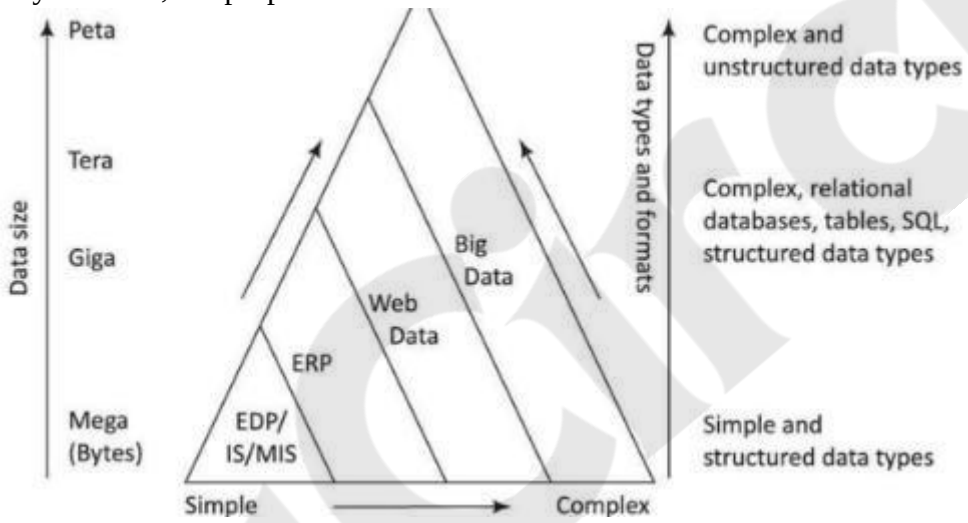
BIG DATA ANALYTICS (21CS71)

MODULE 1

Q.01 a, What is Big Data? Explain evolution of big data & characteristics.

Big data refers to large and complex collection of data that are different to process and analyze using traditional dad processing tools and techniques. Evolution of big data

The rise in technology has led to the production and storage of voluminous amounts data. Earlier megabytes were used but nowadays petabytes are used for processing, analysis, discovering new facts and generating new knowledge. Conventional systems for storage, processing and analysis pose challenges in large growth in volume of data, variety of data, various forms and formats, increasing complexity, faster generation of data and need of quickly processing, analyzing and usage. Figure shows data usage and growth. As size and complexity increase, the proportion of unstructured



Big Data Characteristics

Volume, variety and/or velocity as the key "data management challenges" for enterprises. Analytics also describe the '4Vs', i.e. volume, velocity, variety and veracity

- **Volume** The phrase 'Big Data' contains the term big, which is related to size of the data and hence the characteristic. Size defines the amount or quantity of data, which is generated from an application(s).

- **Velocity** The term velocity refers to the speed of generation of data. Velocity is a measure of how fast the data generates and processes. To meet the demands and the challenges of processing Big Data, the velocity of generation of data plays a crucial role.

- **Variety** Big Data comprises of a variety of data. Data is generated from multiple sources in a system. This introduces variety in data and therefore introduces complexity. Data consists of various forms and formats. The variety is due to the availability of a large number of heterogeneous platforms in the industry. This characteristic helps in effective use of data according to their formats.

PNG.2

- **Veracity** is also considered an important characteristics to take into account the quality of data captured, which can vary greatly, affecting its accurate analysis.

b. Explain the following terms.

i. Scalability & Parallel Processing

ii. Grid & Cluster Computing i Scalability & Parallel Processing

Big Data needs processing of large data volume, and therefore needs intensive computations. Processing complex applications with large datasets (terabyte to petabyte datasets) need hundreds of computing nodes

Analytics Scalability to Big Data

- Vertical scalability means scaling up the given system's resources and increasing the system's analytics, reporting and visualization capabilities. Scaling up means designing the algorithm according to the architecture that uses resources efficiently.

- Horizontal scalability means increasing the number of systems working incoherence and scaling out the workload. Processing different datasets of a large dataset deploys horizontal scalability.Scaling out means using more resources and distributing the processing and storage tasks in parallel.

The easiest way to scale up and scale out the execution of analytics software is to implement it on a bigger machine with more CPUs for greater volume, velocity, variety and complexity of data. The software will definitely perform better on a bigger machine. However, buying faster CPUs, bigger and faster RAM modules and hard disks, faster and bigger motherboards will be expensive compared to the extra performance achieved by efficient design of algorithms.If more CPUs add in a computer, but the software does not exploit the advantage of them, then that will not get any increased performance out of the additional CPUs.

Massively Parallel Processing Platforms

When making software, draw the advantage of multiple computers (or even multiple CPUs within the Scaling uses parallel processing systems. Many programs are so large and/ required to enhance (scale) up the computer system or use massive parallel (MPP) processing (MPPs) platforms.

Parallelization of tasks can be done at several levels:

(i) distributing separate tasks onto separate threads on the same CPU. in distribution
(ii) distributing separate tasks onto separate CPUs on the same computer
(iii) distributing separate tasks onto separate computers

Multiple compute resources are used in parallel processing systems. The computational problem is broken into discrete pieces of sub-tasks that can be processed simultaneously. The system executes multiple program instructions or sub-tasks at any moment in time. Total time taken will be much less than with a single compute resource.

ii. Grid & Cluster Computing

Grid Computing refers to distributed computing, in which a group of computers from several locations are connected with each other to achieve a common task. A group of computers that might spread over remotely comprise a grid. A grid is used for a variety of purposes. A single grid of course, dedicates at an instance to a particular application only. Grid computing provides large-scale resource sharing which is flexible, coordinated and secure among its users. The users consist of individuals, organizations and resources. Grid computing suits data-intensive storage better than storage of small objects of few million of bytes. To achieve the maximum benefit from data grids, they should be used for a large amount of data that can distribute over grid nodes. Besides data grid, the other variation of the grid. Grid computing is scalable. Grid computing also forms a distributed network for resource integration.

Drawbacks of Grid Computing Grid computing is the single point, which leads to failure in case of underperformance or failure of any of the participating nodes. A system's storage capacity varies with the number of users, instances and the amount of data transferred at a given time. Sharing resources among a large number of users helps in reducing infrastructure costs and raising load capacities.

Cluster Computing A cluster is a group of computers connected by a network. The group works together to accomplish the same task. Clusters are used mainly for load balancing. They shift processes between nodes to keep an even load on the group of connected computers. Hadoop architecture uses the similar methods.

Q.02 a. What is Cloud Computing? Explain different services of Cloud.

Cloud computing is a type of Internet-based computing that provides shared processing resources and data to the computers and other devices on demand

One of the best approaches for data processing is to perform parallel and distributed computing in a cloud computing environment.

Cloud resources can be Amazon Web Service (AWS) Elastic Compute Cloud (EC2), Microsoft Azure or Apache CloudStack. Amazon Simple Storage Service (S3) provides simple web services interface to store and retrieve any amount of data, at any time, from anywhere on the web.

Cloud services can be classified into three fundamental types:

1. Infrastructure as a Service (IaaS):

Providing access to resources, such as hard disks, network connections, databases storage, data center and virtual server spaces is Infrastructure as a Service (IaaS). Some examples are Tata CloudStack is an open source software for deploying and managing a large network of virtual machines, and offers public cloud services which provide highly scalable Infrastructure as a Service

2. Platform as a Service (PaaS):

It implies providing the runtime environment to allow developers to build applications and services, which means cloud Platform as a Service. Software at the clouds support and manage the services, storage, networking, deploying, testing, collaborating, hosting and maintaining applications. Examples are Hadoop Cloud Service (IBM BigInsight, Microsoft Azure HD Insights, Oracle Big Data Cloud Services).

3. Software as a Service (SaaS):

Providing software applications as a service to end-users is known as Software as a Service. Software applications are hosted by a service provider and made available to customers over the Internet. Some examples are SQL GoogleSQL, IBM BigSQL, HPE Vertica, Microsoft Polybase and Oracle Big Data SQL.

b. Explain any two Big Data different Applications.

- Big Data Risks:

Large volume and velocity of Big Data provide greater insights but also associate risks with the data used Data included may be erroneous, less accurate or far from reality. Analytics introduces new errors due to such data.Companies need to take risks of using Big Data and design appropriate risk management procedures. They have to implement robust risk management processes and ensure reliable predictions. Corporate, society and individuals must act with responsibility

- Big Data Credit Risk Management:

Financial institutions, such as banks, extend loans to industrial and household sectors.

These institutions in many countries face credit risks, mainly risks of (i) loan defaults, (ii) timely return of interests and principal amount.

Financing institutions are keen to get insights into the following:

Identifying high credit rating business groups and individuals,

Identifying risk involved before lending money

dentifying industrial sectors with greater risks

Identifying types of employees and businesses with greater risks

Anticipating liquidity issues (availability of money for further issue of credit and rescheduling credit over the years

c. How does Berkeley data analytics stack helps in analytics take?

The Berkeley Data Analytics Stack (BDAS) is a powerful tool for big data analytics. It provides a comprehensive solution for data ingestion, processing, and analysis, making it an ideal choice for organizations looking to extract valuable insights from their big data.

BDAS is designed to handle large volumes of data, and it supports various data formats and sources. This includes structured data, such as relational databases, as well as unstructured data, such as text documents and images. BDAS also provides a range of analytics tools and techniques, including machine learning, graph processing, and data mining. One of the key benefits of BDAS is its ability to scale. It can handle large volumes of data and scale to meet the needs of large organizations. This makes it an ideal choice for big data analytics, where large volumes of data need to be processed and analyzed quickly.

BDAS also provides a range of tools and techniques for data management. This includes data ingestion tools, such as Apache Flume and Apache Kafka, as well as data storage tools, such as Apache Hadoop and Apache Cassandra. In addition to its analytics and data management capabilities, BDAS also provides a range of benefits for organizations. It can help to improve decision-making, by providing insights and recommendations based on large volumes of data. It can also help to drive innovation, by providing a platform for experimentation and testing. BDAS is widely used in a range of industries, including finance, healthcare, and retail. It is particularly useful in applications where large volumes of data need to be processed and analyzed quickly, such as in real-time analytics and stream processing. Some of the key use cases for BDAS include real-time analytics, data warehousing, and machine learning. It can also be used for a range of other applications, including predictive maintenance, fraud detection, and customer segmentation.

Overall, BDAS is a powerful tool for big data analytics. Its scalability, flexibility, and range of analytics tools and techniques make it an ideal choice for organizations looking to extract valuable insights from their big data.

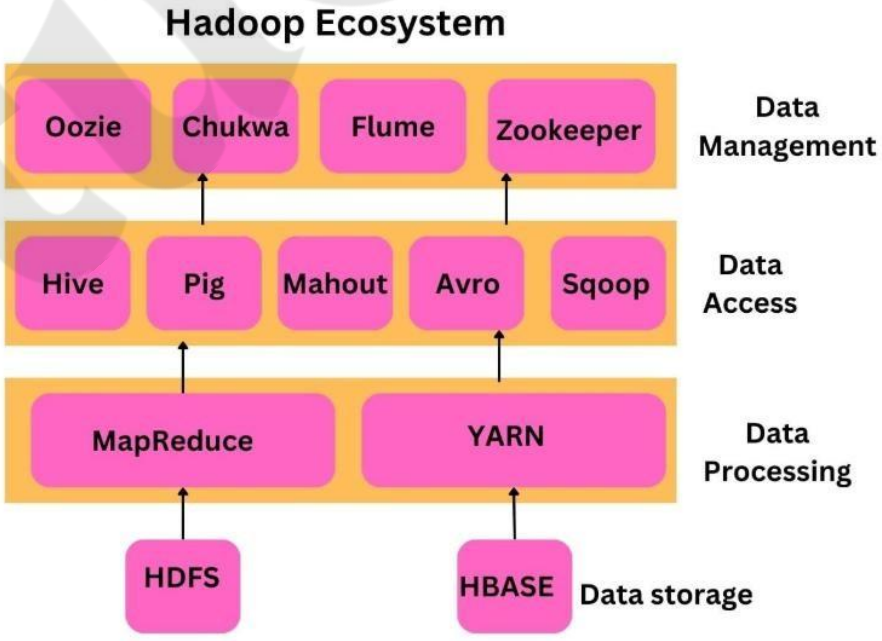
Module-2

Q. 03 a. What is Hadoop? Explain Hadoop eco-system with neat diagram

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models.

Hadoop eco-system

Apache initiated the project for developing storage and processing framework for Big Data storage and processing. Doug Cutting and Machael J. Cafarelle the creators named that framework as Hadoop. Cutting's son was fascinated by a stuffed toy elephant, named Hadoop, and this is how the name Hadoop was derived. The project consisted of two components, one of them is for data store in blocks in the clusters and the other is computations at each individual cluster in parallel with another. Hadoop components are written in Java with part of native code in C. The command line utilities are written in shell scripts. Infrastructure consists of cloud for clusters. A cluster consists of sets of computers or PCs. The Hadoop platform provides a low cost Big Data platform, which is open source and uses cloud services. Tera Bytes of data processing takes just few minutes. Hadoop enables distributed processing of large datasets (above 10 million bytes) across clusters of computers using a programming model called MapReduce. The system characteristics are scalable, selfmanageable, self-healing and distributed file system.

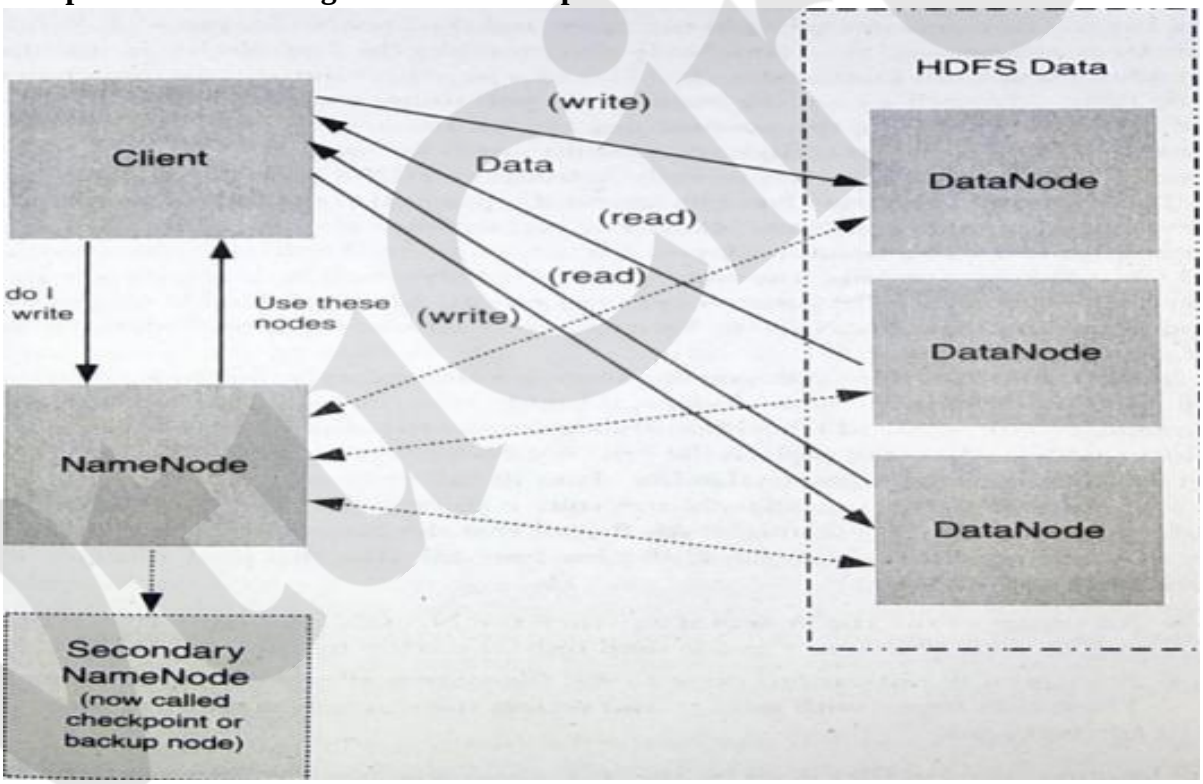


PNG.2

Key Components of the Hadoop Ecosystem

* HBase: A distributed, column-oriented database for real-time, random, and strong consistency data access.
* Hive: A data warehouse infrastructure built on top of Hadoop, allowing users to query data using SQL-like queries.
* Pig: A high-level scripting language for processing large datasets.
* Spark: A fast and general-purpose cluster computing system that can be used for both batch and real-time processing.
* ZooKeeper: A distributed coordination service used for maintaining configuration information, naming, and synchronization.
* Sqoop: A tool for efficiently transferring bulk data between Hadoop and relational databases.
* Flume: A distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.
* Oozie: A workflow scheduler system to manage Hadoop jobs.

b. Explain with neat diagram HDFS Components.



The design of HDFS is based on two types of nodes: NameNode and multiple DataNodes. In a basic design, NameNode manages all the metadata needed to store and retrieve the actual data from the DataNodes. No data is actually stored on the NameNode. The design is a Master/Slave architecture in which master(NameNode) manages the file system namespace and regulates access to files by clients. File system namespace operations such as opening,

PNG.2

closing and renaming files and directories are all managed by the NameNode. The NameNode also determines the mapping of blocks to DataNodes and handles Data Node failures. The slave(DataNodes) are responsible for serving read and write requests from the file system to the clients. The NameNode manages block creation, deletion and replication. When a client writes data, it first communicates with the NameNode and requests to create a file. The NameNode determines how many blocks are needed and provides the client with the DataNodes that will store the data. As part of the storage process, the data blocks are replicated after they are written to the assigned node.

Depending on how many nodes are in the cluster, the NameNode will attempt to write replicas of the data blocks on nodes that are in other separate racks. If there is only one rack, then the replicated blocks are written to other servers in the same rack. After the Data Node acknowledges that the file block replication is complete, the client closes the file and informs the NameNode that the operation is complete. Note that the NameNode does not write any data directly to the DataNodes. It does, however, give the client a limited amount of time to complete the operation. If it does not complete in the time period, the operation is cancelled. The client requests a file from the NameNode, which returns the best DataNodes from which to read the data. The client then access the data directly from the DataNodes. Thus, once the metadata has been delivered to the client, the NameNode steps back and lets the conversation between the client and the DataNodes proceed. While data transfer is progressing, the NameNode also monitors the DataNodes by listening for heartbeats sent from DataNodes. The lack of a heartbeat signal indicates a node failure. Hence the NameNode will route around the failed Data Node and begin re-replicating the now-missing blocks. The mappings b/w data blocks and physical DataNodes are not kept in persistent storage on the NameNode. The NameNode stores all metadata in memory.In almost all Hadoop deployments, there is a SecondaryNameNode(Checkpoint Node). It is not an active failover node and cannot replace the primary NameNode in case of it failure.

c. Write short note on Apache hive.

Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, ad hoc queries, and the analysis of large data sets using a SQL-like language called HiveQL. Hive is considered the de facto standard for interactive SQL queries over petabytes of data using Hadoop. Some essential features: Tools to enable easy data extraction, transformation, and loading (ETL) A mechanism to impose structure on a variety of data formats Access to files stored either directly in HDFS or in other data storage systems such as HBase Query execution via MapReduce and Tez (optimized MapReduce) Hive is also installed as part of the Hortonworks HDP Sandbox. To work in Hive with Hadoop, user with access to HDFS can run the Hive queries.

Simply enter the hive command. If Hive start correctly,it get a hive> prompt.

$ hive

(some messages may show up here)

hive>

Hive command to create and drop the table. That Hive commands must end with a semicolon(;).

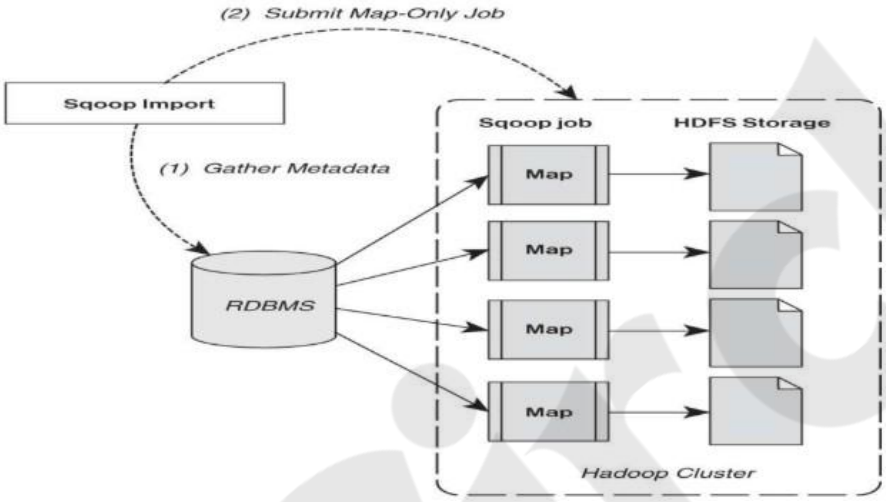
hive> CREATE TABLE pokes (foo INT, bar STRING);

To see the table is created, **hive> SHOW TABLES;**
To drop the table, **hive> DROP TABLE pokes;**

Q.04 a Explain Apache Sqoop Import and Export methods.

Sqoop is a tool designed to transfer data between Hadoop and relational databases. Sqoop is used to -import data from a relational database management system (RDBMS) into the Hadoop Distributed File System(HDFS), transform the data in Hadoop and export the data back into an RDBMS.

Sqoop import method:



The data import is done in two steps :

1) Sqoop examines the database to gather the necessary metadata for the data to be imported.
2) The imported data are saved in an HDFS directory.

* Sqoop will use the database name for the directory, or the user can specify any alternative directory where the files should be populated. By default, these files contain comma delimited fields, with new lines separating different records.

Sqoop Export method :

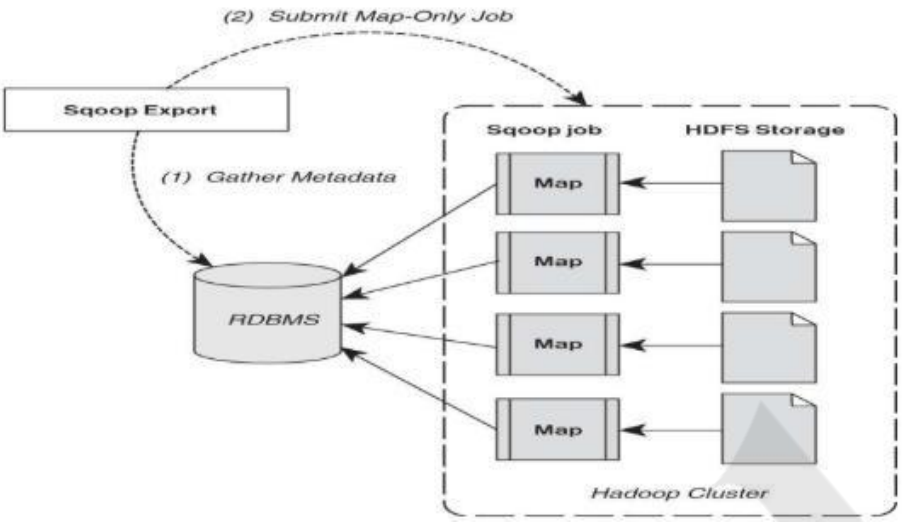
Data export from the cluster works in a similar fashion. The export is done in two steps :

1) examine the database for metadata.

2) Map-only Hadoop job to write the data to the database.

Sqoop divides the input data set into splits, then uses individual map tasks to push the splits to the database.

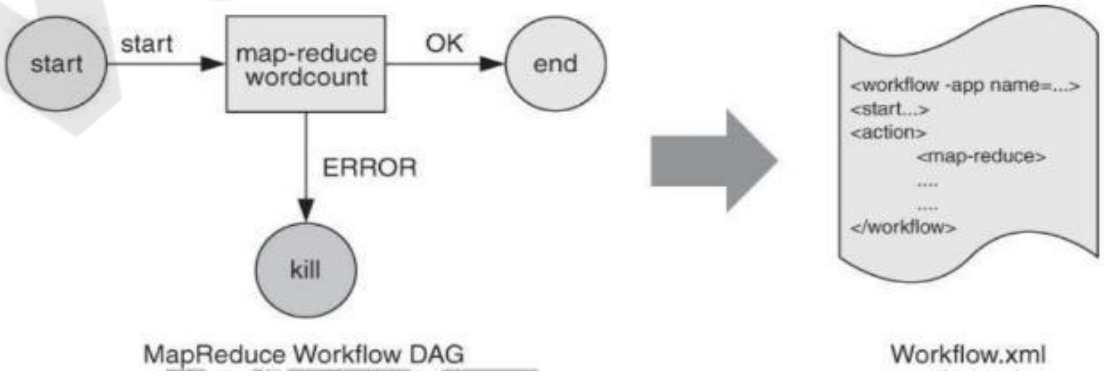
PNG.2



b Explain Apache Oozie with neat diagram.

Oozie is a workflow director system designed to run and manage multiple related Apache Hadoop jobs. For instance, complete data input and analysis may require several discrete Hadoop jobs to be run as a workflow in which the output of one job serves as the input for a successive job. Oozie is designed to construct and manage these workflows. Oozie is not a substitute for the YARN scheduler. That is, YARN manages resources for individual Hadoop jobs, and Oozie provides a way to connect and control Hadoop jobs on the cluster.

Oozie workflow jobs are represented as directed acyclic graphs (DAGs) of actions. (DAGs are basically graphs that cannot have directed loops.) Three types of Oozie jobs are permitted: Workflow—a specified sequence of Hadoop jobs with outcome-based decision points and control dependency. Progress from one action to another cannot happen until the first action is complete. Coordinator—a scheduled workflow job that can run at various time intervals or when data become available. Bundle—a higher-level Oozie abstraction that will batch a set of coordinator jobs. Oozie is integrated with the rest of the Hadoop stack, supporting several types of Hadoop jobs out of the box (e.g., Java MapReduce, Streaming MapReduce, Pig, Hive, and Sqoop) as well as system-specific jobs (e.g., Java programs and shell scripts). Oozie also provides a CLI and a web UI for monitoring jobs. Following figure depicts a simple Oozie workflow. In this case, Oozie runs a basic MapReduce operation. If the application was successful, the job ends; if an error occurred, the job is killed.

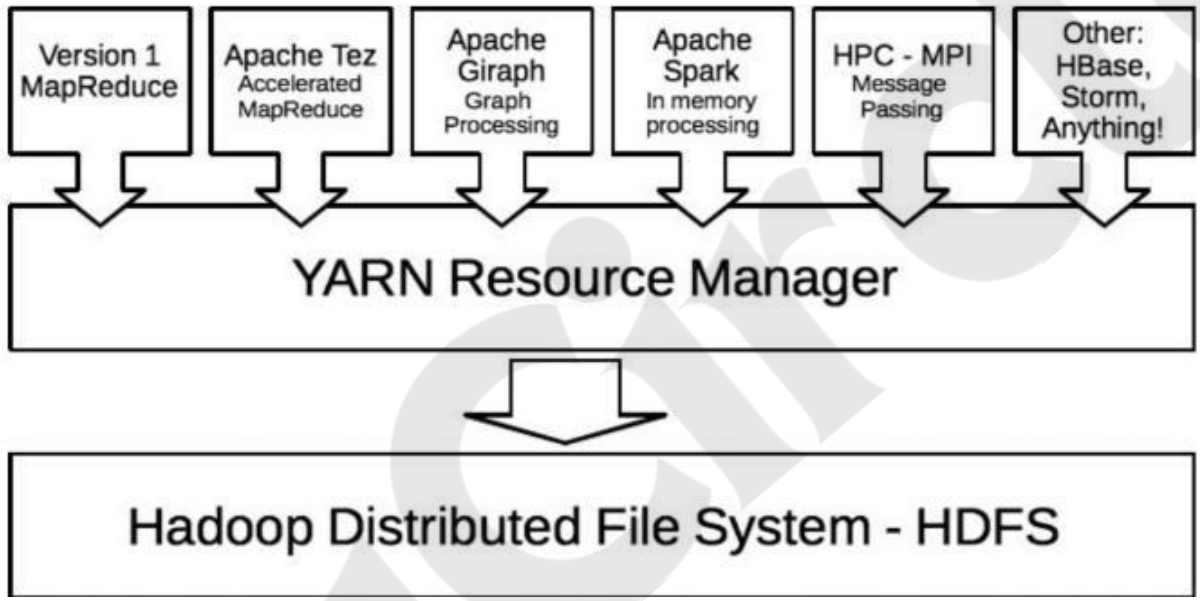


PNG.2

c. Explain YARN application framework.

One of the most exciting aspects of Hadoop version 2 is the capability to run all types of applications on a Hadoop cluster. In Hadoop version 1, the only processing model available to users is MapReduce. In Hadoop version 2, MapReduce is separated from the resource management layer of Hadoop and placed into its own application framework. Indeed, the growing number of YARN applications offers a high level and multifaceted interface to the Hadoop data lake.

YARN presents a resource management platform, which provides services such as scheduling, fault monitoring, data locality, and more to MapReduce and other frameworks. Figure 8.2 illustrates some of the various frameworks that will run under YARN. Note that the Hadoop version 1 applications (e.g., Pig and Hive) run under the MapReduce framework.



This section presents a brief survey of emerging open source YARN application frameworks that are being developed to run under YARN. As of this writing, many YARN frameworks are under active development and the framework landscape is expected to change rapidly. Commercial vendors are also taking advantage of the YARN platform. Consult the webpage for each individual framework for full details of its current stage of development and deployment.

PNG.2

Module-3

Q. 05 a What is NOSQL? Explain CAP Theorem.

A new category of data stores is NoSQL (means Not Only SQL) data stores. NoSQL is an altogether new approach to thinking about databases, such as schema flexibility, simple relationships, dynamic schemas. auto sharding, replication, integrated caching, horizontal scalability of shards, distributable tuples, semi structured data and flexibility in approach. Issues with NoSQL data stores are lack of standardization in approaches, processing difficulties for complex queries CAP Theorem :

Among C,A and P two are at least present for the application service process. **Consistency** means all copies have the same value like in traditional DB, **Availability** means at least one copy is available in case a partition becomes active or fails, **Partition** means parts which are active but may not cooperate (share) as in distributed DBs.

1.***Consistency*** in distributed database means that all nodes observe the same data at the same time. Therefore, the operations in one partition of the database should reflect in other related partitions in case of distributed database Operations, which change the sales data from a specific showroom in a table should also reflect in changes in related tables which are using that sales data.

2. ***Availability*** means that during the transactions, the field values must be available in other partitions of the database so that each request receives a response on success as well as failure. Replication ensures availability.

3 ***Partition*** means division of a large database into different databases without affecting the operations on them by adopting specified procedures

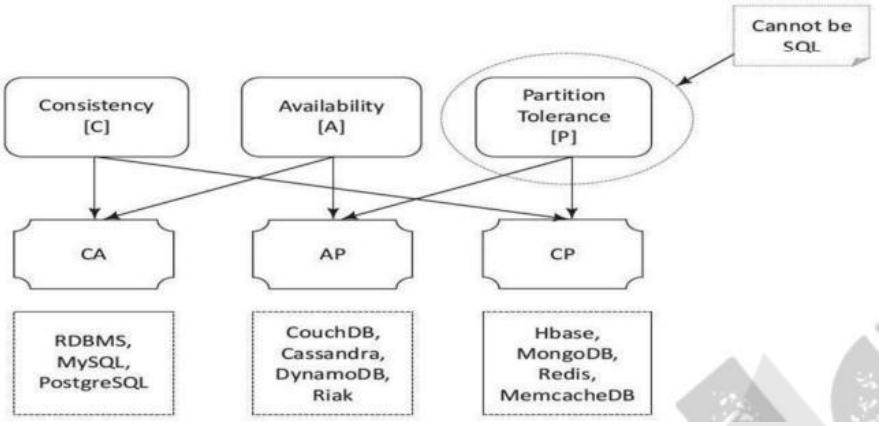
Partition tolerance: Refers to continuation of operations as a whole even in case of message loss, node failure or node not reachable

Brewer's CAP (Consistency. Availability and Partition Tolerance) theorem demonstrates that any distributed system cannot guarantee C. A and P together.

1. Consistency-All nodes observe the same data at the same time.
2. Availability-Each request receives a response on success failure.
3. Partition Tolerance-The system continues to operate as a whole even in case of message

loss, node failure or node not reachable.In case of any network failure, a choice can be:

- Database must answer, and that answer would be old or wrong data (AP)
- Database should not answer, unless it receives the latest copy of the data (CP).



The CAP theorem implies that for a network partition system, the choice of consistency and availability are mutually exclusive. CA means consistency and availability, AP means availability and partition tolerance and CP means consistency and partition tolerance.

b Explain NOSQL Data Architecture Patterns.

Data Architecture Pattern is a logical way of categorizing data that will be stored on the Database. NoSQL is a type of database which helps to perform operations on big data and store it in a valid format. It is widely used because of its flexibility and a wide variety of services

The data is stored in NoSQL in any of the following four data architecture patterns.

1. Key-Value Store Database

2. Column Store Database

3. Document Database

4. Graph Database

1.Key-Value Store Database: This model is one of the most basic models of NoSQL databases. As the name suggests, the data is stored in form of Key-Value Pairs. The key is usually a sequence of strings, integers or characters but can also be a more advanced data type. The value is typically linked or co-related to the key. The key-value pair storage databases generally store data as a hash table where each key is unique. The value can be of any type (JSON, BLOB(Binary Large Object), strings, etc). This type of pattern is usually used in shopping websites or e-commerce applications.

Advantages:

- Can handle large amounts of data and heavy load,
- Easy retrieval of data by keys.

Examples:

- DynamoDB

PNG.2

- Berkeley DB

2. Column Store Database: Rather than storing data in relational tuples, the data is stored in individual cells which are further grouped into columns. Column-oriented databases work only on columns. They store large amounts of data into columns together. Format and titles of the columns can diverge from one row to other. Every column is treated separately. But still, each individual column may contain multiple other columns like traditional databases.
Basically, columns are mode of storage in this type.

Advantages:

- Data is readily available
- Queries like SUM, AVERAGE, COUNT can be easily performed on columns.

Examples:

- HBase
- Bigtable by Google
- Cassandra

3. Document Database: The document database fetches and accumulates data in form of key-value pairs but here, the values are called as Documents. Document can be stated as a complex data structure. Document here can be a form of text, arrays, strings, JSON, XML or any such format. The use of nested documents is also very common. It is very effective as most of the data created is usually in form of JSONs and is unstructured.

Advantages:

- This type of format is very useful and apt for semi-structured data.
- Storage retrieval and managing of documents is easy.

Examples:

- MongoDB
- CouchDB

4. Graph Databases: Clearly, this architecture pattern deals with the storage and management of data in graphs. Graphs are basically structures that depict connections between two or more objects in some data. The objects or entities are called as nodes and are joined together by relationships called Edges. Each edge has a unique identifier. Each node serves as a point of contact for the graph.

Advantages:

- Fastest traversal because of connections.
- Spatial data can be easily handled.

Examples:

- Neo4J
- FlockDB(Used by Twitter)

Q. 06 a Explain Shared Nothing Architecture for Big Data tasks.

1) Single Server Model

Simplest distribution option for NoSQL data store and access is Single Server Distribution (SSD) of an application. A graph database processes the relationships between nodes at a server. The SSD model suits well for graph DBs. An application executes the data sequentially on a single server.

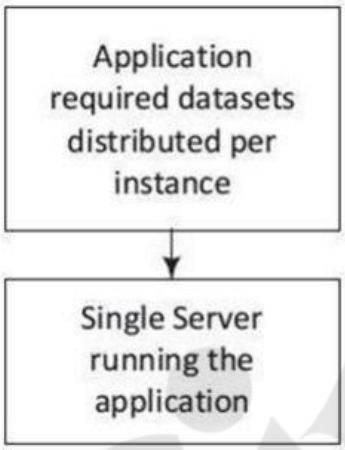


Figure: single server model

2) Sharding Very Large Databases:

- The application programming model in SN architecture is such that an application process runs on multiple shards in parallel.
- Sharding provides horizontal scalability. A data store may add an autosharding feature.
- The performance improves in the SN. However, in case of a link failure with the application, the application can migrate the shard DB to another node.

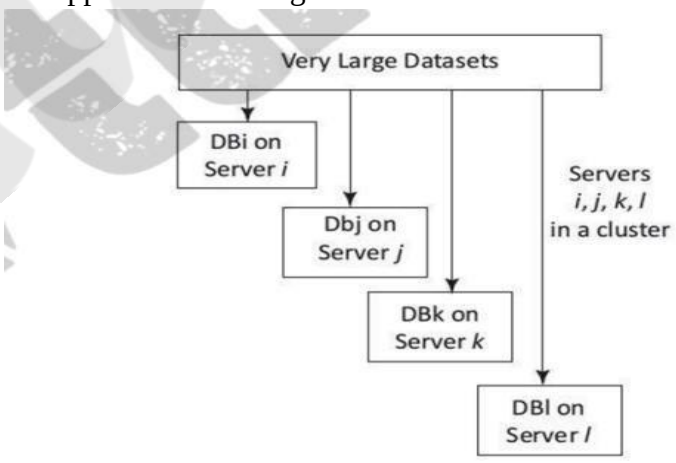
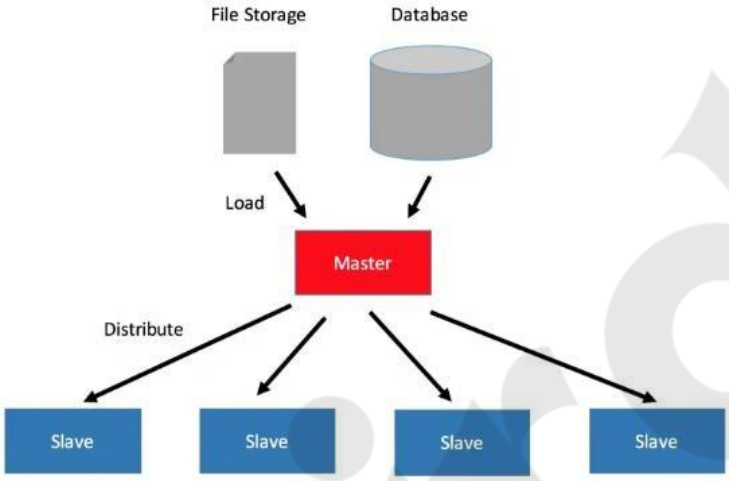


Figure: Shards distributed on four servers in a cluster

PNG.2

3) Master-Slave Distribution Model

A node serves as a master or primary node and the other nodes are slave nodes. Master directs the slaves. Slave nodes data replicate on multiple slave servers in Master Slave Ditribution (MSD) model. When a process updates the master, it updates the slaves also. A process uses the slaves for read operations. Processing performance improves when process runs large datasets distributed onto the slave nodes.



4) Peer-to-Peer Distribution Model

Peer-to-Peer distribution (PPD) model and replication show the following characteristics:

(1) All replication nodes accept read request and send the responses.

(2) All replicas function equally.

(3) Node failures do not cause loss of write capability

Cassandra adoptsthe PPD model.The data distributes among all the nodesin a cluster. Performance can further be enhanced by adding the nodes. Since nodes read and write both, a replicated node also has updated data. Therefore, the biggest advantage in the model is consistency.

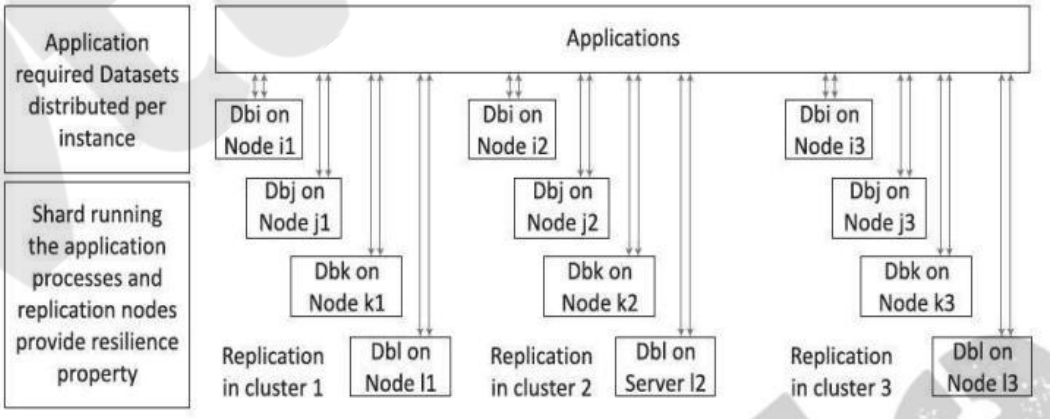


Figure: Shards replicating on the nodes, which does read and write operations both

PNG.2

b Explain MONGO DATABASE.

MongoDB was developed by a NewYork based organization named 10gen which is now known as MongoDB Inc. It was initially developed as a PAAS (Platform as a Service). Later in 2009, it is introduced in the market as an open source database server that was maintained and supported by MongoDB Inc.

- MongoDB is an open source DBMS. MongoDB programs create and manage databases.
- MongoDB manages the collection and document data store.
- MongoDB functions do querying and accessing the required information.
- The functions include viewing, querying, changing, visualizing and running the transactions. Changing includes updating, inserting, appending or deleting.

Characteristics of MongoDB are:

(i) non-relational, (ii) NoSQL, (iii) distributed, (iv) open source, (v) document based, crossplatform, (vii) Scalable, (viii) flexible data model, (ix) Indexed, (x) multi-master (xi) fault tolerant.

Features of MongoDB:

1. MongoDB data store is a physical container for collections. Each DB gets its own set of files on the file system. A number of DBs can run on a single MongoDB server. DB is default DB in MongoDB that stores within a data folder.The database server of MongoDB ismongod and the client ismonqo.
2. Collection stores a number of MongoDB documents. It is analogous to a table of RDBMS. A collection exists within a single DB to achieve a single purpose. Collections may store documents that do not have the same fields. Thus, documents of the collection are schemaless.
3. Document model is well defined. Structure of document is clear; Document is the unit of storing data in a MongoDB database. Documents are analogous to the records of RDBMS table. Insert, update and delete operations can be performed on a collection. Document use JASON(JavaScriptObject Notation) approach for storing data.
4. MongoDB is a document data store in which one collection holds different documents. Data store in the form of JSON-style documents.
5. Storing of data is flexible, and data store consists of JSON-like documents. This implies that the fields can vary from document to document and data structure can be changed over time.
6. Storing of documents on disk is in BSON serialization format. BSON is a binary representation of JSON documents.
7. Querying, indexing, and real time aggregation allows accessing and analyzing the data efficiently.

8. Deep query-ability—Supports dynamic queries on documents using a document-based query language that‟s nearly as powerful as SQL.
9. No complex Joins.
10. Distributed DB makes availability high, and provides horizontal scalability.

MongoDB uses

An organization might want to use MongoDB for the following:

- **Storage.** MongoDB can store large structured and unstructured data volumes and is scalable vertically and horizontally. Indexes are used to improve search performance. Searches are also done by field, range and expression queries.
- **Data integration**. This integrates data for applications, including for hybrid and multi-cloud applications.
- **Complex data structures descriptions**. Document databases enable the embedding of documents to describe nested structures (a structure within a structure) and can tolerate variations in data.
- **Load balancing.** MongoDB can be used to run over multiple servers.

Module-4

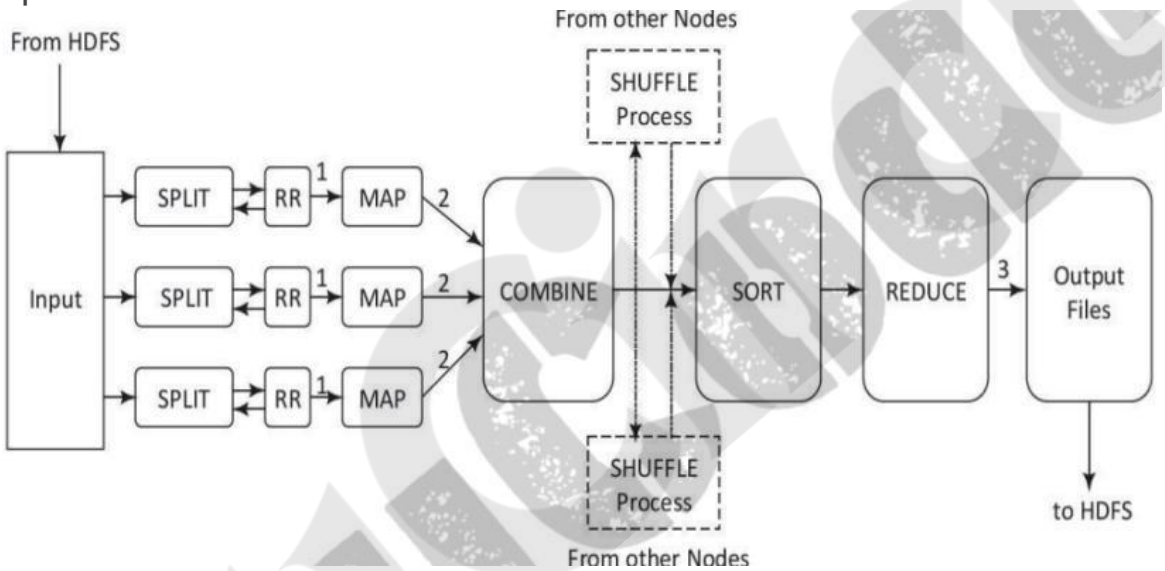
Q. 07 a Explain Map Reduce Execution steps with neat diagram.

MapReduce is the data processing layer. It processes the huge amount of structured and unstructured data stored in HDFS. MapReduce processes data in parallel by dividing the job into the set of independent tasks. So, parallel processing improves speed and reliability.

Hadoop MapReduce data processing takes place in 2 phases- Map and Reduce phase.

- **Map phase-** It is the first phase of data processing. In this phase, we specify all the complex logic/business rules/costly code.
- **Reduce phase-** It is the second phase of processing. In this phase, we specify lightweight processing like aggregation/summation.

MapReduce Execution flow



Steps of MapReduce Job Execution flow

MapReduce processess the data in various phases with the help of different components. Let's discuss the steps of job execution in Hadoop.

1. Input Files

In input files data for MapReduce job is stored. In **HDFS**, input files reside. Input files format is arbitrary. Line-based log files and binary format can also be used.

2. InputFormat

After that InputFormat defines how to split and read these input files. It selects the files or other objects for input. InputFormat creates InputSplit.

3. InputSplits

PNG.2

It represents the data which will be processed by an individual **Mapper**. For each split, one map task is created. Thus the number of map tasks is equal to the number of InputSplits. Framework divide split into records, which mapper process.

4. RecordReader

It communicates with the inputSplit. And then converts the data into **key-value pairs** suitable for reading by the Mapper. RecordReader by default uses TextInputFormat to convert data into a key-value pair.

5. Mapper

It processes input record produced by the RecordReader and generates intermediate keyvalue pairs. The intermediate output is completely different from the input pair. The output of the mapper is the full collection of key-value pairs.

4. Combiner

Combiner is Mini-reducer which performs local aggregation on the mapper's output. It minimizes the data transfer between mapper and reducer. So, when the combiner functionality completes, framework passes the output to the partitioner for further processing.

5. Partitioner

Partitioner comes into the existence if we are working with more than one reducer. It takes the output of the combiner and performs partitioning.

Partitioning of output takes place on the basis of the key in MapReduce. By hash function, key (or a subset of the key) derives the partition.

6. Shuffling and Sorting

After partitioning, the output is shuffled to the reduce node. The shuffling is the physical movement of the data which is done over the network. As all the mappers finish and shuffle the output on the reducer nodes.

7. Reducer

Reducer then takes set of intermediate key-value pairs produced by the mappers as the input. After that runs a reducer function on each of them to generate the output.

8. RecordWriter

It writes these output key-value pair from the Reducer phase to the output files.

9. OutputFormat

OutputFormat defines the way how RecordReader writes these output key-value pairs in output files. So, its instances provided by the Hadoop write files in HDFS. Thus OutputFormat instances write the final output of reducer on HDFS.

b What is HIVE? Explain HIVE Architecture.

- Hive was created by Facebook.
- Hive is a data warehousing tool and is also a data store on the top of Hadoop.
- An enterprise uses a data warehouse as large data repositories that are designed to enable the tracking, managing, and analyzing the data.
- Hive processes structured data and integrates data from multiple heterogeneous sources. Additionally, also manages the constantly growing volumes of data.

Components of Hive architecture are:

- **Hive Server (Thrift**) - An optional service that allows a remote client to submit requests to Hive and retrieve results. Requests can use a variety of programming languages. Thrift server exposes a very simple client API to execute HiveQL statement.
- **Hive CLI (Command Line Interface) -** Popular interface to interact with Hive. Hive runs in local mode that uses local storage when running the CLI on a Hadoop cluster instead of HDFS.
- **WebInterface** - Hive can be accessed using a web browser as well. This requires a HWI Server running on some designated code. The URL http:// hadoop: / hwi command can be used to access Hive through the web.
- **Metastore** - It is the system catalog. All other components of Hive interact with the Metastore. It stores the schema or metadata of tables, databases and columns in a table, their data types and HDFS mapping.
- **Hive Driver** - It manages the life cycle of a HiveQL statement during compilation, optimization and execution.

Q. 08 a Explain Pig architecture for scripts dataflow and processing

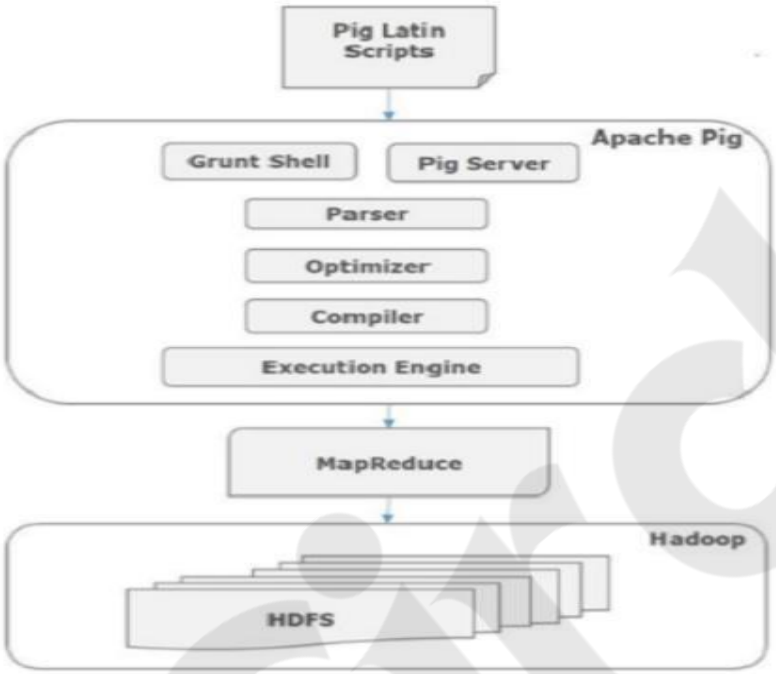
Apache developed Pig, which:

- Is an abstraction over MapReduce
- Is an execution framework for parallel processing
- Reduces the complexities of writing a MapReduce program
- Is a high-level dataflow language. Dataflow language means that a Pig operation node takes the inputs and generates the output for the next node.
- Is mostly used in HDFS environment
- Performs data manipulation operations at files at data nodes in Hadoop.

Pig Architecture:

- Firstly, Pig Latin scripts submit to the Apache Pig Execution Engine.

- The figure shows Pig architecture for scripts dataflow and processing in the HDFS environment.



Parser:

- A parser handles Pig scripts after passing through Grunt or Pig Server.
- The Parser performs type checking and checks the script syntax.
- The output is a Directed Acyclic Graph (DAG).
- Acyclic means only one set of inputs are simultaneously at a node, and only one set of output generates after node operations.
- DAG represents the Pig Latin statements and logical operators.
- Nodes represent the logical operators.
- Edges between sequentially traversed nodes represent the data flows.

Optimizer:

- The DAG is submitted to the logical optimizer.
- The optimization activities, such as split, merge, transform and reorder operators execute in this phase.
- The optimization is an automatic feature.

PNG.2

- The optimizer reduces the amount of data in the pipeline at any instant of time, while processing the extracted data.

Compiler:

- The compiler compiles after the optimization process.
- The optimized codes are a series of MapReduce jobs.

Execution Engine:

- Finally, the MapReduce jobs submit for execution to the engine.
- The MapReduce jobs execute and it outputs the final result.

b Explain Key Value pairing in Map Reduce.

In MapReduce process, before passing the data to the mapper, data should be first converted into key-value pairs as mapper only understands key-value pairs of data. key-value pairs in Hadoop MapReduce is generated as follows:

- **InputSplit –** It is the logical representation of data. The data to be processed by an individual Mapper is presented by the InputSplit.
- **RecordReader –** It communicates with the InputSplit and it converts the Split into records which are in form of key-value pairs that are suitable for reading by the mapper. By default, RecordReader uses TextInputFormat for converting data into a key-value pair. RecordReader communicates with the InputSplit until the file reading is not completed. Learn more about

In MapReduce, map function processes a certain key-value pair and emits a certain number of key-value pairs and the Reduce function processes values grouped by the same key and emits another set of key-value pairs as output. The output types of the Map should match the input types of the Reduce as shown below:

- **Map:** (K1, V1) -> list (K2, V2)
- **Reduce:** {(K2, list (V2 }) -> list (K3, V3)

key-value pair generated in Hadoop Map Reduce

Generation of a key-value pair in Hadoop depends on the data set and the required output. In general, the key-value pair is specified in 4 places: Map input, Map output, Reduce input and Reduce output. **a. Map Input**

Map-input by default will take the line offset as the key and the content of the line will be the value as Text. By using custom InputFormat we can modify them. **b. Map Output**

Map basic responsibility is to filter the data and provide the environment for grouping of data based on the key.

- **Key** – It will be the field/ text/ object on which the data has to be grouped and aggregated on the **reducer** side.
- **Value** – It will be the field/ text/ object which is to be handled by each individual reduce method.

c. Reduce Input

The output of Map is the input for reduce, so it is same as Map-Output.

d. Reduce Output

It depends on the required output.

Example

Suppose, the content of the file which is stored in HDFS is **John is Mark Joey is John**. Using InputFormat, we will define how this file will split and read. By default, RecordReader uses TextInputFormat to convert this file into a key-value pair.

- **Key –** It is offset of the beginning of the line within the file.
- **Value –** It is the content of the line, excluding line terminators. From the above content of the file-
  - **Key** is 0
  - **Value** is John is Mark Joey is John.

Module-5

Q. 09 a What is Machine Learning? Explain different types of Regression Analysis.

Machine learning is the ability of computer systems to learn to make predictions from observations and data. Machine learning can use the information provided by the study of big data to generate valuable business insights.

Simple Linear Regression

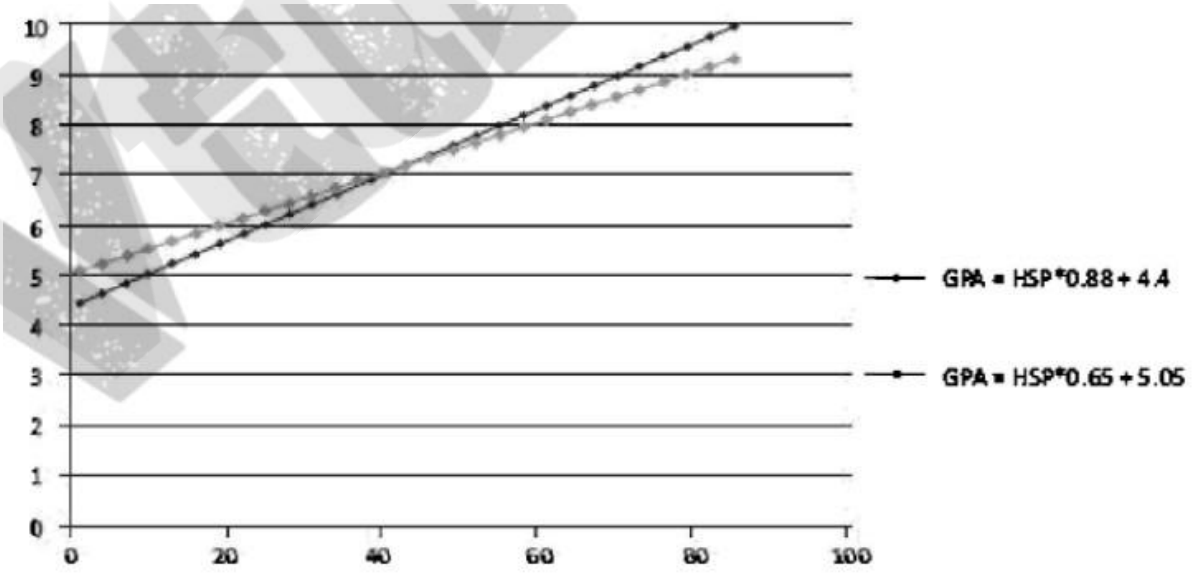
Linear regression is a simple and widely used algorithm. It is a supervised ML algorithmfor predictive analysis. It models a relationship between the independent predictor or explanatory, and the dependent outcome or variable, y using a linearity equation.

$$y = f(a_0, a_1) = a_0 + a_1 x,$$

where n_0 is a constant and n_1 is the linearity coefficient.

Simple linear regression is performed when the requirement is prediction of values of one variable, with given values of another variable. The following example explains the meaning of linear regression.

The purpose of regression analysis is to come up with an equation of a line that fits through a cluster of points with minimal amount of deviation from the line. The best-fitting line is called the regression line. The deviation of the points from the line is called an „error". Once this regression equation is obtained, the GPA of a student in college examinations can be predicted provided his/her high school percentage is given. Simple linear regression is actually the same as a correlation between independent and dependent variables. Figure 6.6 shows a simple linear regression with two regression lines with different regression equations. Looking at the scatter plot, two lines can fit best to summarize the relation between GPA and high school percentage.



PNG.2

Following notations can be used for examining which of the two lines is a better fit:

1. y denotes the observed response for experimental unit i

2. x, denotes the predictor value for experimental unit i

3. y is the predicted response (or fitted value) for experimental unit i

Then, the equation for the best fitting line using a sum of the error estimating function is:

$$y_i' = a_0' + a_1' x_i,$$

where n'o and n'l £tre the coefficients in Equation (6.10). Use of the above equation to predict the actual response 7i› leads to a prediction error (or residual error) of size

Least Square Estimation

Assume n data-points, i - 1, 2, ..., n. A line out of two lines (Figure 6.6) that fits the data best will be one for which the sum of the squares of the n prediction errors (one for each observed data point) is as small as possible. This is the „least squares criterion", which says that the best fit is one, which „minimizes the sum of the squared prediction errors". This implies that when the equation of the best fitting line is:

$$y_i' = b_0 + b_1 x_i,$$

where bo and bl are the coefficients which minimize the errors. The coefficients values make the sum of the squared prediction errors as small as possible. Thus,

$$\text{Minimize } Q = \sum_{i=1}^{n} (y_i - y_i')^2$$

$$Q = \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i)^2$$

Q is also called chi-square function. To minimize compute the derivative with respect to bo and b1, set to 0, respectively, and get the „least squares estimates" for bo and bl as follows:

$$b_0 = \bar{y} - b_1 \bar{x},$$

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

b Explain with neat diagram K-means clustering.

PNG.2

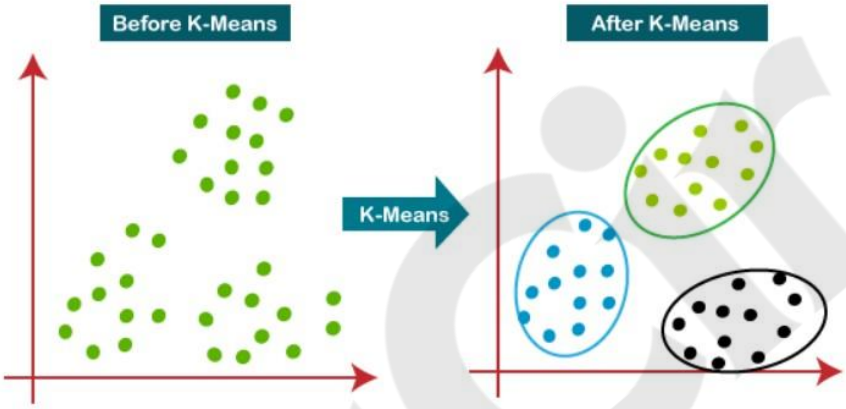
It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs to only one group that has similar properties.

* K-Means Clustering is an unsupervised learning algorithm that is used to solve clustering problems in machine learning or data science.

* It allows us to cluster the data into different groups and is a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

* It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

* The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.



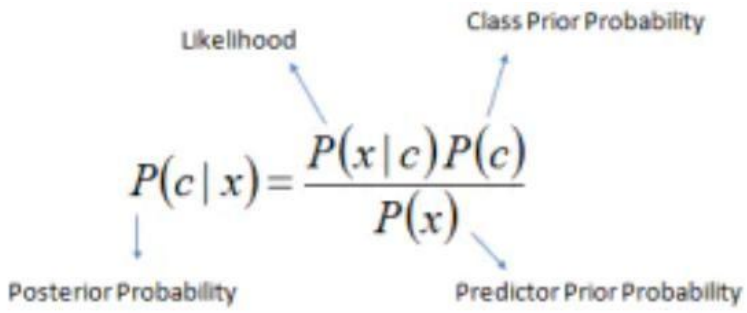
K-means clustering is the unsupervised machine learning algorithm that is part of a much deeper pool of data techniques and operations in the realm of Data Science. It is the fastest and most efficient algorithm to categorize data points into groups, even when very little information is available about data.

c Explain Naïve Bayes Theorem with example.

Naïve Bayes Theorem

Provides a way of computing posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

PNG.2



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

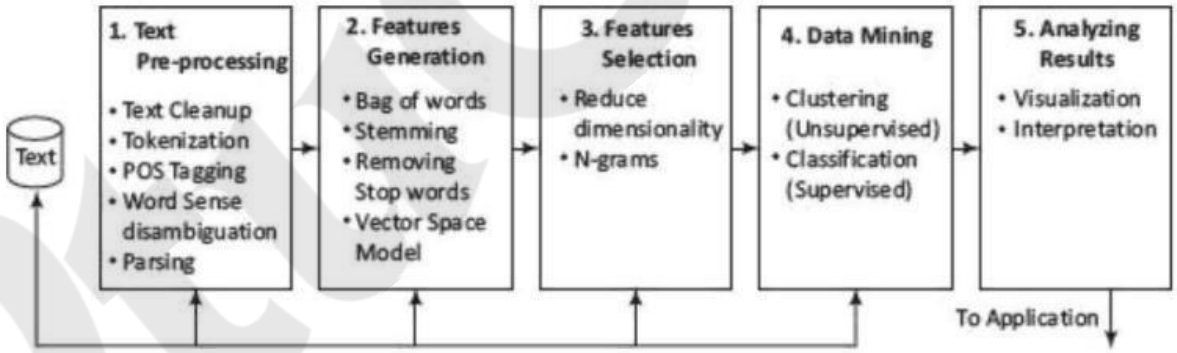
Above,

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of the *predictor* given *class*.
- $P(x)$ is the prior probability of the *predictor*.

EXAMPLE:

REFER TEXTBOOK

Q. 10 a Explain five phases in a process pipeline text mining.



The five phases for processing text are asfollows:

Phase 1: **Text pre-processing** enables Syntactic/Semantic text-analysis and does the followings:

1. Text cleanup is a process of removing unnecessary or unwanted information. Text cleanup converts the raw data by filling up the missing values, identifies and removes outliers, and resolves the inconsistencies. For example, removing comments, removing or

PNG.2

escaping "%20" from URL for the web pages or cleanup the typing error, such as teh (the), do n't (do not) [%20 specifies space in a URL].

2. Tokenization is a process of splitting the cleanup text into tokens (words) using white spaces and punctuation marks as delimiters.

3. Part of Speech (POS) tagging is a method that attempts labeling of each token (word) with an appropriate POS. Tagging helps in recognizing names of people, places, organizations and titles. English language set includes the noun, verb, adverb, adjective, prepositions and conjunctions. Part of Speech encoded in the annotation system of the Penn Treebank Project has 36 POS tags.4

4. Word sense disambiguation is a method, which identifies the sense of a word used in a sentence; that gives meaning in case the word has multiple meanings. The methods, which resolve the ambiguity of words can be context or proximity based. Some examples ofsuch words are bear, bank, cell and bass.

5. Parsing is a method, which generates a parse-tree for each sentence. Parsing attempts and infers the precise grammatical relationships between different words in a given sentence.

Phase 2: Features Generation is a process which first defines features (variables, predictors). Some of the ways of feature generations are:

1. Bag of words-Order of words is not that important for certain applications. Text document is represented by the words it contains (and their occurrences). Document classification methods commonly use the bag-of-words model. The preprocessing of a document first provides a document with a bag of words. Document classification methods then use the occurrence (frequency) of each word as a feature for training a classifier. Algorithms do not directly apply on the bag of words, but use the frequencies.

2. Stemming-identifies a word by its root. (i) Normalizes or unifies variations of the same concept, such as speak for three variations, i.e., speaking, speaks, speakers denoted by [speaking, speaks, speaker- + speak] (ii) Removes plurals, normalizes verb tenses and remove affixes. Stemming reduces the word to its most basic element. For example, impurification -+ pure.

3. Removing stop words from the feature space-they are the common words, unlikely to help text mining. The search program tries to ignore stop words. For example, ignores a, at, for, it, in and are.

4. Vector Space Model (VSM)-is an algebraic model for representing text documents as vector of identifiers, word frequencies or terms in the document index. VSM uses the method of term frequency-inverse document frequency (TF-IDF) and evaluates how important is a word in a document.

When used in document classification, VSM also refers to the bag-of-words model. This bag of words is required to be converted into a term-vector in VSM. The term vector provides the numeric values corresponding to each term appearing in a document. The term vector is very helpful in feature generation and selection.

Term frequency and inverse document frequency (IDF) are important metrics in text analysis.

TF-IDF weighting is most common• Instead of the simple TF, IDF is used to weight the importance of word in the document.

Phase 3: Features Selection is the process that selects a subset of features by rejecting irrelevant and/or redundant features (variables, predictors or dimension) according to defined criteria. Feature selection process does the following:

1. Dimensionality reduction-Feature selection is one of the methods of division and therefore, dimension reduction. The basic objective is to eliminate irrelevant and redundant data. Redundant features are those, which provide no extra information. Irrelevant features provide no useful or relevant information in any context. Principal Component Analysis (PCA) and Linear Discriminate Analysis (LDA) are dimension reduction methods. Discrimination ability of a feature measures relevancy of features. Correlation helps in finding the redundancy of the feature. Two features are redundant to each other if their values correlate with each other.

2. N-gram evaluation-finding the number of consecutive words of interest and extract them. For example, 2-gram is a two words sequence, ["tasty food", "Good one"]. 3-gram is a three words sequence, ["Crime Investigation Department"].

3. Noise detection and evaluation of outliers methods do the identification of unusual or suspicious items, events or observations from the data set. This step helps in cleaning the data.

The feature selection algorithm reduces dimensionality that not only improves the performance of learning algorithm but also reduces the storage requirement for a dataset. The process enhances data understanding and its visualization.

Phase 4: Data mining techniques enable insights about the structured database that resulted from the previous phases. Examples of techniques are:

1. Unsupervised learning (for example, clustering)

(i) The class labels (categories) of training data are unknown

(ii) Establish the existence of groups or clusters in the data Good clustering methods use high intra-cluster similarity and low inter-cluster similarity.

Examples of uses - biogs, pattern and trends.

2. Supervised learning (for example, classification)

(i) The training data is labeled indicating the class

(ii) New data is classified based on the training set Classification is correct when the known label of test sample is identical with the resulting class computed from the classification model.

Examples of uses are news filtering application, where it is required to automatically assign incoming documents to pre-defined categories; email spam filtering, where it is identified whether incoming email messages are spam or not.

Example of text classification methods are Naive Bayes Classifier and SVMs.

3. Identifying evolutionary patterns in temporal text streams-the method is useful in a wide range of applications, such as summarizing of events in news articles and extracting the research trends in the scientific literature.

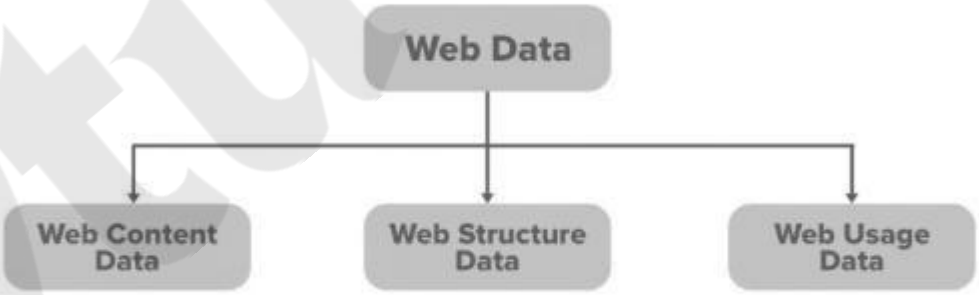
Phase 5: <u>Analysing results</u>

(i) Evaluate the outcome of the complete process.
(ii) Interpretation of Result- If acceptable then results obtained can be used as an input for next set of sequences. Else, the result can be discarded, and try to understand what and why the process failed.
(iii) Visualization - Prepare visuals from data, and build a prototype.
(iv) Use the results for further improvement in activities at the enterprise, industry or institution.

b Explain Web Usage Mining.

Web usage mining, a subset of Data Mining, is basically the extraction of various types of interesting data that is readily available and accessible in the ocean of huge web pages, Internet- or formally known as World Wide Web (WWW). Being one of the applications of data mining technique, it has helped to analyze user activities on different web pages and track them over a period of time. Basically, Web Usage Mining can be divided into 2 major subcategories based on web usage data.

There are 3 main types of web data:



1. Web Content Data: The common forms of web content data are HTML, web pages, images audio-video, etc. The main being the HTML format. Though it may differ from browser to browser the common basic layout/structure would be the same everywhere. Since it's the most popular in web content data. XML and dynamic server pages like JSP, PHP, etc. are also various forms of web content data.

PNG.2

2. Web Structure Data: On a web page, there is content arranged according to HTML tags (which are known as intrapage structure information). The web pages usually have hyperlinks that connect the main webpage to the sub-web pages. This is called Inter-page structure information. So basically relationship/links describing the connection between webpages is web structure data.

3. Web Usage Data: The main source of data here is-Web Server and Application Server. It involves log data which is collected by the main above two mentioned sources. Log files are created when a user/customer interacts with a web page. The data in this type can be mainly categorized into three types based on the source it comes from:

- Server-side
- Client-side
- Proxy side.

Advantages of Web Usage Mining

- Government agencies are benefited from this technology to overcome terrorism.
- Predictive capabilities of mining tools have helped identify various criminal activities.
- Customer Relationship is being better understood by the company with the aid of these mining tools. It helps them to satisfy the needs of the customer faster and efficiently.

Disadvantages of Web Usage Mining

- Privacy stands out as a major issue. Analyzing data for the benefit of customers is good. But using the same data for something else can be dangerous. Using it within the individual's knowledge can pose a big threat to the company.
- Having no high ethical standards in a data mining company, two or more attributes can be combined to get some personal information of the user which again is not respectable.