Big data is too compler for traditionalstorage:

→ Since big data consib of more than rows & colums Some may be structured but large part of it is unstructured. which can be an issue as they can't properly catogrize it.

→ Big data is too fast for traditional storage.
→ An RDBMS isn't designed for rapid fluctuations. But Big data grows almost instantaneously & analysis needs to occur in real time. for example senson data, which produce at concurrent rate. which maybe dificult to analyze.

5. Discuss the required Properties for Big Data Systems.

→ Robustness & error tolerance:
→ Systems are required to behave in a right manner despite machines going down randomly. Robustness of big data System is the solution to overcome the obstacles associated with it.

→ De buggability: A system must be debug when onfair thing happens by the required information. delivered by the big Data system

→ Scalability: It is the tendency to handle the performance in the context of growing data & load by adding resource to the system.

→ Generalization: It is important to generalize a wide range of application can be function in a general system.

→ Ad hoc queries: Every large dataset contains an anticipated value in it. The ability to perform ad hoc queries on data is significant.

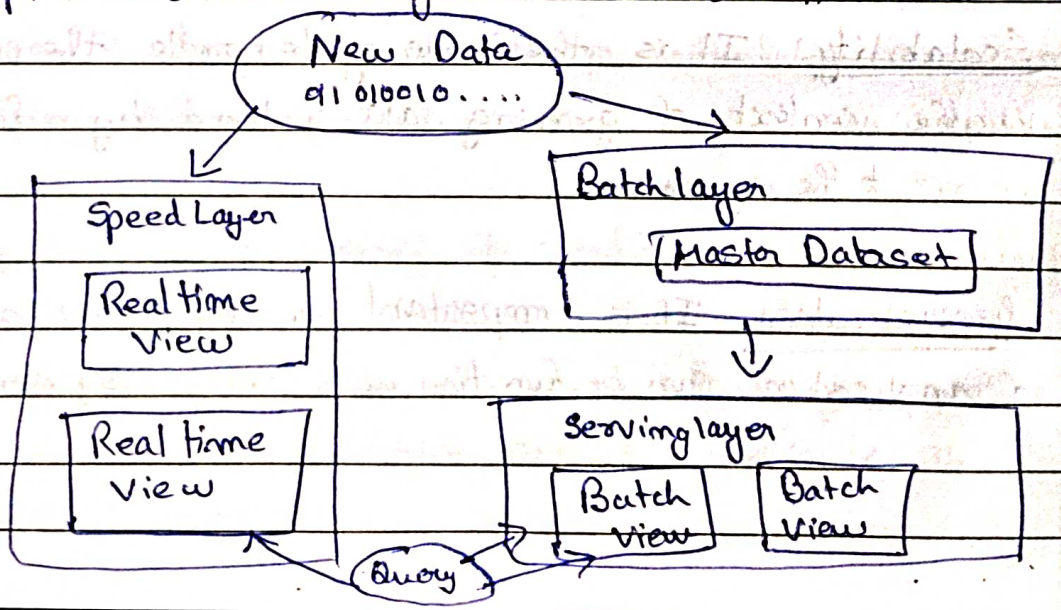
→ extensivity: Extensible system enables to function to be added cost effectively.

→ Low latency reads & updates: In big data system, there is a need of applications low latency on updates propogated shortly

→ Minimal maintenance: selecting components with probably little complexity plays asignificant role in minimal maintenace.

Most of these problems are solved by Lambda Architecture.

6. Explain diffent layers of Lambda Architecture.



classmate
Date
Page

Speed Layer: Batch Layer:

→ The batch layer stores the master copy of the dataset & precomputes batch views on that master dataset which can be thought of as a very large list of records.

→ Batch layer should be able to store an immutable, constantly growing master dataset and,
→ compute arbitarily functions on the dataset.

Serving Layer:

→ The batch layer emits batch views as the result of its functions.
→ The serving layer is a specilized distributed database that loads in a batch view & makes it possible to do random reads.

Speed layer:

→ The serving layer updates whenever the batch layer finishes precomputing a batch view.
→ The goal is to ensure new data is represented in query functions as quickly as needed for the application requirement.

7. Differentiate between re-computation algorithm & increment algorithm.

re-computation algorithm	Increment Algorithm
→ When new data arrives it throws away the old batch views and recomputing functions over the entire master dataset.	→ When new data arrives it will update the batch views directly when new data arrives
→ It will update the count by first appending the new data to master dataset & then counting all the records from scratch.	→ It will count the number of new data records & add it to the existing count.
→ Lower efficiency compared to Increment Algorithm	→ Higher efficiency compared to re-computation Algorithm.
→ Extremely tolerant of human errors	→ Doesn't facilitate repairing errors in the batch views.
→ Requires computational effort to process the entire master dataset	→ Requires less computational ressources. but may generate much larger batch views.

8. List the requirements & responsibilities of batch layer.

Responsiblites of the Batch Layer:

→ **Data Processing:** Batch Layer processes large volume of data in batches. It is designed to handle comprehensive computations & genrate results.

→ **Data Storage:** It relies on immutable storage systems which helps in ensuring data consistency & reliability

→ **Consistency & Accuracy:**
→ Accurate Results ensure high accuray in data processing by relying on complete datasets & executing through computations.

Requirements of Batch layer:

1. **Scalability:** It must be able to scale horizontally to handle large volumes of data.

2. **Fault tolerance:** It should be designed to recover gracefully from failures.

3. **Performance:** The system should efficiently perform large scale data computations.

4. **Data Intergity:** Ensuring that the results produced are consistent & accurate, even in the face of failures.

9. Requirements of serving layer in Lambda Architecture.

→ Batch writable: The batch views for a serving layer are produced from Scratch. When a new version of a view becomes available, it must be possible to completely swapout the older version.

→ Scalable: A serving layer database must be capable of handling views of arbitary size.

→ Random reads: A serving layer database must support random reads, with indexes providing directaccess

→ Fault tolerant - Because a serving layer database is distributed it must be tolerant of machine failures.

10. With example show how low latency & high throughput can be achieved in serving layer of lambda architecture

→

11. List the requirements & responsiblites of speed layer

→ Random reads - A realtime view should support fast random reads to answer qu'os quickly. This means the data it contains must be indexed.

→ Random writes - To support incremental algorithms, it must also be possible to modify a realtime view with low latency.

→ Scalablity: As with the saving bygo views, the realtime views should scale with amount of data they store and the read/write rates required by the application.

→ Fault tolerance: If a disk or a machine crashes, a realtime view should continue to function normally.

12. Differentiate between batch and speed layers.