Big Data Analysis - BDA class notes

Big Data Analytics (University of Mumbai)

Date / /
Page

* Difference between Small and Big Data.

Small Data	Big Data
① Structured (in row & col)	① Unstructured. (any kind of data).
② stores data in GB / MB / TB	② Peta PB / Exa EB / Zeta ZB PB = 1000 of TB EB = 1000 of PB
③ Increases gradually	③ Increases exponentially.
④ SQL & Oracle	④ Hadoop, Spark
⑤ Single Node (centralized system)	⑤ Multinode cluster.

** imp

[Q] Big data characteristics.

- 3V's (Volume, Velocity, Veracity) (Variety, Value)

1) Volume :- large amount of data.
2) Velocity :- speed of using data.
3) Veracity :- ignore the missing.
4) Variety :- different types of data.
5) Value :- finding the current meaning (value) of data

** imp [Q] Types of Big data

(1) Structured (Table format eg. SQL, RDBMS etc)
(2) Semi structured (JSON, XML, YML, NoSQL) (Key: value)
(3) Unstructured

3. Semi structured

XML → eg: text based markup language

```
<Programmer Details>
<FirstName> John </FirstName>
<lastName> Doe </LastName>
<coding Platforms>
<codingplatform = "FAV"> coded Era </codingPlatform>
</programmeDetails>
```

YML → eg: Yet another markup language

```
Emp 1 : .Yaml 1
name : .Yml
name : Mary Jacob
Job : Developer
skills : Java
         Go
         Rust
```

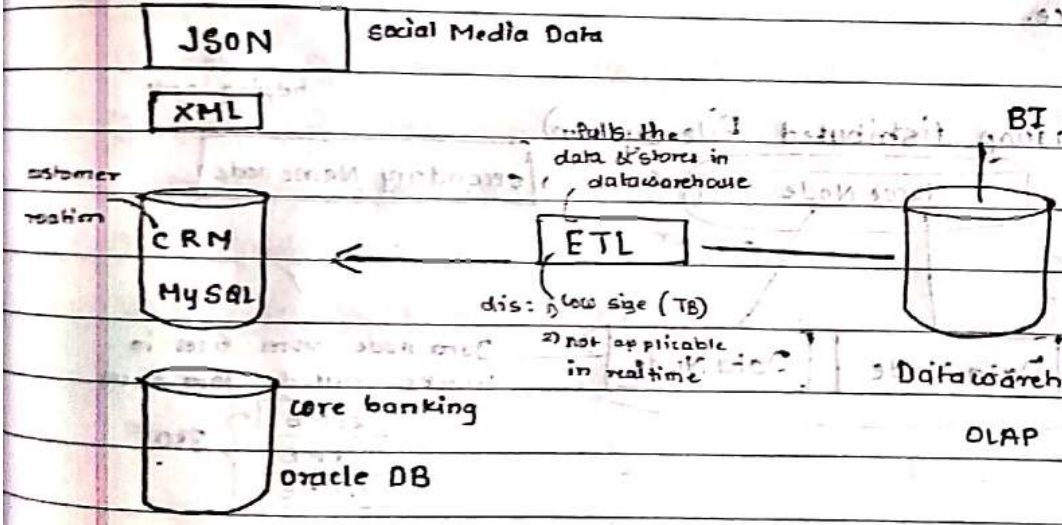
★

[Q] Why Big Data ?

Traditional Data	Big Data
① It is generated in Enterprize level	① Generated outside the Enterprize level
② Its volume ranges from GB to TB	② It's volume ranges from PB to ZB
③ Traditional DB system deals with structured data.	③ Big data deals with semistructured unstructured.
④ Generated per hour / per day / or more.	④ Generated more frequently mainly per seconds
⑤ Size of data is very small	⑤ Size is more than tradition data

⑥ Tradition data is stabled and inter	⑥ Big data is not stable & unknown relationship.
⑦ Easy to manage & manipulate the data.	⑦ Difficult to manage & manipulate data.
⑧ Data sources include ERP Transaction data, CRN Transaction data, financial data, web transaction	⑧ Data sources social media, sensor data, video, audio, images.
⑨ Traditional data source is centralized and managed in centralized form	⑨ Big data source is distributed and managed in distributed form.
⑩ Data integration is easy	⑩ Data Integration is difficult.

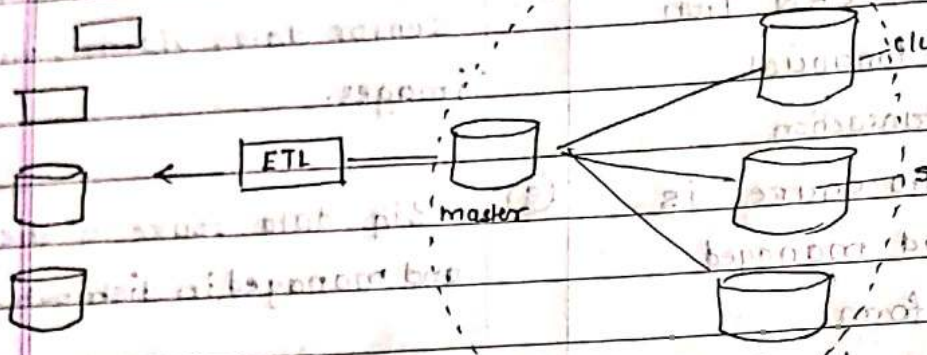
[Q] Extract, Transform, Load



Traditional Enterprize architecture

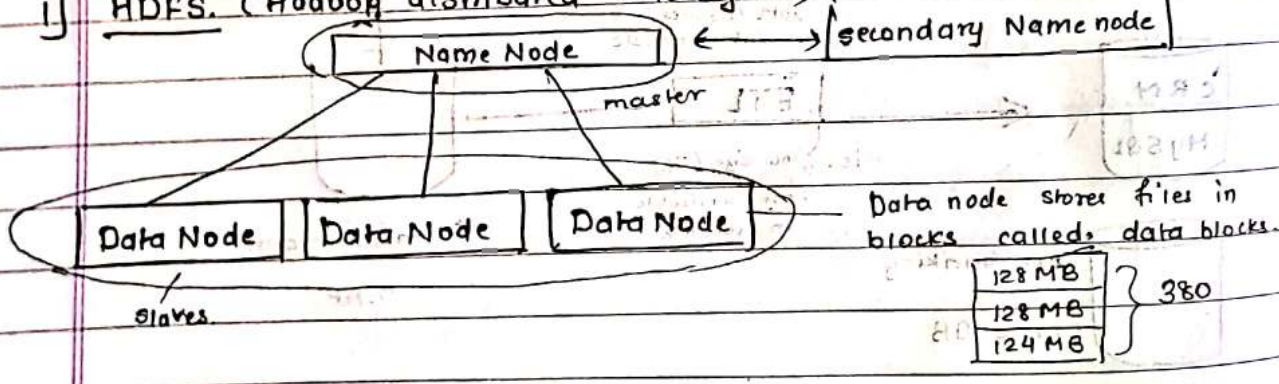
Tools in ETL → 1) informatica
2) oracle

* MPP (Massive Parallel Processing)
- hadoop - cloud era
- Apache (using this download hadoop V 2.6)
- hadoop use Master/slave Architecture



* Hadoop is divided in components.
1) HDFS
2) YARN
3) MAR Reduce

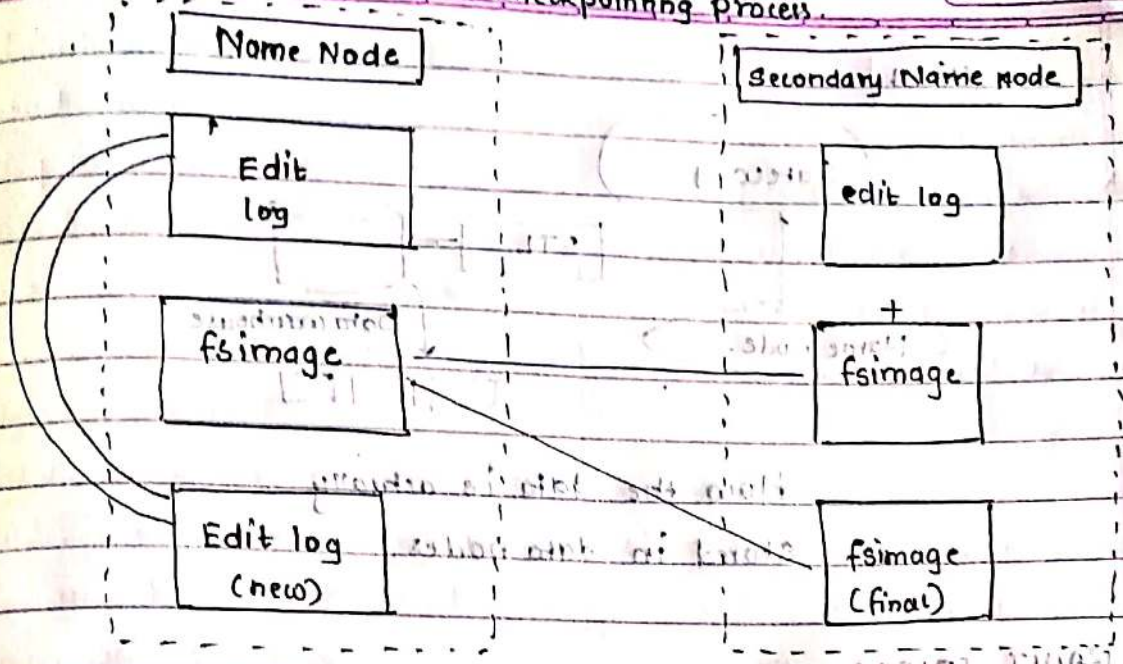
1] HDFS. (Hadoop distributed File system)



- master manages the Datanodes and checks for the file related information.
- It checks for the data nodes be working or are either dead.
- Secondary Node is also called as helping node.

Date / /
Page

Checkpointing Process.



fsimage → file system image

checkpointing → proces of combining fsimage & Edit logs

- fs image cannot be continuously updated

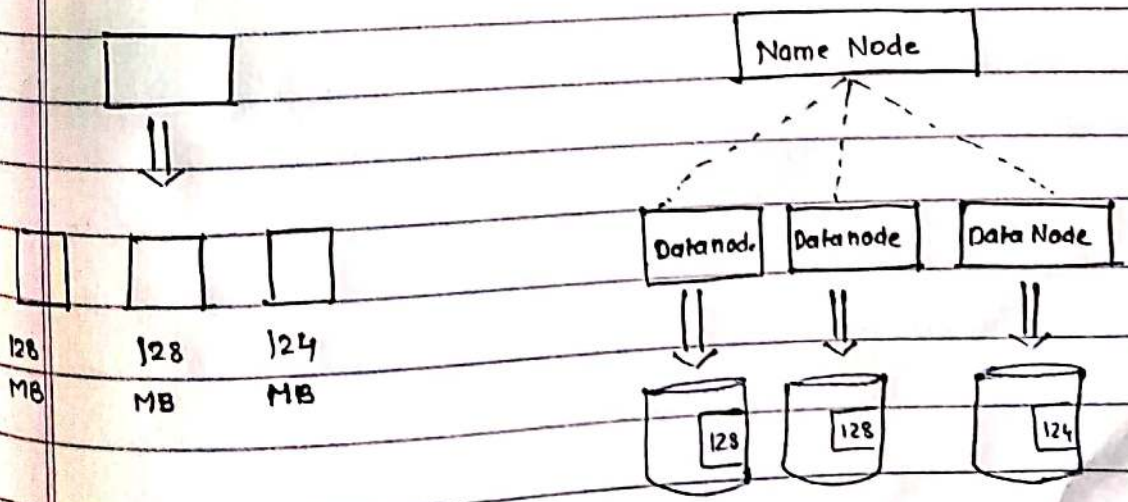
* Secondary Namenode:- (SNN)

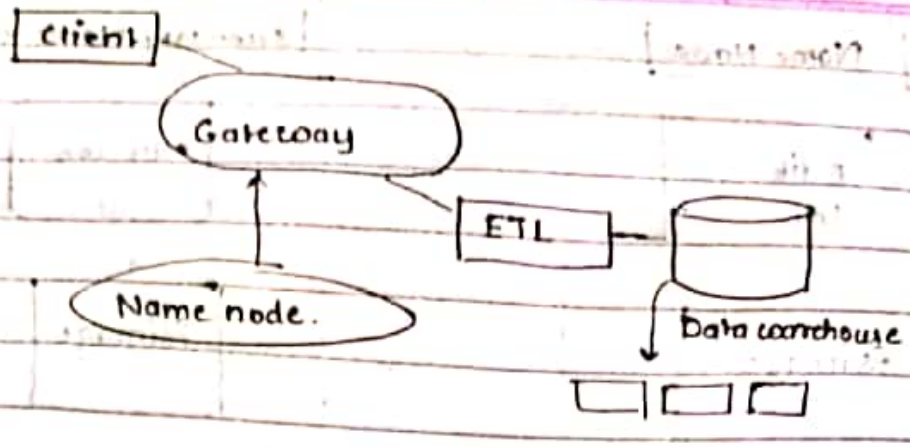
(1) proces of combining fsimage & Edit logs (checkpoints)

(2) S.NN takes over the responsibility of checkpointing therefore making Name node more available.

(3) Checkpointing happens periodically (by default 1hr)

(4) If Name node fails, from secondary name node we will recover the data.





Q] How the data is actually stored in data nodes.

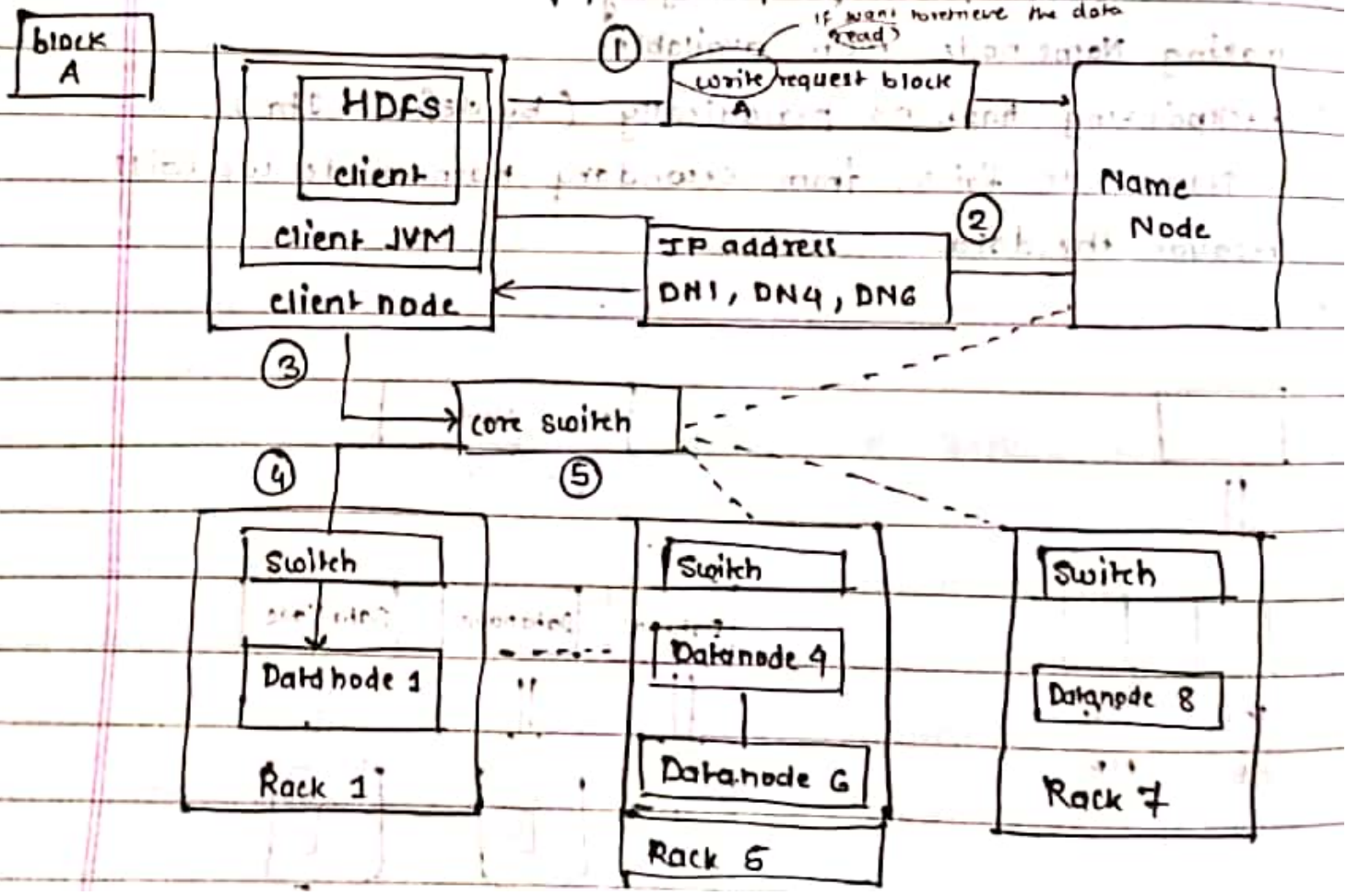
* FAULT TOLERENCE

** How Hadoop cope up with data node failure?

- Each data blocks are replicated (3 thrice by default) and distributed accross different data nodes.

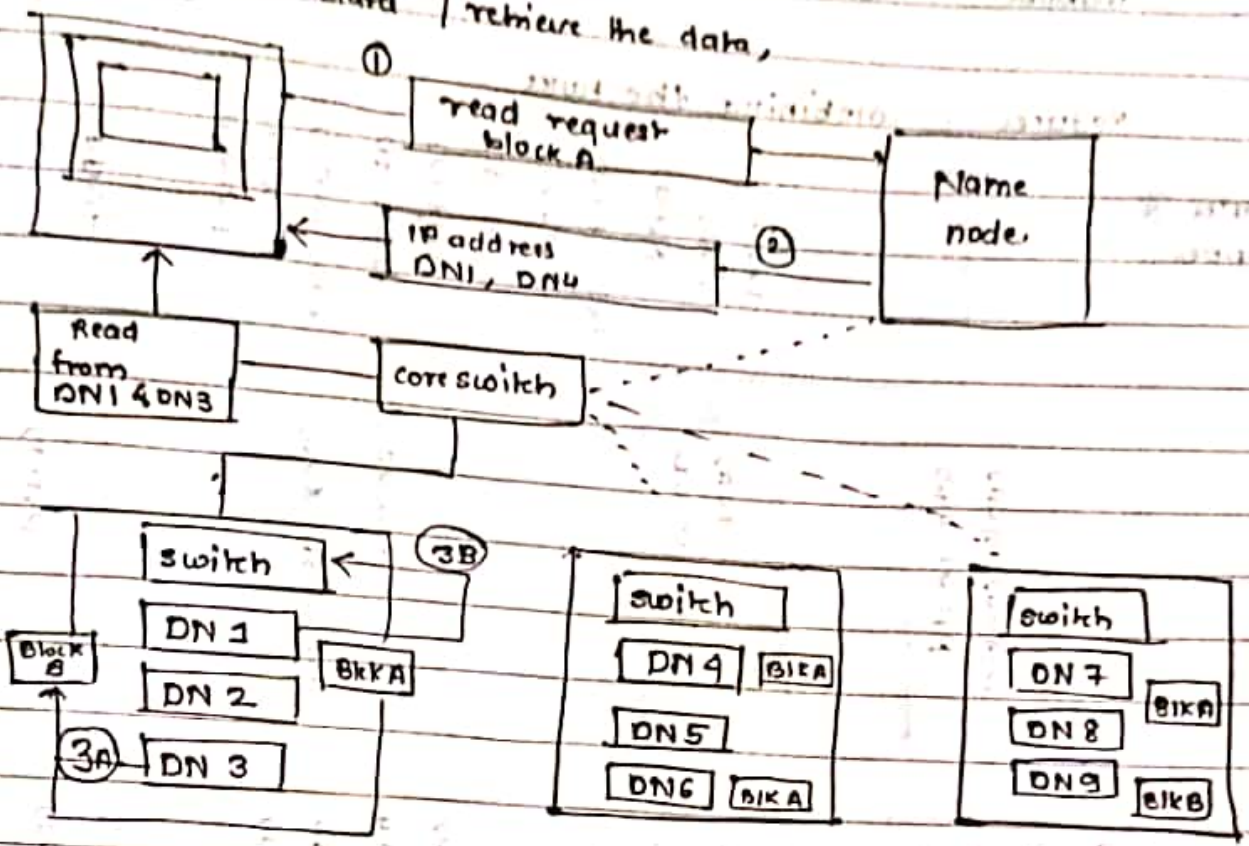
** Hadoop

write a mechanism pipeline setup.



Data nodes are placed randomly.

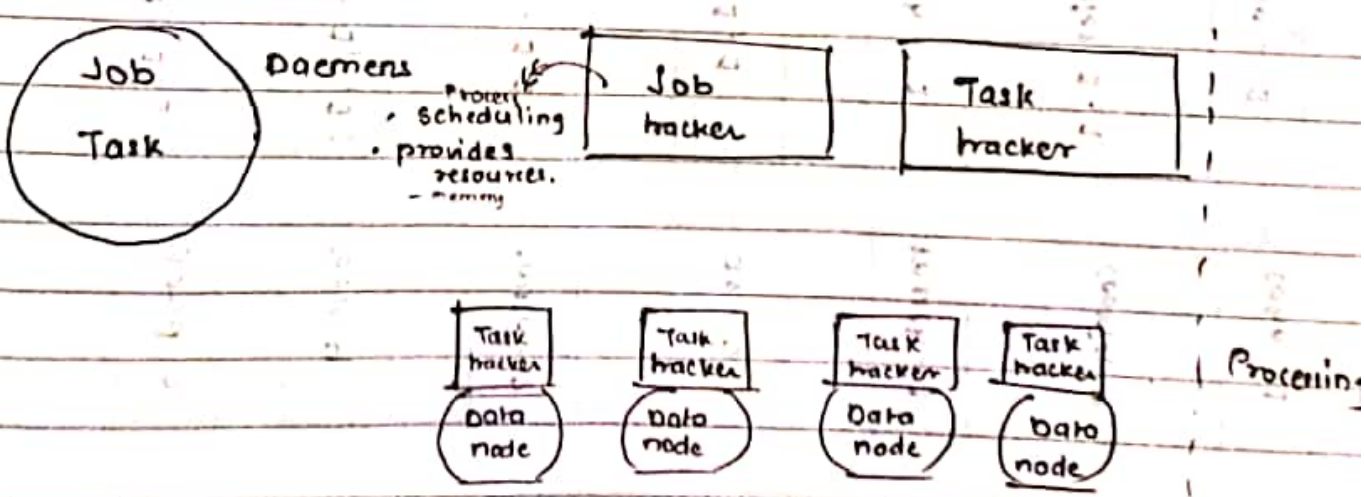
for reading the data / retrieve the data,



* MAP Reduce

- It stores data in distributed manner. the result is gained parallely manner.

Class → Mapper class & Reducer class.



To check network, there will be delay.

$$\therefore PD = \frac{\text{Distance}}{\text{Time}}$$

mapper & Reducer

Mapper - Data nodes are divided into task.

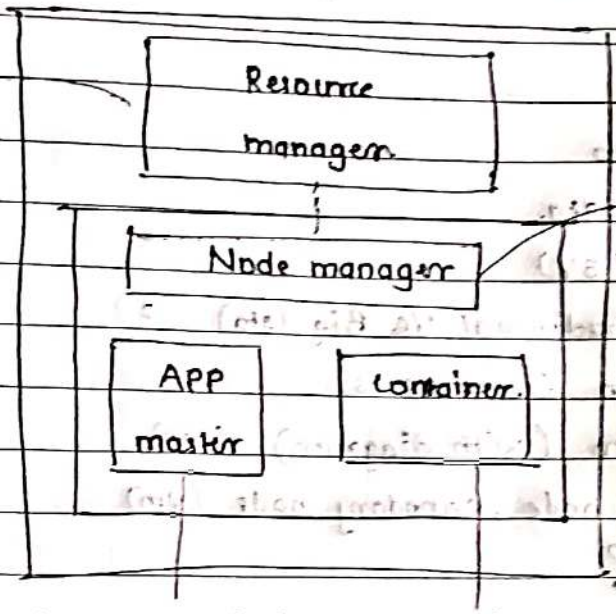
Reducer - combining the tasks

Reducer & mapper.

3) YARN (Yet another resource manager) (hadoop 2.0)

* Hadoop Ecosystem

Apache Hadoop Yarn management (Yet another resource negotiator)



* ECOSYSTEM

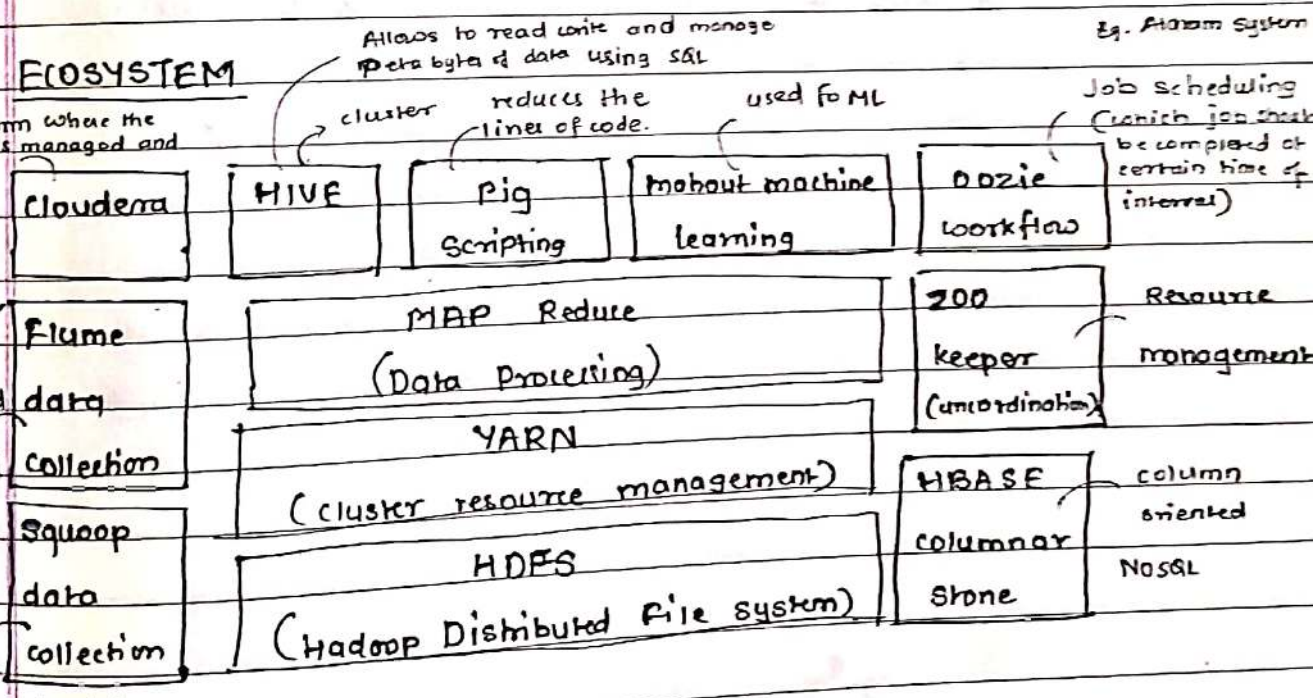
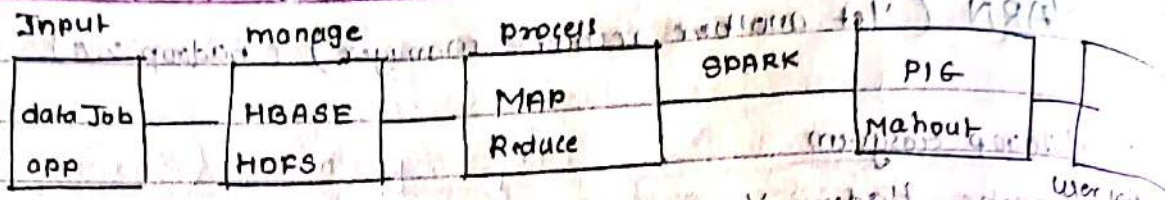


fig. Ecosystem (Hadoop)



Imp Questions:

Q) What is Bigdata
Q) Types of Big data.
Q) Characteristics (5V)
Q) Difference (Traditional Vs Big data) (2)
Q) Explain Hadoop (components)
Q) Hadoop Ecosystem (with diagram) (5m)
Q) data node, name node, secondary node (2m)
Q) Features of BD?

18/7/23 Tuesday. LAB (BDA)

- vi & nano editors.

for vi editor,
vi filename (enter)
write the content and
press esc, :, wq! (enter) [you will come out of the editor]

create: touch filename.txt

instead of hadoop fs -ls we can use
hdfs dfs -ls

when creating a directory,
drwxr is seen where (d) is prefixed.

* Basic of MAP Reducer

mapper

* matrix Vector multiplication by map reducer

$$\underset{A}{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_{2\times 2} \times \underset{B}{\begin{bmatrix} 2 \\ 5 \end{bmatrix}}_{2\times 1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}_{2\times 2}$$

Whenever we are multiplying two matrices there is a rule the total no. of columns in the first matrix should be equal to the total no. of rows from the second matrix. And if they are equal

$$\begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix}$$

Steps for two matrice mul. using MAP reducer.

* ① for matrix 1 $\longrightarrow$ A.

$$((i, k), (A, j, A_{ij}))$$

key, value.

for matrix $2 \rightarrow B$

$((i, k), (B, Bj, Bjk))$

key value.

for matrix A (i, j) (A, Aj, Aij)

$A_{00} = 1$ $((0,0), (A, 0, 1))$

Aij $((0,1), (A, 0, 1))$

$A_{01} = 2$ $((0,0), (A, 1, 2))$

Aij $((0,1), (A, 1, 2))$

$A_{10} = 3$ $((0,0), (A, 0, 3))$

Aij $((1,0), (A, 0, 3))$

$A_{11} = 4$ $((1,0), (A, 1, 4))$

Aij $((1,1), (A, 1, 4))$

for matrix A, key, value (i,j) (Aij, Aij)

$\begin{bmatrix} 2 & 1 \\ 5 & 1 \end{bmatrix}$ for matrix B,

$B_{00} = 2$ $((0,0) (B, 0, 2))$

Bijk $((0,1) (B, 0, 2))$

$B_{01} = 1$ $((0,1) (B, 0, 1))$

Bijk $((0,0) (B, 0, 1))$

$B_{10} = 5$ ((0,0) (B,1,5))
B_{jk} ((1,0) (B,1,5))

$B_{11} = 1$ ((1,0) (B,1,5))
B_{jk} ((1,1) (B,1,1))

*(2) Combiner Task

used to combine key value pairs to reduce network congestion.

(0,0), (A,0,1), (A,1,2), (B,0,2), (B,1,5)
(0,1), (A,0,1), (A,1,2), (B,0,2) (B,1,1)
(1,0), (A,0,3), (A,1,4), (B,0,2), (B,1,5)
(1,1), (A,0,3), (A,1,4), (B,0,1), (B,1,1)

*(3) Reducer Task

(a) for (0,0)

(A,0,1) (A,1,2)
(B,0,2) (B,1,5)
1×2 2×5

2 + 10 = 12

Element selected at foo = 12

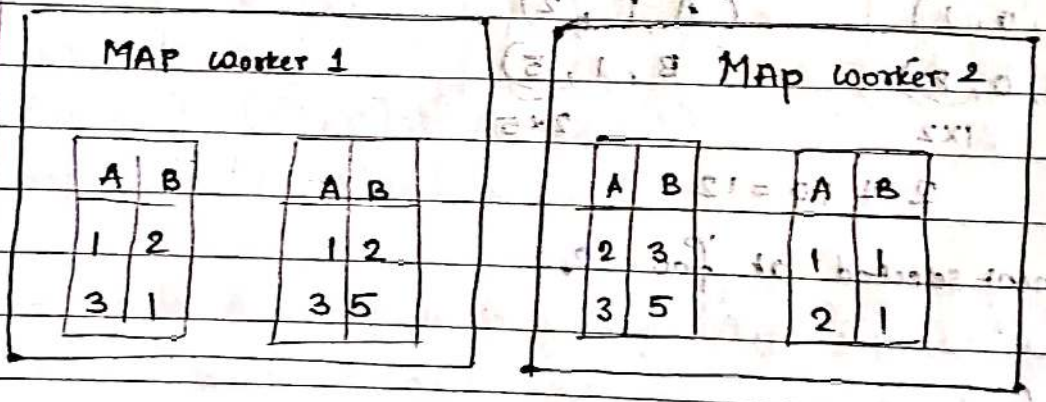
(b) for (

* Relational algebra operations using map reduce.
  - selection
  - projection
  - union
  - intersection
  - difference
  - natural join
  - Grouping & Aggregation.

(1) Selection Algorithm

```
Map (key, value) :
    for tuple in value :
        if tuple satisfies condition :
            emit (tuple, tuples)

reduce (key, values) :
    emit (key, key)
```



Big data divides entire data in various parts each ~~wi~~ part will be process separately & parallely.

★ Selection conditions

eg. B <= 2

MW1

key	value
(1, 2)	(1,2), (1,2)
(3, 1)	(3, 1)
(1, 2)	(1, 2)

add

MW2

key	value
(2, 1)	(2, 1)

Date / /
Page

Map worker checks for duplicacy of the data.

★ A hash function will be applied.

MW1

key	value
(1,2)	(1,2), (1,2)

MW

key	value
(3,1)	(3,1)

MW2

key	value
(1,1)	(1,1)

key	value
(2,1)	(2,1)

★ To reduce task Swapping is done.

- Swapping will be done to discard redundancy that can be caused by duplicate.
- It will swap the adjacent tables with the adjacent workers.

RW1

key	value
(1,2)	(1,2), (1,2)

K	V
(1,1)	(1,1)

RW2 reduced.

K	V
(3,1)	(3,1)

K	V
(2,1)	(2,1)

★ To join two tables

RW1

Key	value
(1,2)	(1,2) (1,2)
(1,1)	(1,1)

RW2

Key	value
(3,1)	(3,1)
(2,1)	(2,1)

★

RW1

A	B
1	2
1	1

RW2

A	B
3	1
2	1

Map worker checks for duplicacy of the data.

★ A hash function will be applied.

MW1

Key	value.
(1,2)	(1,2),(1,2)

MW2

Key	value
(1,1)	(1,1)

MW

Key	value.
(3,1)	(3,1)

Key	value
(2,1)	(2,1)

★ To reduce task Swapping is done.

- Swapping will be done to discard redundancy that can be caused by duplicate.
- It will swap the adjacent tables with the adjacent workers.

RW1

key	value.	K	V
(1,2)	(1,2),(1,2)	(1,1)	(1,1)

RW2 reduced.

K	V	K	V
(3,1)	(3,1)	(2,1)	(2,1)

★ To join two tables

RW1

key	value.
(1,2)	(1,2) (1,2)
(1,1)	(1,1)

RW2

Key	value
(3,1)	(3,1)
(2,1)	(2,1)

★

RW1

A	B
1	2
1	1

RW2

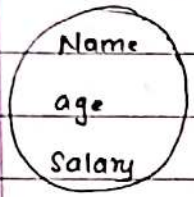
A	B
3	1
2	1

* Projection Algorithm

```
Map(Key, Value)
    for tuple in value:
        ts = tuple with only attributes in S
        emit(ts, ts)
Reduce(key, value)
    emit(key, key)
```

project B : C

Eg



Map coorker 1

A	B	C
1	2	3
3	1	4

A	B	C
2	2	3
4	3	4

Map coorker 2

A	B	C
6	7	2
7	1	1

A	B	C
3	2	5
4	1	0

Step 1:

MW1

key	value
(2,3)	(2,3) (2,3)
(1,4)	(1,4)
(3,4)	(3,4)

MW2

key	value
(7,2)	(7,2)
(1,1)	(1,1)
(2,5)	(2,5)
(1,0)	(1,0)

Step 2: Hash function will be applied, to divide entire table in two parts.

for Map coorker 1

MW1

K	V
(2,3)	(2,3) (2,3)
(1,4)	(1,4)

key	value
(3,4)	(3,4)

for map worker 2.

MW2

key	value
(7,2)	(7,2)
(1,1)	(1,1)

T1

key	value
(2,5)	(2,5)
(1,0)	(1,0)

T2

Step3: Swapping is done to remove the redundancy, i.e present in both map workers.

(Reducer worker 1)

RW1

key	value	key	value
(2,3)	(2,3) (2,3)	(7,2)	(7,2)
(1,4)	(1,4)	(1,1)	(1,1)

(Reducer worker 2)

RW2

key	value	key	value
(3,4)	(3,4)	(2,5)	(2,5)
		(1,0)	(1,0)

Step4: Merge after combining.

RW1

key	value
(2,3)	(2,3) (2,3)
(1,4)	(1,4)
(7,2)	(7,2)
(1,1)	(1,1)

RW2

key	value
(3,4)	(3,4)
(2,5)	(2,5)
(1,0)	(1,0)

Step5

RW1

B	C
2	3
1	4
7	2
1	1

RW2

B	C
3	4
2	5
1	0

Date / Page

* Union Algorithm

```
Map (key, value):
    for tuple in value:
        emit (tuple, tuple)
reduce (key, value)
    emit (key, key)
```

Map worker 1

A	B
1	2
3	1

A	B
1	2
2	1

Map worker 2

A	B
2	3
4	5

A	B
1	1
2	1

Step 1:

MW1

key	value
(1,2)	(1,2) (1,2)
(3,1)	(3,1)
(2,1)	(2,1)

MW2

key	value
(2,3)	(2,3)
(4,5)	(4,5)
(1,1)	(1,1)
(2,1)	(2,1)

Step 2:

MW1

K	V	K	V
(1,2)	(1,2) (1,2)	(2,1)	(2,1)
(3,1)	(3,1)		

MW2

K	V	K	V
(2,3)	(2,3)	(1,1)	(1,1)
(4,5)	(4,5)	(2,1)	(2,1)

Step 3:

RW1

K	V	K	V
(1,2)	(1,2) (1,2)	(2,3)	(2,3)
(3,1)	(3,1)	(4,5)	(4,5)

RW2

K	V	K	V
(2,1)	(2,1)	(1,1)	(1,1)
		(2,1)	(2,1)

Step 4: Merger & combine.

RW1

Key	value
(1,2)	(1,2) (1,2)
(3,1)	(3,1)
(2,3)	(2,3)
(4,5)	(4,5)

RW2

key	value
(2,1)	(2,1) (2,1)
(1,1)	(1,1)
~~(2,1)~~	~~(2,1)~~

Step 5: Union

RW1

A	B
2	2
3	1
2	3
4	5

RW2

A	B
2	1
1	1

* Intersection Algorithm

```
Map (key, value):
    for tuple in value:
        emit (tuple, tuple)
reduce (key, values)
    if values == [key, key]
```

Step 1:

MW1

K	V
(1,2)	(1,2) (1,2)
(3,1)	(3,1)
(2,1)	(2,1)

MW2

K	V
(2,3)	(2,3)
(4,5)	(4,5)
(1,1)	(1,1)
(2,1)	(2,1)

Step2:

MW1

K	V
(1,2)	(1,2)(1,2)
(3,1)	(3,1)

K	V
(2,1)	(2,1)

MW1

K	V
(2,3)	(2,3)
(4,5)	(4,5)

K	V
(1,1)	(1,1)
(2,1)	(2,1)

Step3:

RW1

K	V
(1,2)	(1,2)(1,2)
(3,1)	(3,1)

K	V
(2,3)	(2,3)
(4,5)	(4,5)

RW2

K	V
(2,1)	(2,1)

K	V
(1,1)	(1,1)
(2,1)	(2,1)

Step4:

RW1

K	V
(1,2)	(1,2)(1,2)
(3,1)	(3,1)
(2,3)	(2,3)
(4,5)	(4,5)

RW2

K	V
(2,1)	(2,1)(2,1)
(1,1)	(1,1)

Condition:- If length of values is greater than 1 then emit the key pair of it.

A	B
1	2

A	B
2	1

Date / /
Page

* Difference Algorithm

```
map (key, value)
  if (key == 'R')
    for tuple in value:
      emit (tuple, R)
  else
    for tuple in value:
      emit (tuple, S)
reduce (key, value):
  if values == [R]:
    emit (key, key)
```

Stp 1.

MW1

A	B
1	2
3	1

A	B
1	2
2	1

MW2

A	B
2	3
4	5

A	B
1	1

Stp 2

Key	Value
(1,2)	[T1, T2]
(3,1)	(T1)
(2,1)	(T2)

Key	Value
(2,3)	(T1)
(4,5)	(T1)
(1,1)	(T2)
(2,1)	(T2)

Stp 2: Apply hash function.

Key	Value
(1,2)	[T1, T2]
(3,1)	[T1]

K	V
(2,1)	[T2]

K	V
(2,3)	(T1)
(4,5)	(T1)

K	V
(1,1)	(T2)
(2,1)	(T2)

Step 3:

RW1

K	V
(1,2)	[T1,T2]
(3,1)	[T1]
(2,3)	[T1]
(4,5)	[T1]

RW2

K	V
(2,1)	[T2]
(1,1)	[T2]
(2,1)	[T2]

Step 4: Eliminate the keys that are present both in [T1] & [T2] as well as only from T2.

A	B
3	1
2	3
4	5

* Hadoop Limitation

(1) Security problem.

- there is no inbuilt security in hadoop. Instead we can use SPARK technology to provide security. (AES) : Advance Encryption Standard.

(2) Issue with the small files

- Hadoop is typically java based. Each block size is 128 MB

(3) Slow processing speed:

- Hadoop SPARK overcomes this limitation.

(4) Support for batch processing

- The batch wise processing. by using hadoop SPARK, the continuous processing is possible. (flink) is also useful for live streaming of data.

(5) Not easy to use.

- basic hadoop is java oriented and requires hardcode. instead HIVE Spark can be used for implem

Date / /
Page

3. No SQL.

- No SQL stands for 'Not only SQL'.
- Why use No SQL?
  - DBMS, RDBMS can be used for tabular data storage, but, the data now-a-days is in semi structured, unstructured and structured form. Therefore, No SQL can store data 'not only' tabular form but also can store semi-structured & unstructured data.
- Who invented :- No SQL is invented in 1998 by Karl Strauzie.

* No SQL handles various types of data:
  - Semi-structure
  - Unstructure
  - Rapidly changing data.
  - Big Data.

No SQL database types:
- Key-value pair based.
- Graph based
- column based
- Document based.

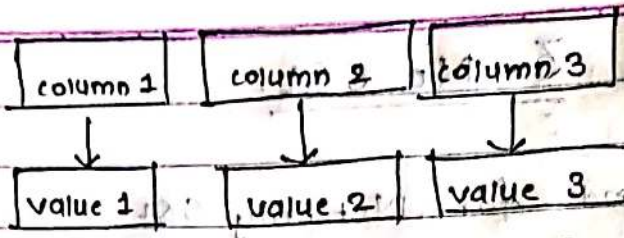
[1] Key-Value pair based :

Eg

Key	value
Name	Sakshi
DOB	30-10-2002

Eg - Redis, Riak are the databases that follow key-value pair.

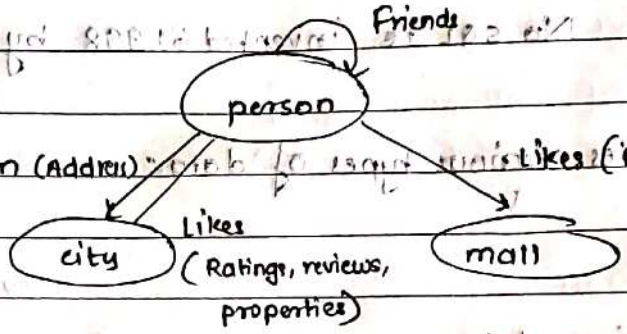
[2] Column based oriented based:
- Eg. HBASE, Cassandra



- HBase and Cassandra follows column oriented based database.

[3] Graph based database:

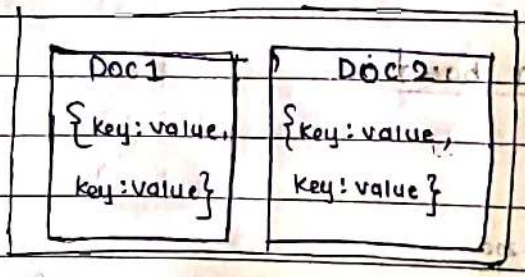
neO4j, OrientDB, Spatial Data (Google maps, ola, uber, etc.) follows graph based database.



- It can be used in multirelational database.
- It considers entity and relations.

[4] Document Oriented based database:

- Stores data in form of XML, json.

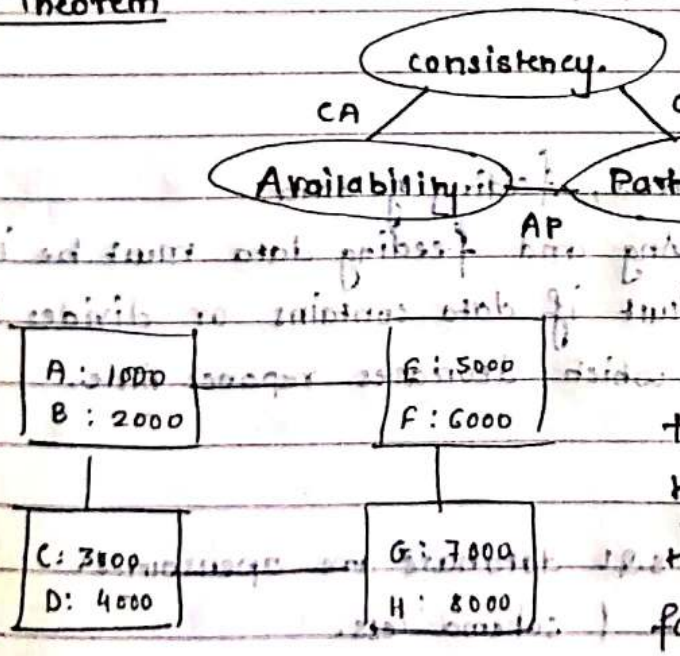


Eg. MongoDB, CouchDB are databases which follows document based database.

[Q] Difference between SQL & NoSQL.

SQL	NoSQL
(1) Relational Database Management System.	(1) Non-Relational / Distributed database system.
(2) Fixed or static or predefined schema.	(2) It has dynamic schema.
(3) Not suited for hierarchical data storage	(3) Best suited for hierarchical storage
(4) Vertically scalable	(4) Horizontally scalable.
(5) It follows acid ACID properties. (Atomicity, consistency, Isolation and Durability).	(6) It follows CA Principle (Consistency, Availability, Partition tolerance)

* CAP Theorem



Eg (Network, where nodes have some data).

for consistency, the nodes connected together have the replicated copies stored therefore if some node is fails, the other nodes will have the replicated copies, hence no work is stopped.

for partitioning, (horizontal arrangement, so to distribute the work)
for availability - The system works even if network fails.

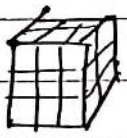
— In No SQL, only two condition is satisfied at a time.
either, CA (Maria DB, SQL server)
or AP (Couch DB, Cassandra)
or CP (Mongo DB, Redis)

* Business ~~Rivals~~ Drivers

parameters:
1) Volume
2) Velocity
3) Variability
4) Agility

[1] volume:
- need to scale up
- Data is extending continuously.

[2] Velocity:
- Slow response time
- RDMS

[3] variability:
- OLAP server where  3D, where variety of data can be stored easily.

[4] Agility:
- Quick and easy access, feeding of data.
- The time or retrieving and feeding data must be instant.
- RDBM find difficult if data contains or divides into many subgroups which decreases responce time.

* No SQL features:
- Most of the No SQL database are opensource.
- It is schema free / schema less.
- They doesn't require object oriented mapping & data normalization.

Date / /
Page

- Most of the No SQL database executed in distributed fashion and they offer auto-scaling.
- No SQL not follow relational model and table with flat fixed column records.
- It offers heterogeneous structure of data in the same domain.
- It is generally used to store big data and real time data.
- It follows CAP theorem (any two at once).

* Features of MongoDB.

(1) Adhoc queries:

- These are types of queries which are not known while structuring the database. Whenever we are designing a database we dont know in future which types of queries may come.

(2) Aggregation:

- We can batch process the data and get single result after doing some sequence of task on group of data.

steps in. input → match → group → sort → output.

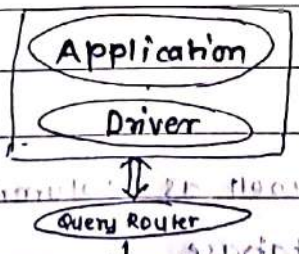
(3) Schemaless / schema free:

(4) Grid File system (GFS):

(5) Document oriented.

(6) Sharding:

- means data partitioning



- high availability of data.
- scalability

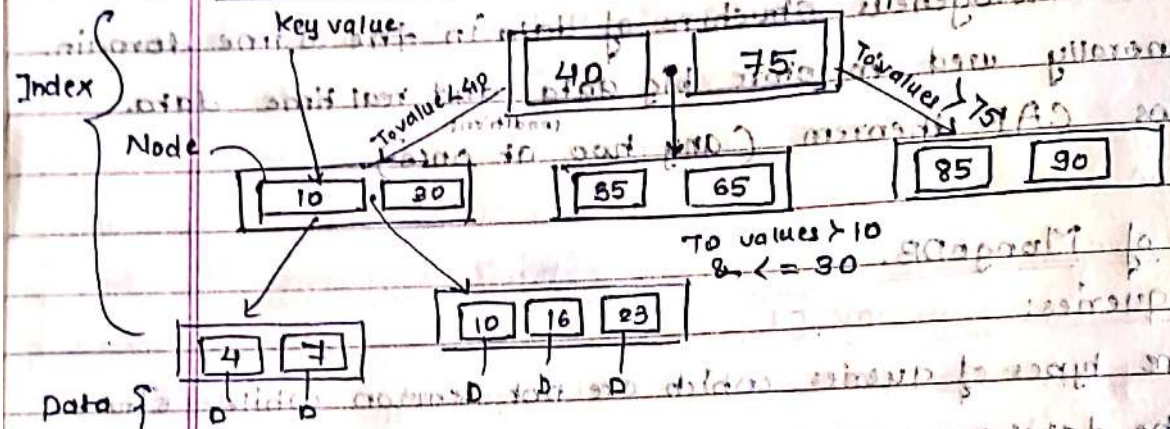
Shard 1 (Primary, Secondary, Secondary) shard 2 ... Shard n

RDBMS → Database → Tables.
MongoDB → DB → Collection → Document.

Date
Page

Sharding separates very large db's into smaller, faster more easily managed called data shards.

(7) Data retrieval



(8) high performance

- MongoDB shows high availability
- It has better query response for indexing and replication

(9) Replication

- This region feature creates copy of document into different machines. We can have primary node and its replicated node where primary node cannot work for some reason then replicated or secondary node takes care of it.

* components of MongoDB.

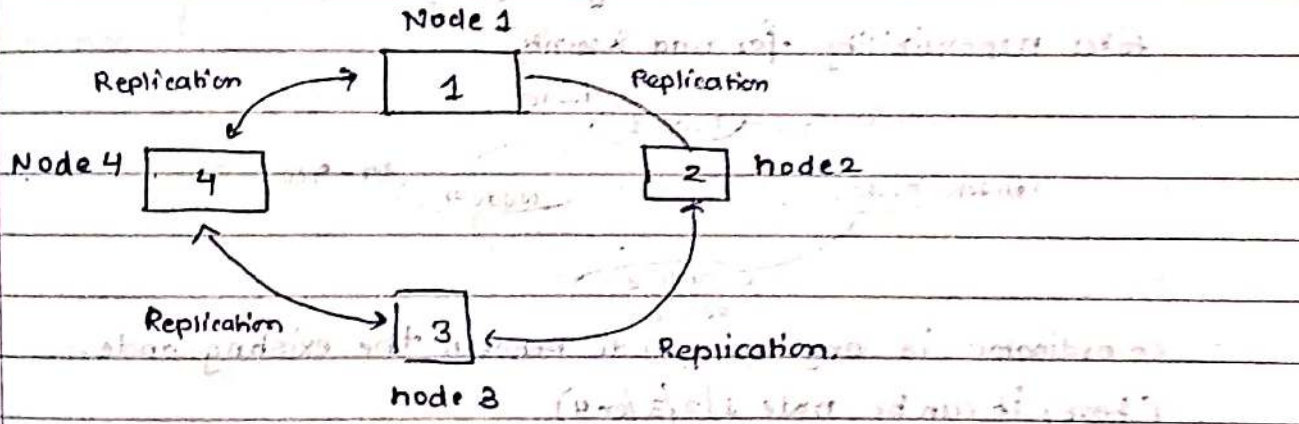
(1) Collection.
(2) Database
(3) document

* Cassandra.

- It is a key value as well as column oriented database.
- It is a open source database
- You can go for 3rd party support.
-

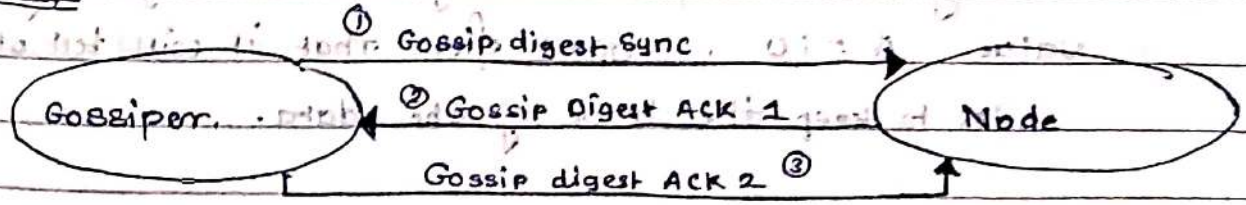
- It is also schemaless / schema free.
- It is scalable & distributed.
- data is distributed accross multiple nodes called as a cluster.
- It is horizontally scalable
- fault tolerant.
- If any node is failure in cluster it will not affect client. (Read & write still occurs even it fail)
- high performance
- It handles multiple petabytes (These are unit of memory) of data.
- It uses best optimized algorithm for read & write.
- Eg. Netflix
- It supports ACID properties.

* Architecture of Cassandra



- Note:- It uses Gossip protocol which allows node to communicate with each other or to find faulty node in structure.
- Data replication in Cassandra.
- It will check for failure detection and used for communication.

① Gossip



To detect failure (Phi accural) is used.

(2) * Snitches (latency)

- It comes under the Cassandra architecture.
- Note: which node provides minimum latency for read & write operation.

(i) Simple Snitching:

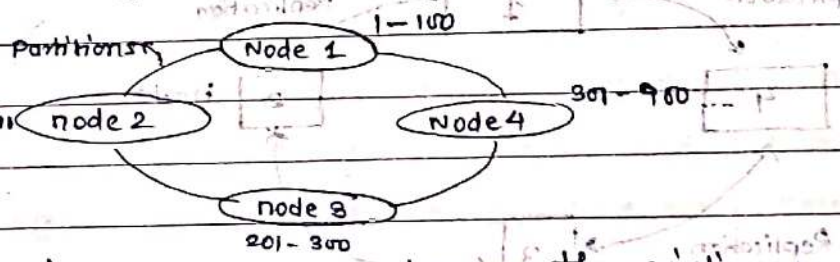
- It identifies the node in a single datacenter which of the rack provide minimum latency for read & write operation

(ii) Dynamic Snitching:

- which of the node will provide least latency for read & write operation, for this Gossip protocol will be used in a multiple cluster.

(3) * Rings and tokens

- co-ordinator node takes care of assigning the communication & takes responsibility for read & write.



- co-ordinator is any one node between the existing node. (here, it can be node 1/2/3/or 4)
- Read & write operations

Write:-

- In this, all the request go to the co-ordinate, multiple node
- It will take hash of the key for eg hash value = 50.
- Co-ordinator node sends the request to the node that is dealing with the partition range (token range) by to save value A = 10, apart from that it will tell other nodes to keep the track of the data.

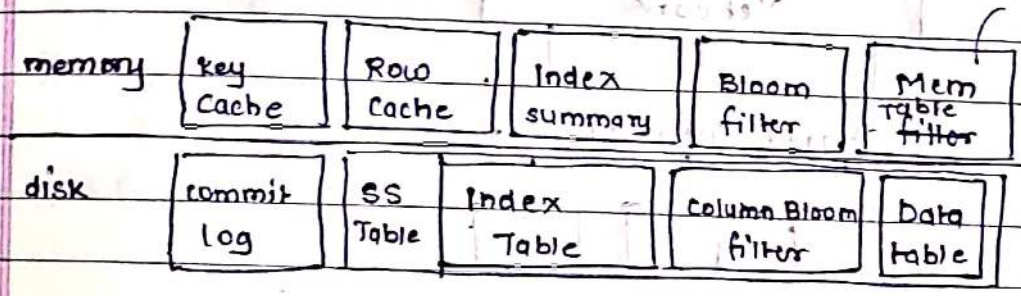
* 8) write the short note on Cassandra Architecture & Cassandra MongoDB.

Read :-

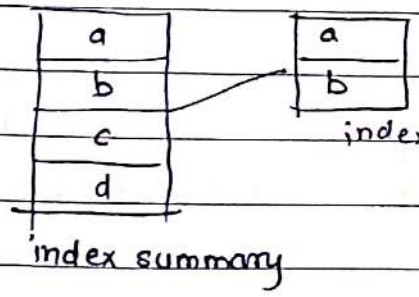
- If two of the nodes responded value of a=10 & one node is responded with value a=5, in this case coordinator will revert to client with value a=10
- The co-ordinator node will tell that node to update the value i.e, a=5 to 10, such type of mechanism is called read repair.

* Components

- Node - data
- Data center - collection of related nodes
- Cluster - one or more data centers
- commit log - sequence of action, memory
- mem table
- SS table
- Bloom filter - (probabilistic structure)
  - value is (0 to 1) range
  - tells if data is present or not present.



- SS ⇒ sorted string tables → (Index table, column Bloom filter, Data table).
- Eg.



- Column bloom filter: It will tell whether the column is present inside key-value pair.

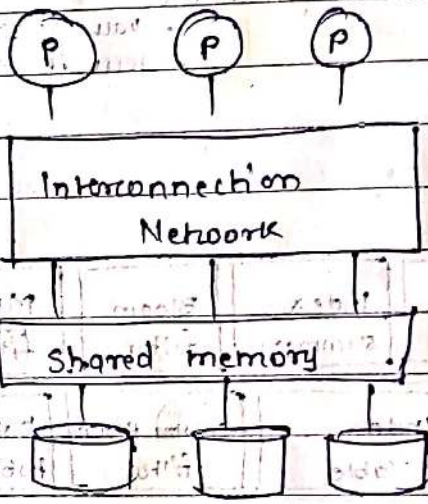
Key cache:- stores primary key of row vs offset (location) in data file

row cache :- holds the list of rows.
- a complete row can be stored in a single node.

* Analyzing Big Data with shared nothing Arch.

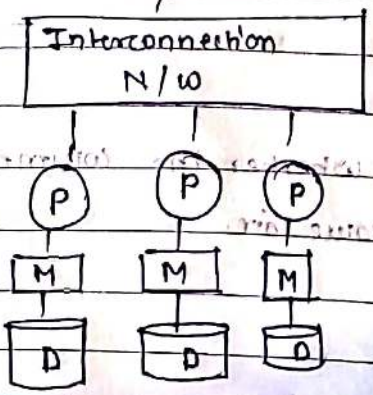
- Shared memory system. (eg. UNIX Filesystem)

IPC - Interprocess memory communication system.

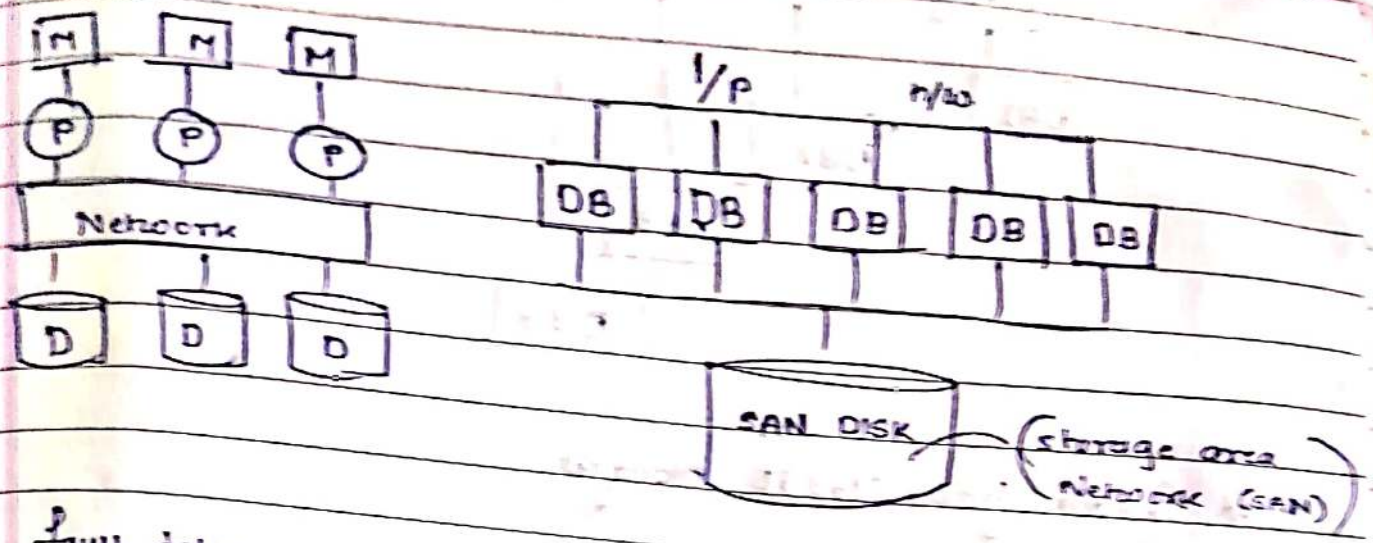


Disadvantage:-
- like bus topology if one line fails, entire system fails
- System down.

- Shared Nothing → HDFS



- Shared Disk ⟶ ORACLE

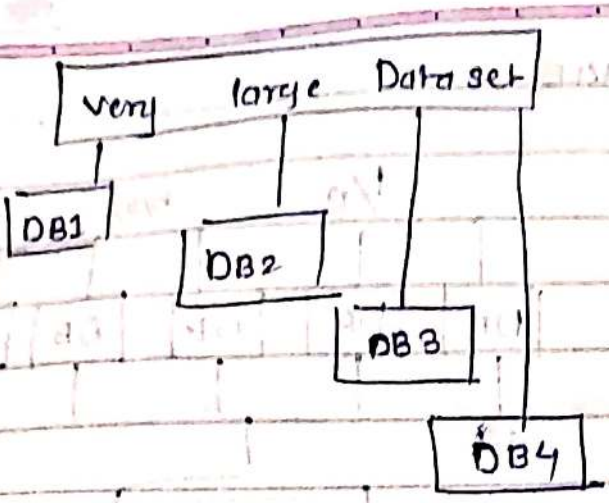


- Fault tolerance :-
If a processor or its memory fails, the other processor can take over its task since the Database is present on disk and are accessible to all processor.

* Hierarchical System
  - Shared memory
  - shared nothing
  - Shared DISK

→ Distributed Model.
  - Master-slave → MongoDB
  - Peer-to-Peer. → Cassandra.

→ Sharding
- different Database present in different server hence implementing sharding



→ * Explain how NoSQL manages Big Data ?

1 — No SQL limits

Types of

* CQL → Astra (database)
cassandra Query language.

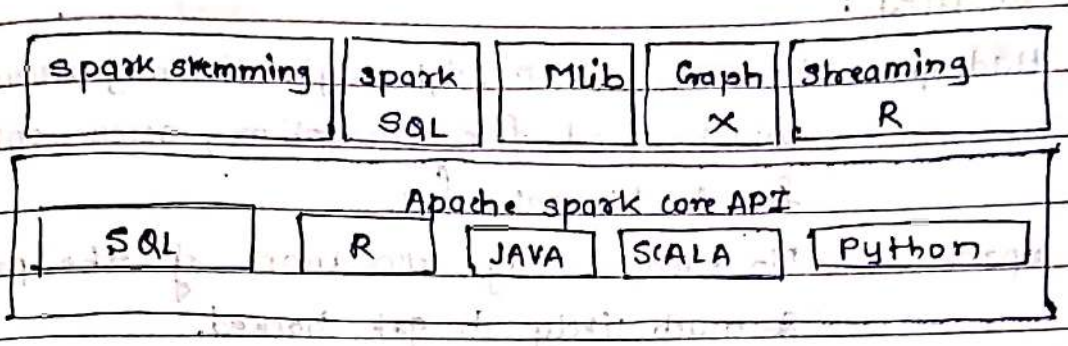
* Dynamo db uses key-value

* Datamodels → mongoDB → BSON (binary) (Json like)
Cassandra (CQL)
Neo4J (cyphr QL)
DynamoDB.

consistency: MDB → cassandra → Neo4j → DynamoDB.

3. Apache Spark

* Basic Spark Architecture.



- 100 times faster in memory (RAM)
- Requires more RAM, hence cost is more.

* Spark Vs Hadoop.

1. Data processing :

- Hadoop : - Batch processing, historical data
  - data which is collected over a period of time and process at later stage.
- spark : - Base on string processing
  - Real time information deliver.

2. Performance :

- Hadoop : slow speed of process (processing speed)
- spark : - Much faster procesing speed
  - It stores & processes everything in memory.
  - Suitable for Machine learning, IoT sensors, Security analysis.

3. Ease of Use :

- Hadoop - It provides addon to support user but doesn't have interactive mode.

- Spark : user friendly API's (Sql, R, Java, Scala, Python) which provides interactive modes for developers.

4. Security :
- Hadoop :- Kerbros, LDAP (lightweight directory access protocol) will be used for encryption, access control lists for traditional file.
- Spark : - No much security, uses more of the passwords & much likely to get hacked.

5. Price :
- Hadoop : less costlier.
- Spark : more costlier.

6. Job scheduler :
Hadoop : oozie workflow
Spark : No external Job scheduler is required.

7. Scalability :
Hadoop : Yahoo has 40 K nodes.
Spark : It has around 7000 nodes.

8. Latency :
- Hadoop : High latency computing web framework
Spark : low latency computing framework.

9. Fault tolerance :
Hadoop : Job tracker, task trackers. (Hadoop uses map reduce and task tracker that provides heart beat if heart beat is missed, all the pending operation will be rescheduled to another job tracker)
Spark : RDD (Re
- for building blocks for fault tolerance.
- they operate in a parallel.

10. compatibility

Hadoop : compatible.

spark : Compatible with each other.
(spark & map reducer)

MODULE 4: Mining Data Stream

* Characteristics of data stream

(1) Hyge volume of infinite continuous data.
(2) Fast changing require real time & fast response.
(3) single scan → entire data is streaming continuously.
(4) System stores only summary (knowledge from the data received till now).
eg. IOT sensors, financial market, amazon website.

* DBMS Vs DSMS

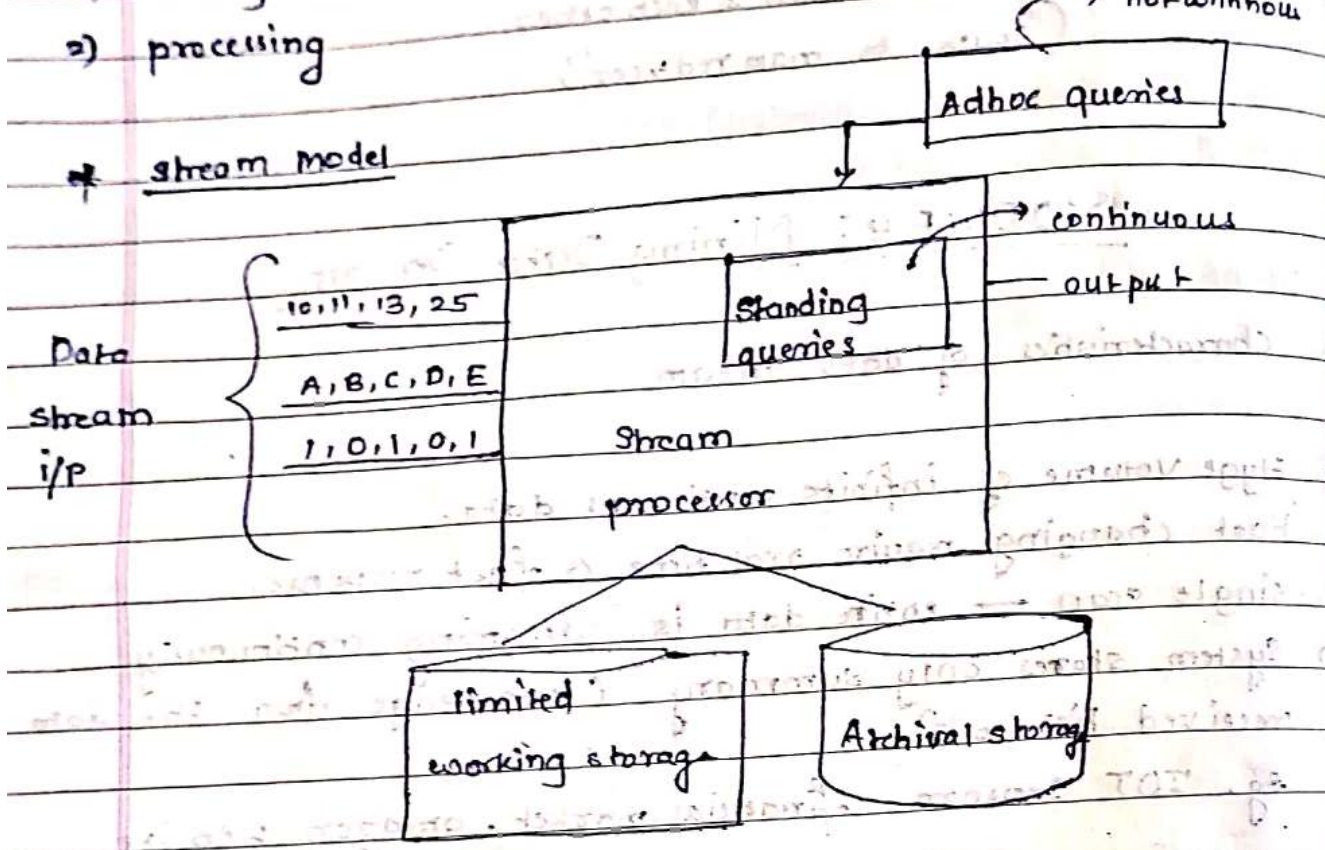
	DBMS	DSMS
(1)	Deals with persistent & static data	Stream data.
(2)	Random Access of Data.	sequential access of data
(3)	Query provides exact answers	Approximate answers.
(4)	One time queries	continuous.
(5)	No real time requirement	real time.

* Challenges

1) Storage

2) processing

* Stream model



→ Standing Q: Queries are asked to the stream (continuou
Eg. Alert me whenever current value exceeds 50 mA

→ Adhoc Queries: Queries can be asked only one time in stream / It is not continuous.
Eg. what is the average of current values captures so far.

op *

* Bloom filter. (filtering data stream)

False Positive : True negative.

1) It is used to check (weather an element belongs to set (weather element is present or not).

(2) - two categories : false

=) • probably says that element belongs to set (False Positive) it won't give assurity that element is 100% present

• Accurately says that the element does not belong to set (only true Negative).

* Two steps in bloom filter :-

(1) Insertion.

(2) Presence of element.

Eg Insert element 10 and 7

Given:- $h_1(x) = x \bmod 5$

$h_2(x) = (2x + 6) \bmod 5$

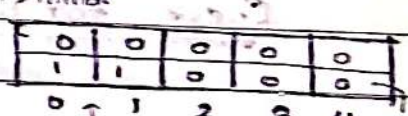
(array size) $m = 5$

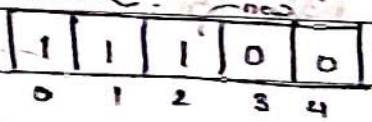
comment on presence of element 14 & 15

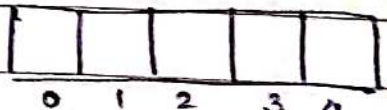
element to insert	$h_1(x)$	$h_2(x)$
(1) 10	0	01
(2) 7	02	0
* To find	for 14 & 15	
(3) 14	4	4

Bloom filter

initial







Accurate says doesn't exists (T. Neg)

At index 4 the value is 0 which means element is not present in

(4) 15 0 1

- condition:
if any of indices is zero, then in that case element is not present.

- At both the indices one (i
- is present [for 0 & 1 positio
- which depicts that element is present in the set but in reality it is not.
- It only provides the probabilit of presence (false positive).

[Q] $h_1(x) = x \bmod 5$ $m = 5$
$h_2(x) = (2x+3) \bmod 5$
- insert 9 & 11

	Element to insert	$h_1(x)$	$h_2(x)$	Bloom filter
1)	9	4	1	1 1 0 0 1 (0 1 2 3 4)
2)	11	1	0	1 1 0 0 1 (0 1 2 3 4)
	Check for 15			
(3)	15	0	3	1 1 0 X 1 (0 1 2 3 4) TN
(3a)	16	1	0	1 1 0 0 1 (0 1 2 3 4) FP

* Flajolet Martin Algorithm:

- counting distinct values from the data input string.
- It is one of the approximate algorithm.

[Q] I/P stream Given: 1, 3, 2, 1, 2, 3, 4, 3, 1, 2, 3, 1;
hash function $h(x) = (6x + 1) \bmod 5$

- calculate the function (hash fn) for following string.

$h(1) = 2$
$h(3) = 4$
$h(2) = 3$
$h(1) = 2$
$h(2) = 3$
$h(3) = 4$
$h(4) = 0$

$h(3) = 4$
$h(1) = 2$
$h(2) = 3$
$h(3) = 4$
$h(1) = 2$

Step: Binary representation.

		trailing zeros
$2 = 010$	$\longrightarrow$	1
$4 = 100$	$\Rightarrow$	2
$3 = 011$	$\longrightarrow$	0
$2 = 010$	$\longrightarrow$	1
$3 = 011$	$\longrightarrow$	0
$4 = 100$	$\longrightarrow$	2

$0 = 000$	$\longrightarrow$	0
$4 = 100$	$\longrightarrow$	2
$2 = 010$	$\longrightarrow$	1
$3 = 011$	$\longrightarrow$	0
$4 = 100$	$\longrightarrow$	2
$2 = 010$	$\longrightarrow$	1

Step: Count the count of trailing zeros (right side zeros from binary).
(right side zero only after 1's).

Step: To find out distinct elements.

Trailing zero's $= (r) = 2$ (maximum trailing zero is 2)
distinct value $= (R) = 2^r$
$= 2^2 = 4$
1, 2, 3, 4

Date / /
Page

Stream

- Approximate no. of unique objects in a string or data in one pass.
- If the string contains n elements with m of them unique, this algo runs O(n) time & need O(log(m)) memory. this algo for approximating the no of distinct element in a stream.

Example: Stream :- 4, 2, 5, 9, 1, 6, 3, 7

$h(x) : (3x+7) \bmod 32$

$h(4) = 19$
$h(2) = 13$
$h(5) = 22$
$h(9) = 2$
$h(1) = 10$
$h(6) = 25$
$h(3) = 16$
$h(7) = 28$

8
32, 16, 8, 4, 2, 1
32, 16, 8, 4, 2, 1

Step: Binary.

trailing zero

$h(4) = 19 = 0,1,2,0,1,1 \rightarrow 010011 \rightarrow 0$
$h(2) = 13 \rightarrow 001101 \rightarrow 0$
$h(5) = 22 \rightarrow 010110 \rightarrow 1$
$h(9) = 2 \rightarrow 000010 \rightarrow 1$
$h(1) = 10 \rightarrow 001010 \rightarrow 1$
$h(6) = 25 \rightarrow 011001 \rightarrow 0$
$h(3) = 16 \rightarrow 010000 \rightarrow 4$
$h(7) = 28 \rightarrow 011100 \rightarrow 2$

Step. maximum = 4

$2^r = 2^4 = 16$.

25 Explain function with its type.

BDA:

module 1

1 Give all characteristic of big data.
2 Explain 5V's of big data. (value, velocity...)
3 Types of big data (Short note)
4 Difference between Traditional & Bigdata.
5 Explain various Tools used in big data.
6 Explain component of Hadoop ecosystem.

module 2

* What is hadoop; component of hadoop; feature of hadoop
8 Difference between hadoop 1.x & 2.x.
9 List down hadoop limitations.
10 What is the use of Job & Task Tracker.
[map reduce & HDFS] — Imp's.
11 Explain concept of map reduce using an example (using word count example).
12 what is map reduce? Explain in detail different Phases with word count eg.
13 Short note on HDFS.
14 List down relational algebra Operation in Map Reduce & Explain
* 10 marks 15 Explain matrix vector multiplication by map Reduce.

module 3

16 What is no sql? list down Advantages.
17 Difference between SQL & NOSQL.
18 Explain NOSQL Database types.
19 Explain Following based Architecture:- Share nothing, Share bus
20 Explain Following Distribution model:- Master slave, Peer-to-Peer
21 Short note:- Cassandra, MongoDB.