structured data

- stored in rdms
- easily searchable

semi structured data

- no fixed format
- diff to anaylse
  un structured data
- data have some kind of tags or markers not as rigid
- like json

charactertic of big data platforms

- distributed storage
- fault tolerance
- scalability
- distributed processing
- real time processing
- support for diverse data
- cost effective

drivers for big data

- explosion of data
- emergence of internet
- emrgence of iot
- cheap storage solution
- cloud computing availability
- business need for anaylisis

big data architecture

- data source -> iot devices , social media
- data ingestion -> kafka , flume
- storage layer -> hdfs , nosql
- processing layer -> mapraduce , spark
- anaysis layer -> hive , pig , ML lib

- visualization layer -> power bi , tableau
- management and monitoring -> ambari , zookeeper

5vs of big data

- velocity
- value
- veracity
- volume
- variety

big data tech component
data storage component -> distributed file sys, nosql db , cloud storage
data processing component -> batch , stream processing , query engine
data ingestion -> batch , stream ingesiton

big data importance

- data driven decision making
- business optimization
- cost reduction
- real time insight
- improved customer experiecnce
- competetive advantage

big data application

- heathcare
- e commerce
- banking
- manufacturing
- government
- education
- transportion and logistics
- media and entertainment

unit 2

history of hadoop

- 2005 -> started under paper (google file system + map reduce simplified data processing on large clusters)
- 2006 -> hdfs + mapreduce
- 2008 -> developed to apache incubator
- 2008 - 13 -> hadoop 1.x versions released . hive + pig + Hbase
- 2013 - > hadoop 2.x version realeased . yarn
- 2015 - present -> hdfs , mapreduce, hive,pig , yarn , Hbase , spark

hdfs

- distributed storage
- block based storage
- fault tolerace
- scalable
- replication upto 3
- write once read many model
- high throughput (large scale processing)
- master slave architecture
- data integrity using checksum
- data locality

challenges of hdfs

- not suitable for small files
- write once read many limitation
- single point of failure of namenode
- large meta data files because of small files

hdfs archecture

- namenode
- datanode

component of hadoop

- data storage component -> hdfs
- data processing component -> mapreduce , yarn , spark
- data management component -> hive , pig , HBase , ZooKeeper

common format in hadoop

- text file
- sequence file
- avro
- parquet
- orc

anaylising data with hadoop

- map reduce for hadoop
  - map phase
    - input data to key val pair , and processes each pair to produce output (intermediate data)
    - this map func is applied paralllely , across nodes for effecinecy
  - reduce phase
    - map out put is filltered or aggregated and final output is generated
    - reduce func aggregate result and show it as result
- hive
- pig
- spark
- hbase

hadoop scaling out

- adding more machine to cluster to increase performance

hadoop streaming

- hadoop utility , to connect mapreduce framework to non java programs
- can write own programs using py
- custom map func
- custom reduce func

hadoop pipe

- c++ api to connect hadoop to c++ programming lang
- can write program using c++
- for perforce critical enironment

more about map reduce

- map reduce (programming model + processing framework)

- used for parallel processing
- data is broken down
- than prcessed in different machine
  mapper
  reducer
  input output
  shuffle and sort

unit 3

block abstraction in hdfs

- block level abstraction
- block location
- block replication
- block rebalacing

data replication in hdfs

- replication basics
- replication factor
- fault tolerance
- block placement strategy

how hdfs stores files ?

- file splitting into blocks
- block assignment to data node
- replication
- data integrity

how hdfs read file ?

- request to namenode
- data node communication
- data transfer
- block retrival

how hdfs write file ?

- request to namenode
- data wrting to datanode

- pipeline mechnism for replication
- acknlowlegement

java interface to hdfs

data ingest

- importing data into big data platform like hadoop
  apache flume

fume architecture

- source
- channel (buffer)
- sinks

sqoop

hadoop io
why compression ?

- storage efficiency
- network efficiency
- improved io tranfer

types of compression

- Gzip
- Bzip2
- snappy
- LZO

serialization

hadoop cluster

unit 4

Yarn

- part of hadoop 2.0
- to manage hadoop cluster , to allocate resource , handle schedule ,to execute job in cluster
- replacement of jobtracker of hadoop 1.0

- components of yarn
  - resource manager -> manage overall cluster resource
  - node manager -> maange single node and report tot the resource manager
  - application master -> each app have its own applciation master to manage job

hadoop 2.0 features

- prev one namenode fails everything fails -> now two name node active and standby namenode
- hadoop federation
- mapreduce version 2 -> run map reduce via yarn
- yarn

shedulers in yarn

- fair sheduler
- capacity sheduler

SQL vs NoSQL Database (Comparison Table)

Feature	SQL Database (Relational)	NoSQL Database (Non-Relational)
Data Model	**Structured data** in **tables** (rows + columns)	**Flexible** models: document, key-value, column, graph
Schema	**Fixed schema** (predefined, strict structure)	**Dynamic schema** (can change anytime, no fixed structure)
Examples	MySQL, PostgreSQL, Oracle, MS SQL Server	MongoDB (Document), Redis (Key-Value), Cassandra (Column), Neo4j (Graph)
Query Language	**SQL** (Structured Query Language)	**Varies** — Mongo Query, CQL (Cassandra), custom APIs
Scalability	**Vertical scaling** (scale-up → bigger machine)	**Horizontal scaling** (scale-out → add more servers)
Transactions	**ACID compliant** (Atomicity, Consistency, Isolation, Durability — reliable)	**BASE** (Basically Available, Soft state, Eventual consistency — more flexible)
Best Suited For	**Structured** & **consistent** data (banking, ERP, CRM)	**Unstructured**, **semi-structured**, large, fast-changing data (social media, IoT, Big Data)
Joins Support	**Supports joins** (multiple tables relations)	Generally **does not support joins** (except some graph DBs)

Feature	SQL Database (Relational)	NoSQL Database (Non-Relational)
Data Integrity	**High integrity** (foreign keys, constraints)	Less strict, focuses on **speed & flexibility**
Performance	Best for **complex queries & transactions**	Best for **fast reads/writes**, massive data, distributed systems
Examples of Use-case	Banking system, Inventory, HR system	Chat apps, Content management, Recommendation systems, IoT

types of no sql db

- key value store
- document store
- column family store
- graph db

addvantage and disadvantage of no sql db

Pig

- run over the hadoop
- part of hadoop
- used for data processing
- used for large complex data transformation , data analysis , data processing
- uses its own lang pig latin
- good for batch processing
- used for strucutred and unstructered data
- supports user defined function for complex task
- used to create data pipeline
- two way to exectire pig
  - local mode
  - map reduce mode
- with the help of grunt we can run pig

Feature	Pig	Databases
Data Model	Unstructured and Semi-structured	Structured (tables with rows and columns)
Query Language	Pig Latin	SQL
Processing Type	Batch processing	OLTP (Online Transaction Processing)
Scalability	Horizontally scalable	Vertically scalable
Data Handling	Large volumes of data	Small to medium-sized data

Hive

- run over the hadoop
- used to query large data
- uses it own lang HiveQL
- used to hide complex hadoop complexity and give simple interface to user

PAPER ID-410082

Subject Code: KOE097

Roll No: | | | | | | | | | | | | | |

BTECH
(SEM VIII) THEORY EXAMINATION 2023-24
BIG DATA

TIME: 3 HRS **M.MARKS: 100**

Note: 1. Attempt all Sections. If require any missing data; then choose suitably.

SECTION A

1. Attempt *all* questions in brief. **2 x 10 = 20**

Q no.	Question	Marks	CO
a.	List any five Big Data platforms.	02	1
b.	Discus the importance of Hadoop technology in Big Data Analytics.	02	1
c.	Explain three benefits of MapReduce.	02	2
d.	Define heartbeat in HDFS.	02	2
e.	List any five Big Data platforms.	02	3
f.	Define data replication in Hadoop Distributed File System.	02	3
g.	Name any two data ingestion tool in Hadoop.	02	4
h.	Compare and Contrast No SQL and Relational Databases	02	4
i.	Discuss the advantages of scala over java.	02	5
j.	Differentiate between Pig and Hive	02	5

SECTION B

2. Attempt any *three* of the following: **3 x 10 = 30**

a.	Differentiate between structured, semi-structured and unstructured data with suitable example.	10	1
b.	Illustrate the anatomy of a MapReduce job run.	10	2
c.	Demonstrate the design of HDFS and concept in detail.	10	3
d.	Explain how CRUD operations with example are performed in MongoDB.	10	4
e.	Illustrate how Zookeeper facilitates coordination and synchronization among HBase.	10	5

SECTION C

3. Attempt any *one* part of the following: **1 x 10 = 10**

a.	Discuss the 5 Vs of Big Data and their implications.	10	1
b.	Elaborate various components of Big Data architecture.	10	1

4. Attempt any *one* part of the following: **1 x 10 = 10**

a.	Explain the architecture of Hadoop Distributed File System (HDFS) and its fault tolerance mechanisms.	10	2
b.	Discuss Hadoop streaming and pipes.	10	2

5. Attempt any *one* part of the following: **1 x 10 = 10**

a.	Examine how a client read and write data in HDFS.	10	3
b.	Mention about the Cluster specification? Describe how to Setting up a Hadoop Cluster?	10	3

2.