CBCS SCHEME

USN | | | | | | | | | | |

18CS72

Seventh Semester B.E. Degree Examination, Feb./Mar.2022
Big Data Analytics

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Discuss the Evolution of Big Data. **(06 Marks)**

b. Explain the characteristics of Big Data. **(04 Marks)**

c. With a neat block diagram, explain Data Architecture Design. **(10 Marks)**

OR

2 a. Write notes on Analytics Scalability to Big Data and Massive Parallel Processing Platforms. **(12 Marks)**

b. Highlight Big Data Analytics applications with one case study. **(08 Marks)**

Module-2

3 a. What are the core components of Hadoop? Explain in brief its each of its components. **(10 Marks)**

b. Explain Hadoop Distributed File System. **(10 Marks)**

OR

4 a. Define MapReduce Frame work and its functions. **(06 Marks)**

b. Write down the steps on the request to MapReduce and the types of process in MapReduce. **(10 Marks)**

c. Write short notes on Flume Hadoop Tool. **(04 Marks)**

Module-3

5 a. Discuss the characteristics of NoSQL data store along with the features in NoSQL transactions. **(08 Marks)**

b. With neat diagrams, explain the following for shared Nothing Architecture for Big Data Tasks.

(i) Single Server model
(ii) Sharding very large databases
(iii) Master Slave distribution model.
(iv) Peer-to-Peer distribution model. **(12 Marks)**

OR

6 a. Define key-value store with example. What are the advantages of key-value store? **(10 Marks)**

b. Write down the steps to provide client to read and write values using key-value store. What are the typical uses of key value store? **(10 Marks)**

Module-4

7 a. With a neat diagram, explain the process in MapReduce when client submitting a Job. **(10 Marks)**

b. Explain Hive Integration and work flow steps involved with a diagram. **(10 Marks)**

1a. Discuss the evolution of big data.

The rise in technology has led to the production and storage of voluminous amounts of data. Earlier megabytes (106 B) were used but nowadays petabytes (1015 B) 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 1.1 shows data usage and growth. As size and complexity increase, the proportion of unstructured data types also increase. An example of a traditional tool for structured data storage and querying is RDBMS. Volume, velocity and variety (3Vs) of data need the usage of number of programs and tools for analyzing and processing at a very high speed.

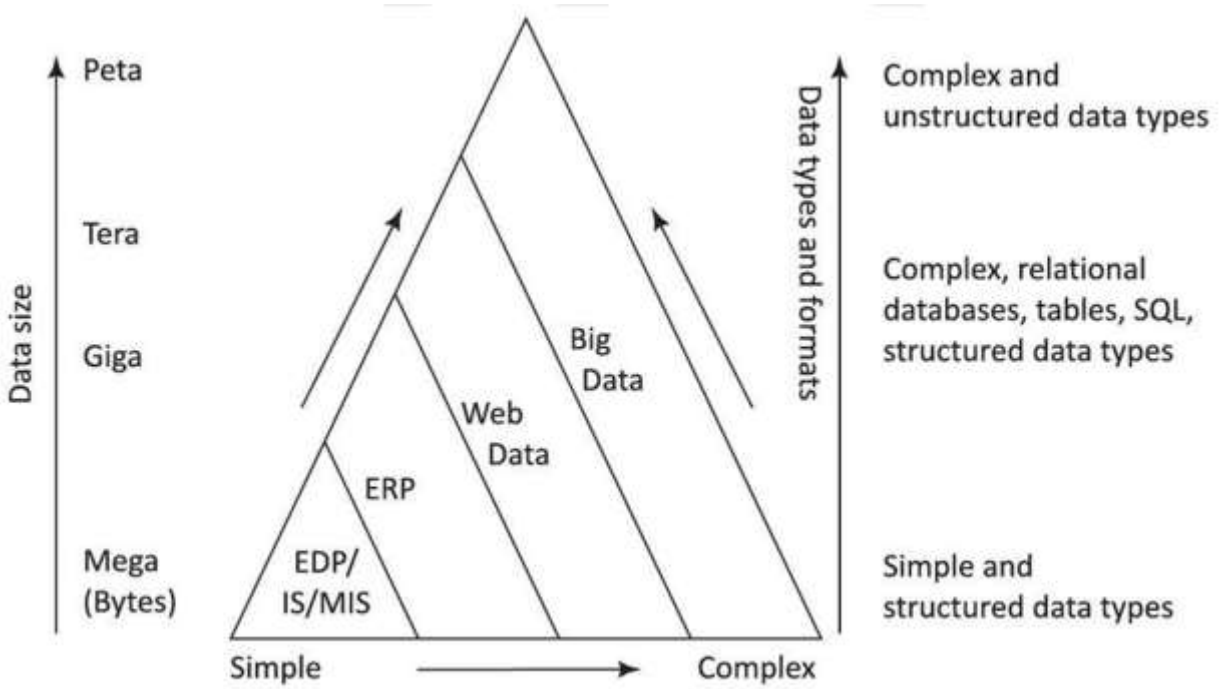


Figure 1.1 Evolution of Big Data and their characteristics

1b. Explain the characteristics of Big data.

Big Data Characteristics

Volume: is related to size of the data

Veloctiy: refers to the speed of generation of data.

Variety: comprises of a variety of data

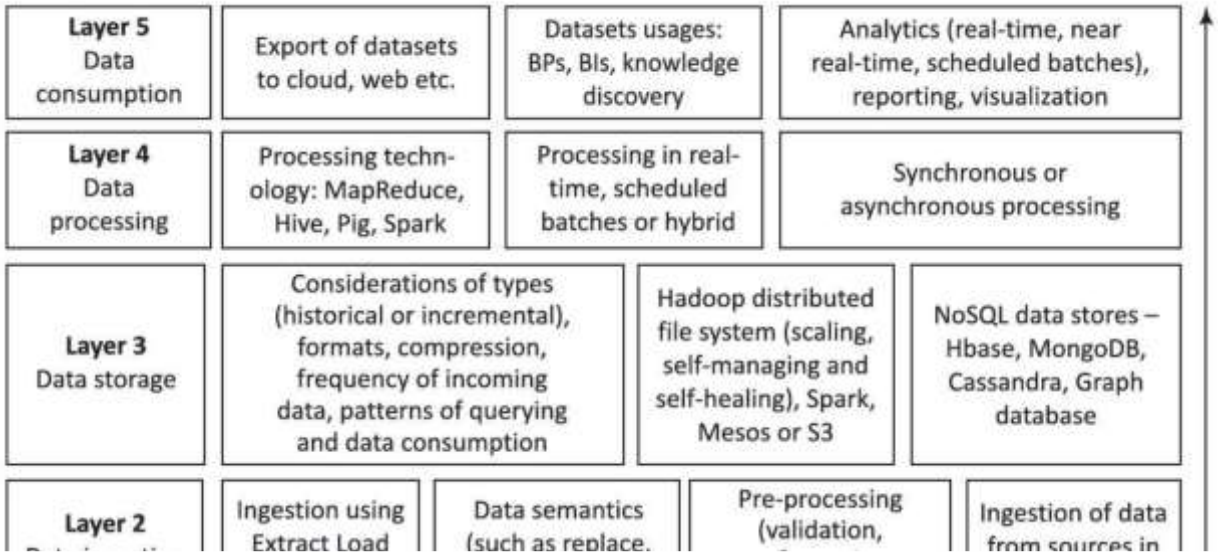
Veracity: quality of data captured, which can vary greatly, affecting its accurate analysis

1c. With a neat diagram explain Big data architecture design

Big Data architecture is the logical and/or physical layout/structure of how Big Data willbe stored, accessed and managed within a Big Data or IT environment. Architecture logically defines how Big Data solution will work, the core components (hardware, database, software, storage) used, flow of information, security and more.Data analytics need the number of sequential steps. Big Data architecture design task simplifies when using the logical layers approach. Figure 1.2 shows the logical layers and the functions which are considered in Big Data architecture.

Data processing architecture consists of five layers:

- identification of data sources,
- acquisition, ingestion, extraction, pre-processing, transformation of data,
- Data storage at files, servers, cluster or cloud,
- data-processing,
- data consumption in the number of programs and tools.



Data ingestion and acquisition	and Transform (ELT)	append, aggregate, compact, fuse)	transformation or transcoding) requirement	batches or real time
Layer 1 Identification of internal and external sources of data	Sources for ingestion of data	Push or pull of data from the sources for ingestion	Data types for database, files, web or service	Data formats: structured, semi- or unstructured for ingestion

Fig. Design of logical layers in a data processing architecture, and functions in the layers

2a. Write notes on analytical scalability of Big data and massively parallel processors

Following are the techniques deployed for Big Data storage, applications, data management and mining and analytics:

- Huge data volumes storage, data distribution, high-speed networks and high performance computing
- Applications scheduling using open source, reliable, scalable, distributed file system, distributed database, parallel and distributed computing systems, such as Hadoop or Spark
- Open source tools which are scalable, elastic and provide virtualized environment, clusters of data nodes, task and thread management
- Data management using NoSQL, document database, column-oriented database, graph database and other form of databases used as per needs of the applications and in memory data management using columnar or Parquet formats during program execution,Data mining and analytics, data retrieval, data reporting, data visualization and machine learning Big Data tools.

Scalability and 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.
- Processing of this much distributed data within a short time and at minimum cost is Problematic.
- Scalability is the capability of a system to handle the workload as per the magnitude of the work.
- System capability needs increment with the increased workloads.
- When the workload and complexity exceed the system capacity, scale it up and scale it out.
- Scalability enables increase or decrease in the capacity of data storage, processing&

Vertical scalability means scaling up the given system's resources and increasing the system's analytics, reporting and visualization capabilities. This is an additional way to solve problems of greater complexities. Scaling up means designing the algorithm according to the architecture that

uses resources efficiently. x terabyte of data take time t for processing, code size with increasing complexity increase by factor n, then scaling up means that processing takes equal, less or much less than (n * t).

Horizontal scalability means increasing the number of systems working in coherence and scaling out the workload. Processing different datasets of a large dataset deploys horizontalscalability. Scaling out means using more resources and distributing the processing and storage tasks in parallel. The easiest way to scale up and scale out 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.

Massive Parallel Processing Platforms

- Parallelization of tasks can be done at several levels:
- Distributing separate tasks onto separate threads on the same CPU
- Distributing separate tasks onto separate CPUs on the same computer
- Distributing separate tasks onto separate computers.

2b. Highlight big data applications with on case study.

- **Big Data in Marketing and Sales Data** are important for most aspect of marketing, sales and advertising. Customer Value (CV) depends on three factors - quality, service and price. Big data analytics deploy large volume of data to identify and derive intelligence using predictive models about the individuals. The facts enable marketing companies to decide what products to sell. A definition of marketing is the creation, communication and delivery of value to customers. Customer (desired) value means what a customer desires from a product. Customer (perceived) value means what the customer believes to have received from a product after purchase of the product. Customer value analytics (CVA) means analyzing what a customer really needs. CVA makes it possible for leading marketers, such as Amazon to deliver the consistent customer experiences.

- **Big Data Analytics in Detection of Marketing Frauds** Big Data analytics enable fraud detection. Big Data usages has the following features-for enabling detection and prevention of frauds:

➢ Fusing of existing data at an enterprise data warehouse with data from sources such as social media, websites, biogas, e-mails, and thus enriching existing data
➢ Using multiple sources of data and connecting with many applications
➢ Providing greater insights using querying of the multiple source data
➢ Analyzing data which enable structured reports and visualization
➢ Providing high volume data mining, new innovative applications and thus leading to new business intelligence and knowledge discovery

- **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. Five data risks, described by Bernard Marr are data security, data privacy breach, costs affecting profits, bad analytics and bad data

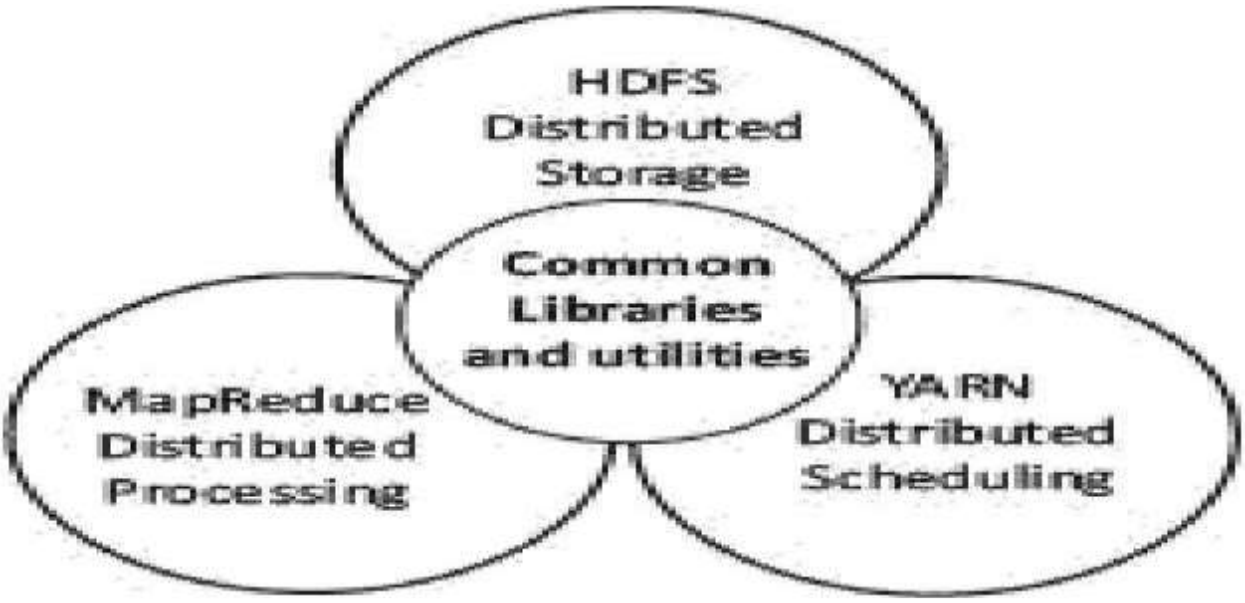
- **Big Data Credit Card Risk Management**

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:
1. Identifying high credit rating business groups and individuals,
2. Identifying risk involved before lending money
3. Identifying industrial sectors with greater risks
4. Identifying types of employees (such as daily wage earners in construction sites) and Businesses (such as oil exploration) with greater risks
5. Anticipating liquidity issues (availability of money for further issue of credit and Re scheduling credit installments) over the years.

- **Big Data in Medicine**

Big Data analytics deploys large volume of data to identify and derive intelligence using predictive models about individuals. Big Data driven approaches help in research in medicine which can help patients Following are some findings: building the health profiles of individual patients and predicting models for diagnosing better and offer better treatment, Aggregating large volume and variety of information around from multiple sources the DNAs, proteins, and metabolites to cells, tissues, organs, organisms, and ecosystems, that can enhance the understanding of biology of diseases. Big data creates patterns and models by data mining and help in better understanding and research, Deploying wearable devices data, the devices data records during active as well as inactive periods, provide better understanding of patient health, and better risk profiling the user for certain diseases.

3a. what are core components of Hadoop? Explain in brief each of its components



Core components of Hadoop

Hadoop core components of the framework are:

Hadoop Common - The common module contains the libraries and utilities that are required by the other modules of Hadoop. For example, Hadoop common provides various components and interfaces for distributed file system and general input/output. This includes serialization, Java RPC (Remote Procedure Call) and file-based data structures.

Hadoop Distributed File System (HDFS) - A Java-based distributed file system which can store all kinds of data on the disks at the clusters.

MapReduce vl - Software programming model in Hadoop 1 using Mapper and Reducer. The vl processes large sets of data in parallel and in batches.
YARN - Software for managing resources for computing. The user application tasks or subtasks run in parallel at the Hadoop, uses scheduling and handles the requests for the resources in distributed running of the tasks. MapReduce v2 - Hadoop 2 YARN-based system for parallel processing of large datasets and distributed processing of the application tasks.

3b. Explain Hadoop distributed file system

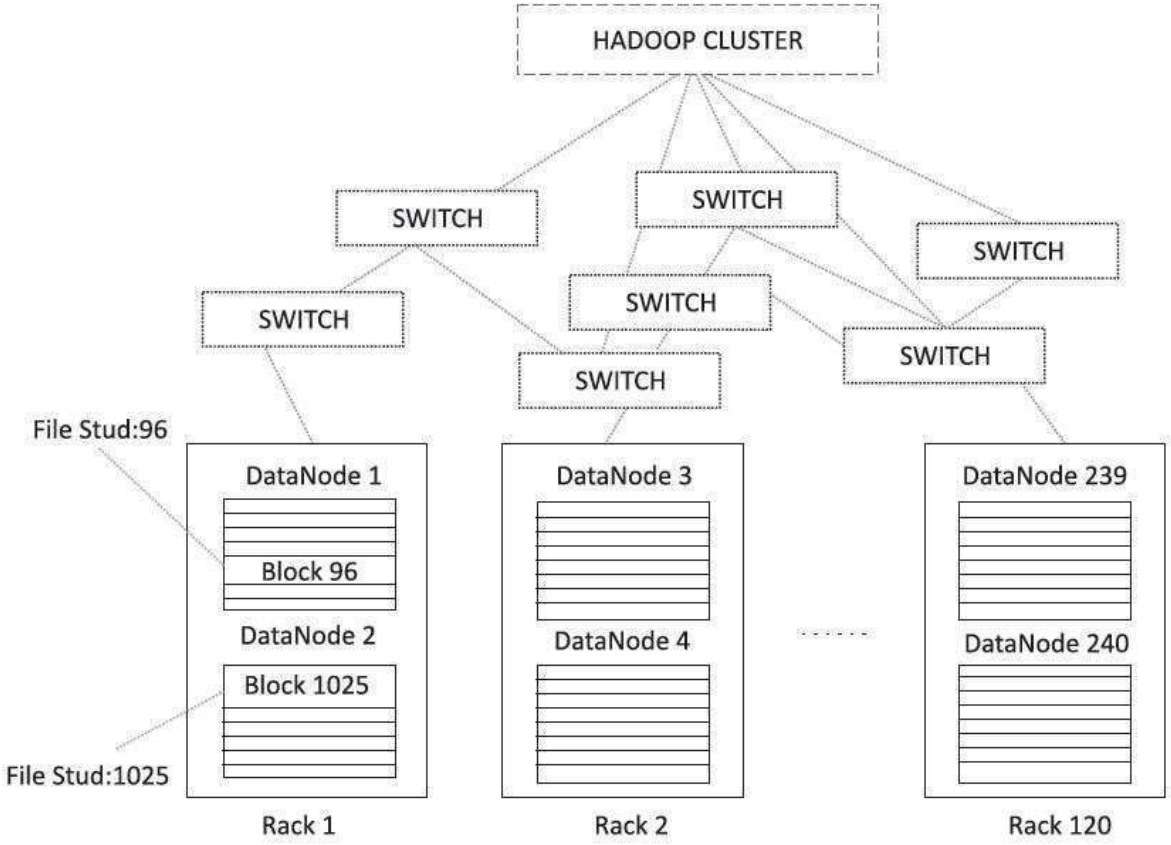
HDFS is a core component of Hadoop. HDFS is designed to run on a cluster of computers and servers at cloud-based utility services. HDFS stores Big Data which may range from GBs (1 GB= 230 B) to PBs (1 PB=1015 B, nearly the 250 B). HDFS stores the data in a distributed manner in order to compute fast. The distributed data store in HDFS stores data in any format regardless of schema.

HDFS Storage

Hadoop data store concept implies storing the data at a number of dusters. Each cluster has a number of data stores, called racks. Each rack stores a number of Data Nodes. Each Data Node has a large number of data blocks. The racks distribute across a cluster. The nodes have processing and storage capabilities. The nodes have the data in data blocks to run the application tasks. The data blocks replicate by default at least on three Data Nodes in same or remote nodes. Data at the stores enable running the distributed applications including analytics, data mining, OLAP using the clusters. A file, containing the data divides into data blocks. A data block default size is 64 MBs.

Hadoop HDFS features are as follows

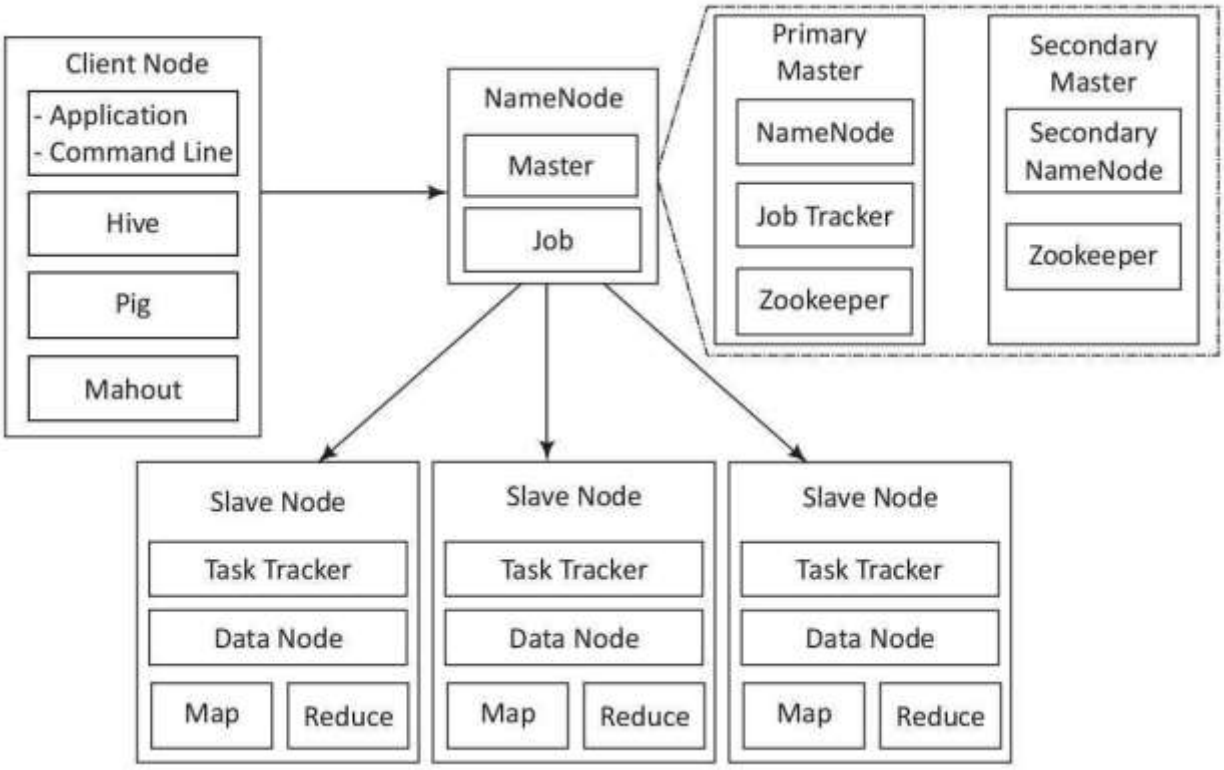
i. Create, append, delete, rename and attribute modification functions
ii. Content of individual file cannot be modified or replaced but appended with new data at the end of the file
iii. Write once but read many times during usages and processing
iv. Average file size can be more than 500 MB.
Figure



A Hadoop cluster example

Hadoop Physical organization

Figure below shows the client, master Name Node, primary and secondary Master Nodes and slave nodes in the Hadoop physical architecture. Clients as the users run the application with the help of Hadoop ecosystem projects. For example, Hive, Mahout and Pig are the ecosystem's projects. They are not required to be present at the Hadoop cluster.



4a. Define Map reduce frame work and its functions

Mapper means software for doing the assigned task after organizing the data blocks imported using the keys. A key specifies in a command line of Mapper. The command maps the key to the data, which an application uses. Reducer means software for reducing the mapped data by using the aggregation, query or user specified function. The reducer provides a concise cohesive response for the application. Aggregation function means the function that groups the values of multiple rows together to result a single value of more significant meaning or measurement. For example, function such as count, sum, maximum, minimum, deviation and standard deviation. Querying function means a function that finds the desired values. For example, function for finding a best student of a class who has shown the best performance in examination. MapReduce allows writing applications to process reliably the huge amounts of data, in parallel, on large clusters of servers. The cluster size does not limit as such to process in parallel. The parallel programs of MapReduce are useful for performing large scale data analysis using multiple machines in the cluster.

Features of MapReduce framework are as follows:

- Provides automatic parallelization and distribution of computation based on several Processors
- Processes data stored on distributed clusters of Data Nodes and racks
- Allows processing large amount of data in parallel
- Provides scalability for usages of large number of servers
- Provides Map Reduce batch-oriented programming model in Hadoop version 1
- Provides additional processing modes in Hadoop 2 YARN-based system and enables required parallel processing. For example, for queries, graph databases, streaming data, messages, real-time OLAP and ad hoc analytics with Big Data 3V

4b. Write down the steps on the request of MapReduce and the types of processes in MapReduce

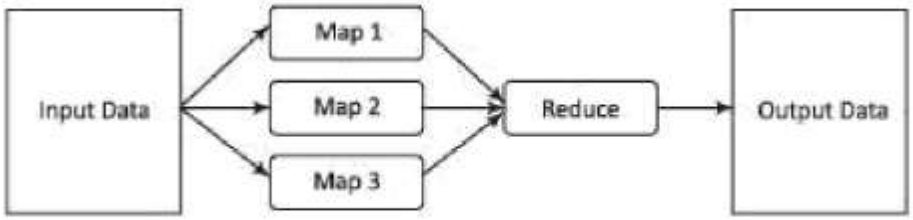


Figure 4.2 MapReduce Programming Model

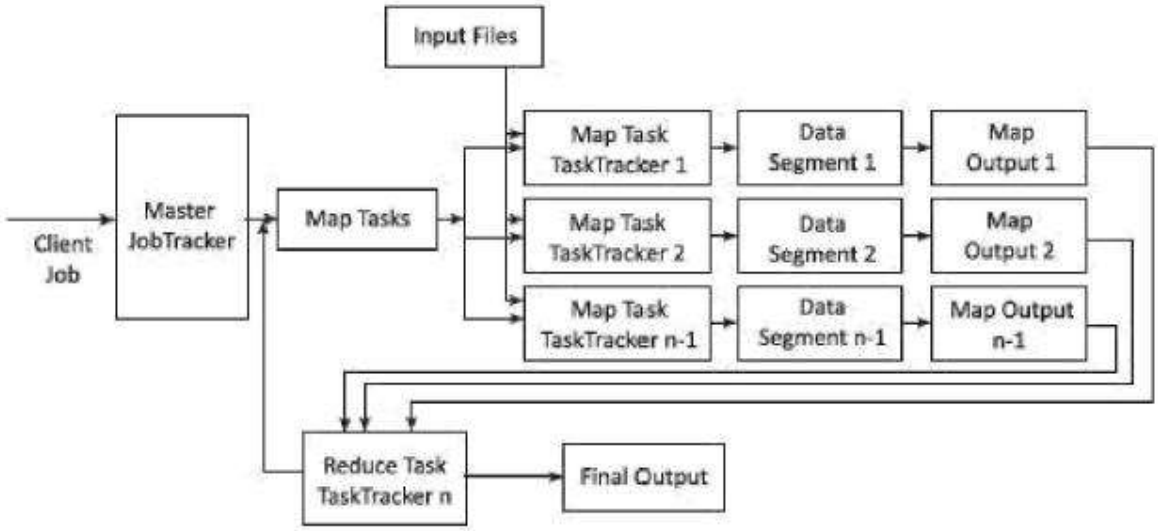


Figure 4.3 MapReduce process on client submitting a job

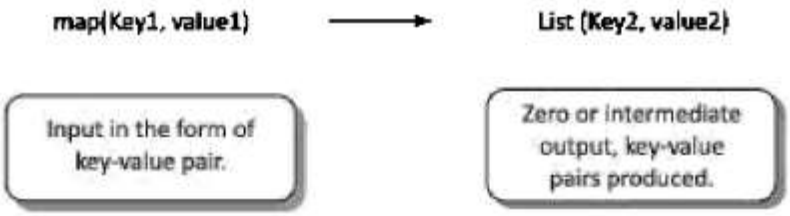


Figure 4.4 Logical view of functioning of map()

The Sample Code for Mapper Class
public clase SampleMapper extends Mapper<kl, Vl, k2, *v2*>
{
void map (kl key, Vl value, Context context) throwe IOException, InterruptedException
{ ..}

Individual Mappers do not communicate with each other.

Number of Maps The number of maps depends on the size of the input files, i.e., the total number of blocks of the input files.If the input files are of 1TB in size and the block size is 128 MB, there will be 8192 maps. The number of map task Nmap can be explicitly set by using *setNumMapTasks(int)*. Suggested numberis nearly 10-100 maps per node. Nmap can be set even higher.

Key-Value Pair
Each phase (Map phase and Reduce phase) of MapReduce has key-value pairs as input and output. Data should be first converted into key-value pairs before it is passed to the Mapper, as the Mapper only understands key-value pairs of data.
Key-value pairs in Hadoop MapReduce are generated as follows:
InputSplit - Defines a logical representation of data and presents a Split data for processing at individual map().
RecordReader - Communicates with the InputSplit and converts the Split into records which are in the form of key-value pairs in a format suitable for reading by the Mapper.
RecordReader uses TextInputFormat by default for converting data into key-value pairs. ***RecordReader*** communicates with the InputSplit until the file is read

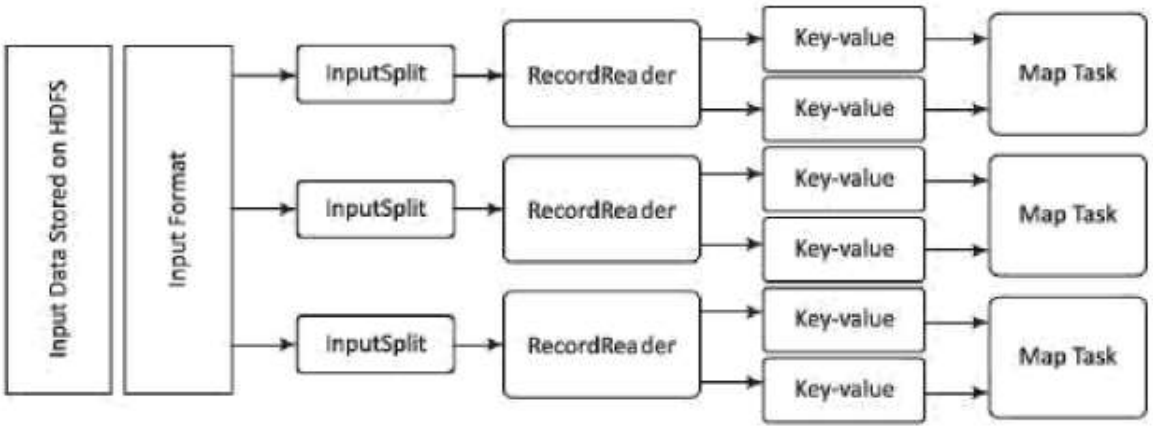


Figure 4.5 Key-value pairing in MapReduce

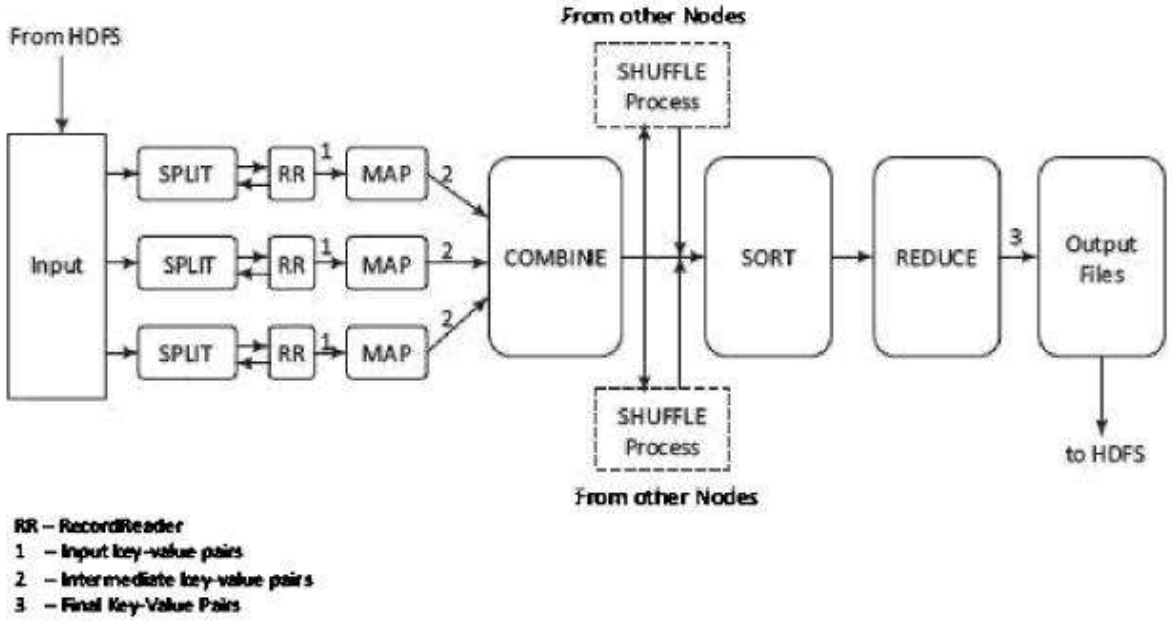
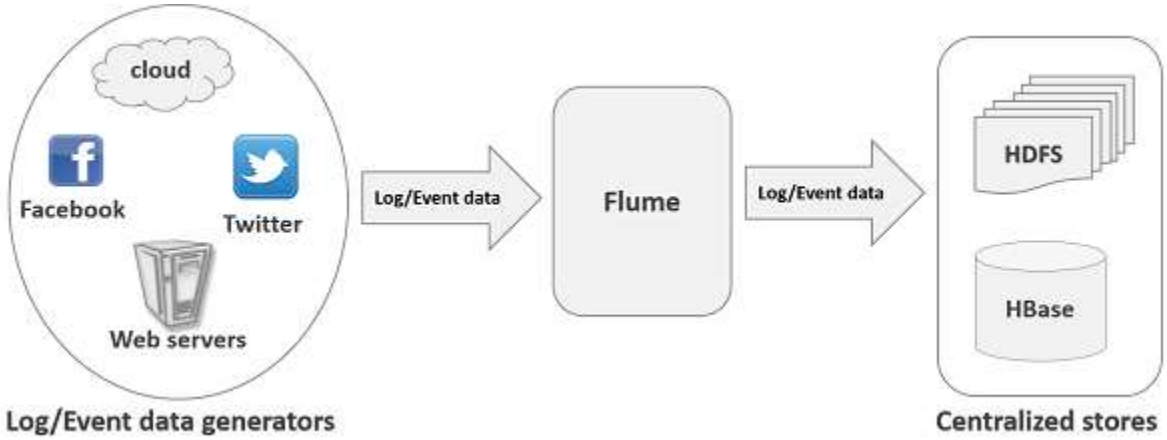


Figure 4.6 MapReduce execution steps

4c. Write notes on FLUME Hadoop tool

Apache Flume is a tool/service/data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log files, events (etc...) from various sources to a centralized data store.

Flume is a highly reliable, distributed, and configurable tool. It is principally designed to copy streaming data (log data) from various web servers to HDFS.



Applications of Flume

Assume an e-commerce web application wants to analyze the customer behavior from a particular region. To do so, they would need to move the available log data in to Hadoop for analysis. Here, Apache Flume comes to our rescue.

Flume is used to move the log data generated by application servers into HDFS at a higher speed.

Advantages of Flume

Here are the advantages of using Flume −

- Using Apache Flume we can store the data in to any of the centralized stores (HBase, HDFS).
- When the rate of incoming data exceeds the rate at which data can be written to the destination, Flume acts as a mediator between data producers and the centralized stores and provides a steady flow of data between them.
- Flume provides the feature of **contextual routing**.
- The transactions in Flume are channel-based where two transactions (one sender and one receiver) are maintained for each message. It guarantees reliable message delivery.
- Flume is reliable, fault tolerant, scalable, manageable, and customizable.

5a. Discuss the characteristics of NOSQL data store along with the features in NOSQL transactions.

NoSQL data store characteristics are as follows:

1. NoSQL is a class of non-relational data storage system with flexible data model. Examples of NoSQL data-architecture patterns of datasets are key-value pairs, name/value pairs, Column family,Big-data store, Tabular data store, Cassandra (used in Facebook/Apache), HBase, hash table [Dynamo (Amazon S3)], unordered keys using]SON (CouchDB),]SON (PNUTS),]SON (MongoDB), Graph Store, Object Store, ordered keys and semi-structured data storage systems.

2. NoSQL not necessarily has a fixed schema, such as table; do not use the concept of Joins (in distributed data storage systems); Data written at one node can be replicated to multiple nodes. Data store is thus faulttolerant. The store can be partitioned into unshared shards.

Features in NoSQL Transactions

NoSQL transactions have following features:

1. Relax one or more of the ACID properties.

2. Characterize by two out of three properties (consistency, availability and partitions) of CAP theorem, two are at least present for the application/ service/process.

3. Can be characterized by BASE properties

NoSQL data stores and their characteristic features

Apache's HBase	HDFS compatible, open-source and non-relational data store written inJava; A

	column-family based NoSQL data store, data store providing BigTable-like capabilities (Sections 2.6 and 3.3.3.2); scalability, strong consistency, versioning, configuring and maintaining data store characteristics
Apache's MongoDB	HDFS compatible; master-slave distribution model (Section 3.5.1.3); document-oriented data store withJSON-like documents and dynamic schemas; open-source, NoSQL, scalable and non-relational database; used by Websites Craigslist, eBay, Foursquare at the backend
Apache's Cassandra	HDFS compatible DBs; decentralized distribution peer-to-peer model (Section 3.5.1.4); open source; NoSQL; scalable, non-relational, column- family based, fault-tolerant and tuneable consistency (Section 3.7) used by Facebook and Instagram
Oracle NoSQL	Step towards **NoSQL data store; distributed key-value data store; provides transactional semantics for data manipulation ,** horizontal scalability, simple administration and monitoring

5b. with neat diagrams explain the following for shared nothing Architecture for Big Data tasks.

1. **Single server Model**
2. **Sharding very large databases**
3. **Master Slave distribution model**
4. **Peer to peer distribution model**

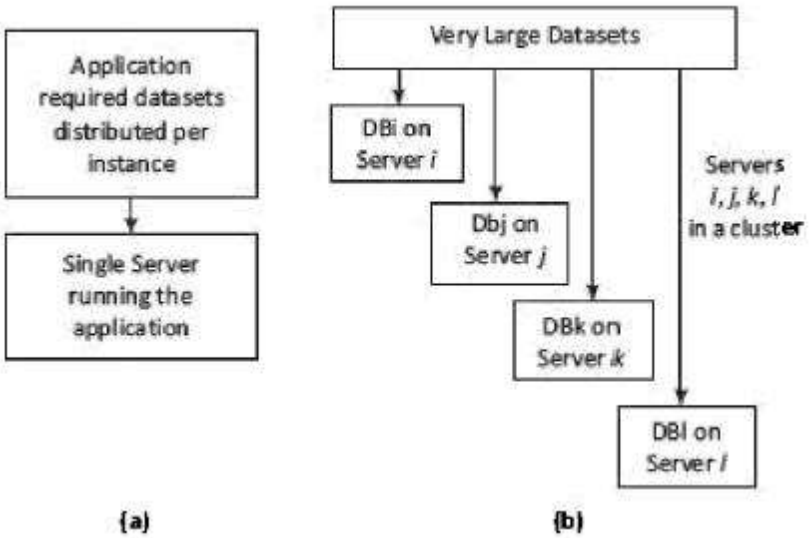
Single Server Model

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. Aggregates of datasets may be key-value, column-family or BigTable data stores which require sequential processing. These data stores also use the SSD model. An application

executes the data sequentially on a single server. Figure 3.9(a) shows the SSD model. Process and datasets distribute to a single server which runs the application.

Sharding Very Large Databases

Figure shows sharding of very large datasets into four divisions, each running the application on four i,j, k and l different servers at the cluster. DBi, DBj, DBk and DB1 .

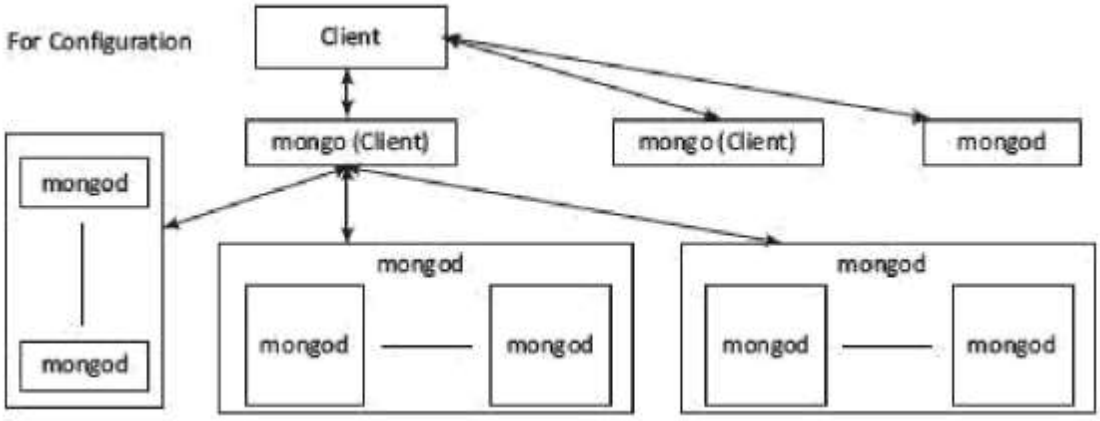


(a) Single server model (b) Shards distributed on four servers in a cluster

Master Slave Distribution

Master directs the slaves. Slave nodes data replicate on multiple slave servers in Master Slave Distribution (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. Figure below shows an example of MongoDB. MongoDB database server is mongodb and the client is mongo.

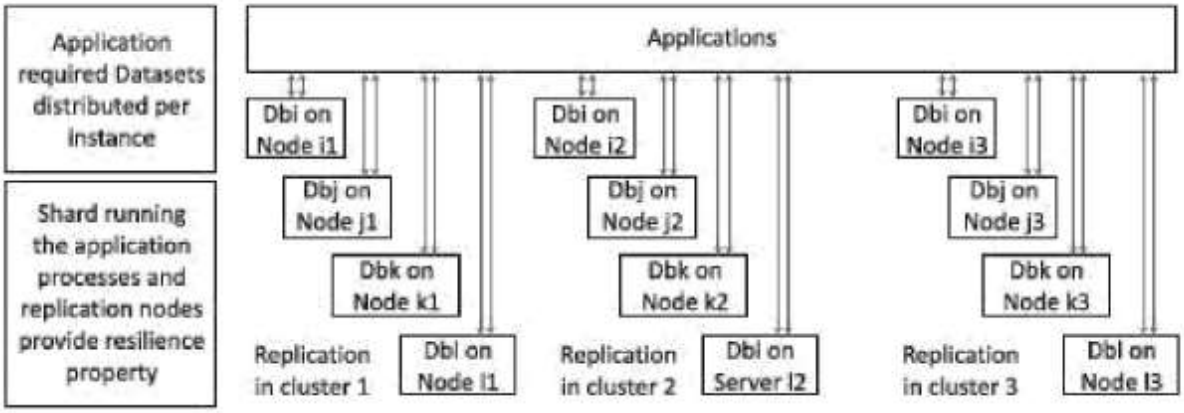
Master-Slave Replication Processing performance decreases due to replication in MSD distribution model. Resilience for read operations is high, which means if in case data is not available from a slave node, then it becomes available from the replicated nodes. Master uses the distinct write and read paths. Complexity Cluster-based processing has greater complexity than the other architectures. Consistency can also be affected in case of problem of significant time taken for updating.



Master-slave distribution model. Mongo is a client and mangodb is the server

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, as other replicated node responds. Cassandra adopts the PPD model. The data distributes among all the nodes in 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. When a write is on different nodes, then write inconsistency occurs.



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

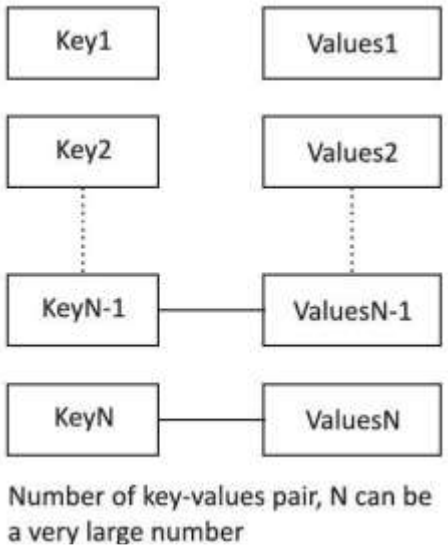
Choosing Master-Slave versus Peer-to-Peer

Master-slave replication provides greater scalability for read operations. Replication provides resilience during the read. Master does not provide resilience for writes. Peer-to-peer replication provides resilience for read and writes both.

Sharing Combining with Replication Master-slave and sharding creates multiple masters. However, for each data a single master exists. Configuration assigns a master to a group of datasets. Peer-to-peer and sharding use same strategy for the column-family data stores. The shards replicate on the nodes, which does read and write operations both.

6a. Define Key value store with example. What are the advantages of storing key value pairs.

The simplest way to implement a schema-less data store is to use key-value pairs. The data store characteristics are high performance, scalability and flexibility. Data retrieval is fast in key-value pairs data store. A simple string called, key maps to a large data string or BLOB (Basic Large Object). Key-value store accesses use a primary key for accessing the values. Therefore, the store can be easily scaled up for very large data. The concept is similar to a hash table where a unique key points to a particular item(s) of data. Figure below shows key-value pairs architectural pattern and example of students' database as key-value pairs



Key	Value
"Ashish"	"Category: Student; Class: B.Tech.; Semester: VII; Branch: Engineering; Mobile:9999912345"
"Mayuri"	"Category: student; class: M.Tech.; Mobile:8888823456"

Advantages of a key-value store are as follows:

1. Data Store can store any data type in a value field. The key-value system stores the information as a BLOB of data (such as text, hypertext, images, video and audio) and return the same BLOB when the data is retrieved. Storage is like an English dictionary. Query for a word retrieves the meanings, usages, different forms as a single item in the dictionary. Similarly, querying for key retrieves the values.
2. A query just requests the values and returns the values as a single item. Values can be of any data type.

3. Key-value store is eventually consistent.

4. Key-value data store may be hierarchical or may be ordered key-value store.

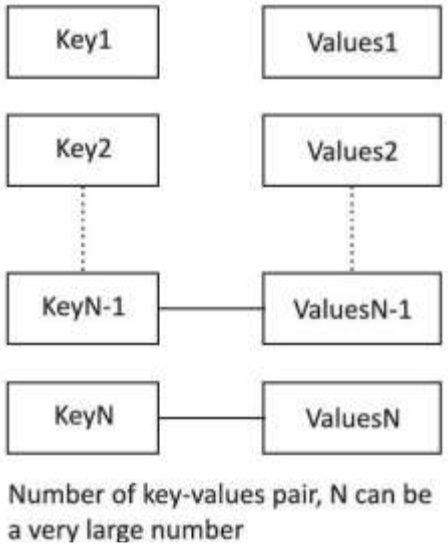
5. Returned values on queries can be used to convert into lists, table columns, data-frame fields and columns.

6. Have (i) scalability, (ii) reliability, (iii) portability and (iv) low operational cost.

7. The key can be synthetic or auto-generated. The key is flexible and can be represented in many formats: (i) Artificially generated strings created from a hash of a value, (ii) Logical path names to images or files, (iii) REST web-service calls (request response cycles), and (iv) SQL queries.

6b. Write down the steps to provide client read and write values using key value store. What are typical uses of key value store?

The simplest way to implement a schema-less data store is to use key-value pairs. The data store characteristics are high performance, scalability and flexibility. Data retrieval is fast in key-value pairs data store. A simple string called, key maps to a large data string or BLOB (Basic Large Object). Key-value store accesses use a primary key for accessing the values. Therefore, the store can be easily scaled up for very large data. The concept is similar to a hash table where a unique key points to a particular item(s) of data. Figure below shows key-value pairs architectural pattern and example of students' database as key-value pairs



Key	Value
"Ashish"	"Category: Student; Class: B.Tech.; Semester: VII; Branch: Engineering; Mobile:9999912345"
"Mayuri"	"Category: student; class: M.Tech.; Mobile:8888823456"

In key-value databases, we work with a very simple data model which resembles dictionary. The syntax for manipulating data is simple. Regardless of the type of an operation, we specify a namespace, and a key to indicate we want to perform an action on a key-value pair. Type of it depends on our call. There are three operations performed on a key-value store: put, get, and delete.

- put adds a new key-value pair to the table or updates a value if this key is already present.
- get returns the value for a given key if it exists.
- delete removes a key and its value from the table if it exists.

Typical uses of key-value store are:

- (i) Image store,
- (ii) Document or file store,
- (iii) Lookup table, and

- (iv) Query-cache.

7a. With a neat diagram explain the process in MAPREDUCE when client submitting a job

Big Data Processing employs the Map Reduce Programming Model. A job means a Map Reduce Program. Each job consists of several smaller unit, called MapReduce Tasks.
A software execution framework in MapReduce programming defines the parallel tasks.
The Hadoop MapReduce implementation uses Java framework

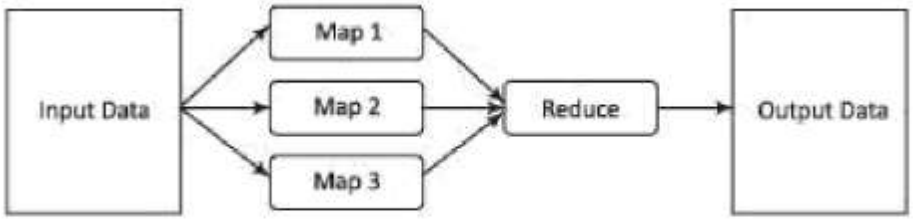


Figure 4.2 MapReduce Programming Model

The model defines two important tasks, namely Map and Reduce.
Map takes input data set as pieces of data and maps them on various nodes for parallel processing. The reduce task, which takes the output from the maps as an input and combines those data pieces into a smaller set of data. A reduce task always run after the map task (s).Many real-world situations are expressible using this model.
I*nner join* is the default natural join. It refers to two tables that join based on common columns mentioned using the ON clause. Inner Join returns all rows from both tables if the columns match.
Node refers to a place for storing data, data block or read or write computations.
Data center in a DB refers to a collection of related nodes. Many nodes form a data center or rack.
Cluster refers to a collection of many nodes.
Keyspace means a namespace to group multiple column families, especially one per partition.
Indexing to a field means providing reference to a field in a document of collections that support the queries and operations using that index. A DB creates an index on the _id field of every collection.
The input data is in the form of an HDFS file. The output of the task also gets stored in the HDFS. The compute nodes and the storage nodes are the same at a cluster, that is, the MapReduce program and the HDFS are running on the same set of nodes.

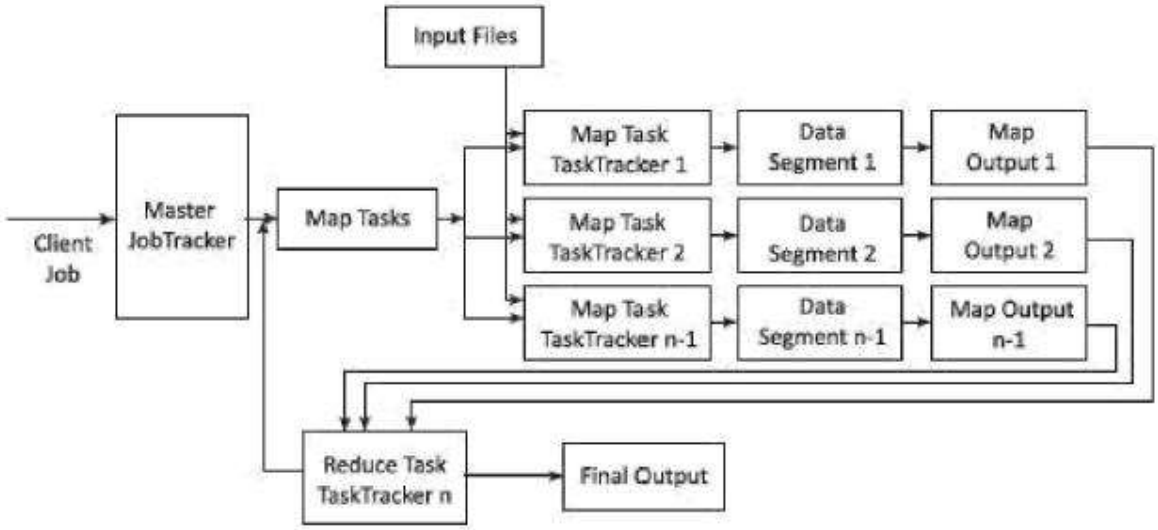


Figure 4.3 MapReduce process on client submitting a job

JobTracker and Task Tracker MapReduce consists of a single master JobTracker and one slave TaskTracker per cluster node.

The **master** is responsible for scheduling the component tasks in a job onto the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master. The data for a MapReduce task is initially at input files. The input files typically reside in the HDFS. The files may be line-based log files, binary format file, multi-line input records, or something else entirely different.

The MapReduce framework operates entirely on key, value-pairs. The framework views the input to the task as a set of (key, value) pairs and produces a set of (key, value) pairs as the output of the task, possibly of different types.

Map-Tasks

Map task means a task that implements a map(), which runs user application codes for each key-value pair **(kl, vl).** Key **kl** is a set of keys. Key **kl** maps to group of data values (Section 3.3.1). Values **vl** are a large string which is read from the input file(s).

The **output** of map() would be zero (when no values are found) or intermediate key-value pairs **(k2, v2).** The value v2 is the information for the transformation operation at the reduce task using aggregation or other reducing functions.

Reduce task refers to a task which takes the output v2 from the map as an input and combines those data pieces into a smaller set of data using a *combiner*. The reduce task is always performed after the map task.

The ***Mapper*** performs a function on individual values in a dataset irrespective of the data size of the input. That means that the Mapper works on a single data set. Figure 4.4 shows logical view of functioning of map().

7b. Explain Hive integration and work flow steps with neat diagrams

Hive Integration and Workflow Steps

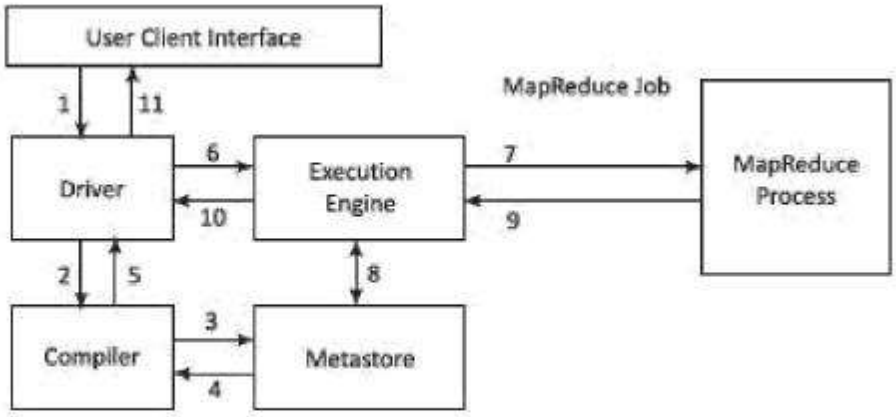


Figure 4.11 Dataflow sequences and workflow steps

The workflow steps are as follows :

Execute Query: Hive interface (CLI or Web Interface) sends a query to Database Driver to execute the query.

Get Plan: Driver sends the query to query compiler that parses the query to check the syntax and query plan or the requirement of the query.

Get Metadata: Compiler sends metadata request to Metastore (of any database, such as MySQL).

Send Metadata: Metastore sends metadata as a response to compiler.

Send Plan: Compiler checks the requirement and resends the plan to driver. The parsing and compiling of the query is complete at this place.

Execute Plan: Driver sends the execute plan to execution engine.

Execute Job: Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Then , the query executes the job.

Metadata Operations: Meanwhile the execution engine can execute the metadata operations with Metastore.

Fetch Result: Execution engine receives the results from Data nodes.

Send Results: Execution engine sends the result to Dr iver.

Send Results: Driver sends the results to Hive Interfaces

8a. Using HIVEQL for the following

1. **Create a table with partition**
2. **Add, rename and drop a partition to a table**

Create a table with partition

If file1 contains client data table:

[php]tab1/clientdata/file1

id, name, dept, yoj
1, sunny, SC, 2009
2, animesh, HR, 2009
3, sumeer, SC, 2010
4, sarthak, TP, 2010[/php]

Now, let us partition above data into two files using years

[php]tab1/clientdata/2009/file2
1, sunny, SC, 2009
2, animesh, HR, 2009

tab1/clientdata/2010/file3
3, sumeer, SC, 2010
4, sarthak, TP, 2010[/php]

Now when we are retrieving the data from the table, only the data of the specified partition will be queried. Creating a partitioned table is as follows:

[php]CREATE TABLE table_tab1 (id INT, name STRING, dept STRING, doj INT)
PARTITIONED BY (year STRING);

ALTER TABLE table_tab1 DROP IF EXISTS PARTITION (year = 2010);
Deleted the partition with year 2010

Alter table table_tab1 ADD PARTITION (year='2016');
Adds a new partition with year 2016

ALTER TABLE table_tab1 PARTITION (year='2010') RENAME TO PARTITION (Year='2016");

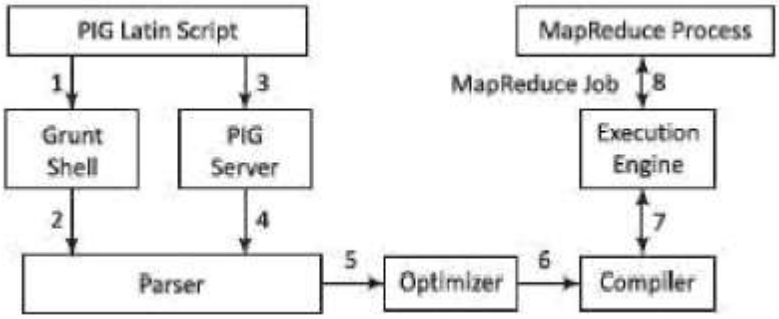
8b. What is PIG in Big Data? Explain the features of PIG

PIG

- ✓ It is an abstract over MapReduce
- ✓ It is a execution frame work 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.

Applications of Apache Pig

- ✓ Analyzing large datasets
- ✓ Executing tasks involving adhoc processing
- ✓ Processing large data sources such as web logs and streaming online data
- ✓ Data processing for search platforms. Pig processes different types of data



1. Grunt Shell: An interactive shell of Pig that executes the scripts.

2. **Script File:** Pig commands written in a script file that execute at Pig Server.

3. **Embedded Script:** Create UDFs for the functions unavailable in Pig builtin operators. UDF can be in other programming languages. The UDFs can embed in Pig Latin Script file.

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).

Acylic 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 dataflows.

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.

The optimizer reduces the amount of data in the pipeline at any instant of time, while processing the extracted data. It executes certain functions for carrying out this task, as explained as follows:

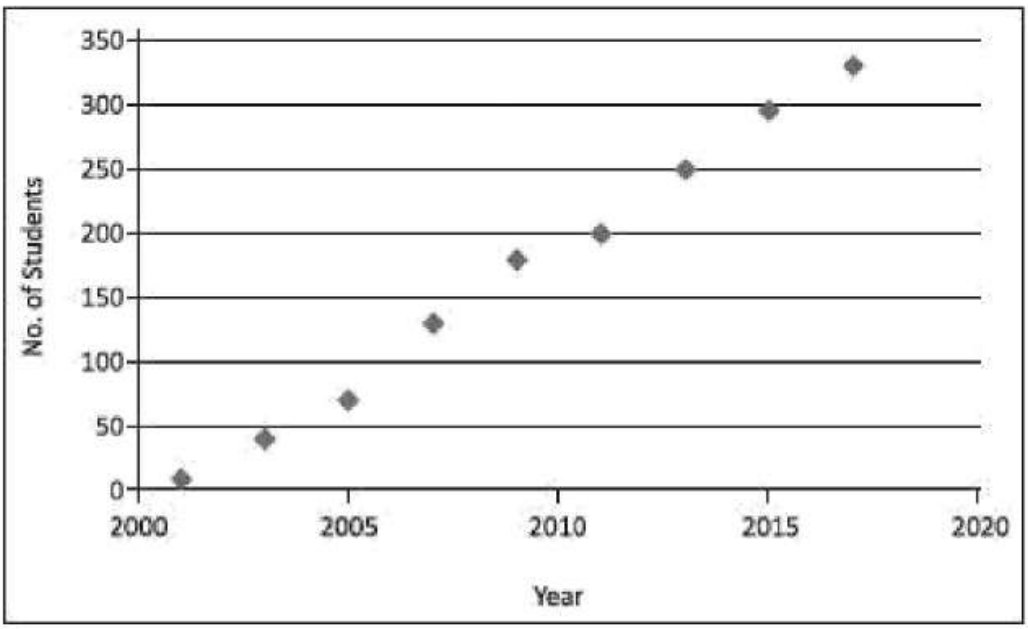
PushUpFilter*:* If there are multiple conditions in the filter and the filter can be split, Pig splits the conditions and pushes up each condition separately. Selecting these conditions at an early stage helps in reducing the number of records remaining in the pipeline.

PushDownf or EachFlatten*:* Applying flatten, which produces a cross product between a complex type such as a tuple, bag or other fields in the record, as late as possible in the plan. This keeps the number of records low in the pipeline.

ColumnPruner*:* Omitts never used columns or the ones no longer needed, reducing the size of the record. This can be applied after each operator, so that the fields can be pruned as aggressively as possible.
MapKeyPruner: Omitts never used map keys, reducing the size of the record.
Limit Optimizer*:* If the limit operator is immediately applied after *load* or *sort* operator, Pig converts the load or sort into a limit-sensitive implementation, which does not require processing the whole dataset. Applying the limit earlier reduces the number of records.
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.

9 a. In machine explain linear and nonlinear relationships with necessary graphs

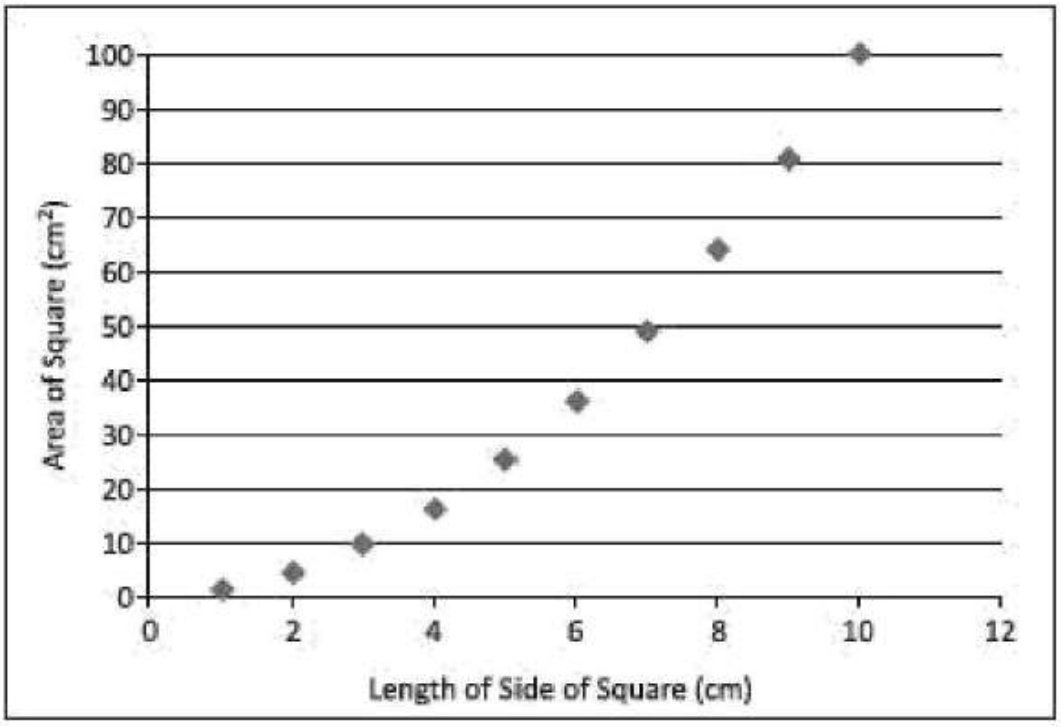
A linear relationship exists between two variables, say x and y, when a straight line (y = a0 + a1.x) can fit on a graph, with at least some reasonable degree of accuracy. The a1 is the linearity coefficient. For example, a scatter chart can suggest a linear relationship, which means a straight line. Figure 6.1 shows a scatter plot, which fits a linear relationship between the number of students opting for computer courses in years between 2000 and 2017.



Scatter plot for linear relationship between students opting for computer courses in years between 2000 and 2017

A linear relationship can be positive or negative. A positive relationship implies if one variable increases in value, the other also increases in value. A negative relationship, on the other hand, implies when one increases in value, the other decreases in value. Perfect, strong or weak linearship categories depend upon the bonding between the two variables.

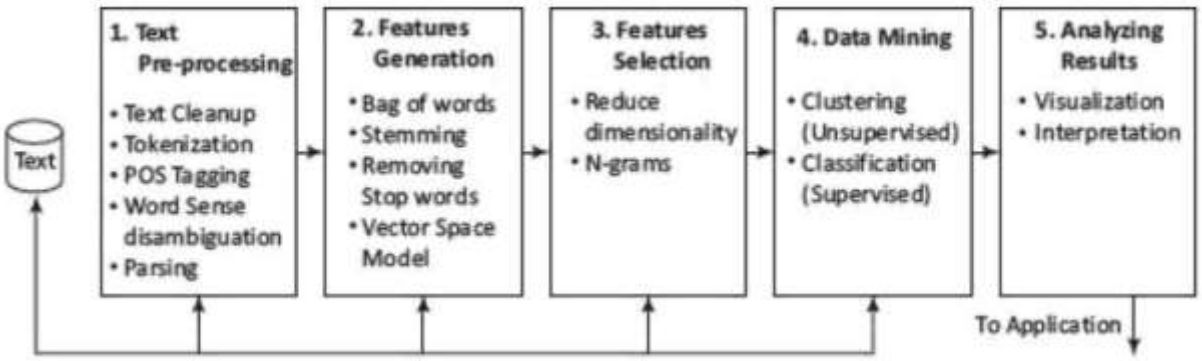
A non-linear relationship is said to exist between two quantitative variables when a curve (y = a0 + a1.x + a2.x2 + •••) can be used to fit the data points. The fit should be with at least some reasonable degree of accuracy for the fitted parameters, a0, a1, a2 ••• Expression for y then generally predicts the values of one quantitative variable from the values of the other quantitative variable with considerably more accuracy than a straight line.
Consider an example of non-linear relationship: The side of a square and its area are not linear. In fact, they have quadratic relationship. If the side of a square doubles, then its area increases four times. The relationship predicts the area from the side.



scatter plot in case of a non-linear relationship between side of square and its area

9b. Write the block diagram of text mining process and explain

Text Mining is a rapidly evolving area of research. As the amount of social media and other text data grows, there is need for efficient abstraction and categorization of meaningful information from the text.



The five phases for processing text are as follows:
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 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 of such 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 pre-processing 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: Analysing results
(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.

10 a. Define multiple regression. Write down the examples involved in forecasting and optimization in regression

A criterion variable can be predicted from one predictor variable in simple linear regression. The criterion can be predicted by two or more variables in multiple regressions.
Multiple regressions are used when two or more independent factors are involved. These regressions are also widely used to make short- to mid-term predictions to assess which factors to include and which to exclude. Multiple regressions can be used to develop alternate models with different factors. More than one variable can be used as a predictor with multiple regressions. However, it is always suggested to use a few variables as predictors necessarily, to get a reasonably accurate forecast. The prediction takes the form:
More than one variable can be used as a predictor with multiple regressions. However, it is always suggested to use a few variables as predictors necessarily, to get a reasonably accurate forecast. The prediction takes the form:
Multiple regression analysis, often referred to simply as regression analysis, examines the effects of multiple independent variables on the value of a dependent variable or outcome.

- Simple linear regression is commonly used in forecasting and financial analysis—for a company to tell how a change in the GDP could affect sales, for example.

Linear regression is a statistical tool used to help predict future values from past values. It is commonly used as a quantitative way to determine the underlying trend and when prices are overextended. A linear regression trendline uses the least squares method to plot a straight line through prices so as to minimize the distances between the prices and the resulting trendline. This linear regression indicator plots the trendline value for each data point.

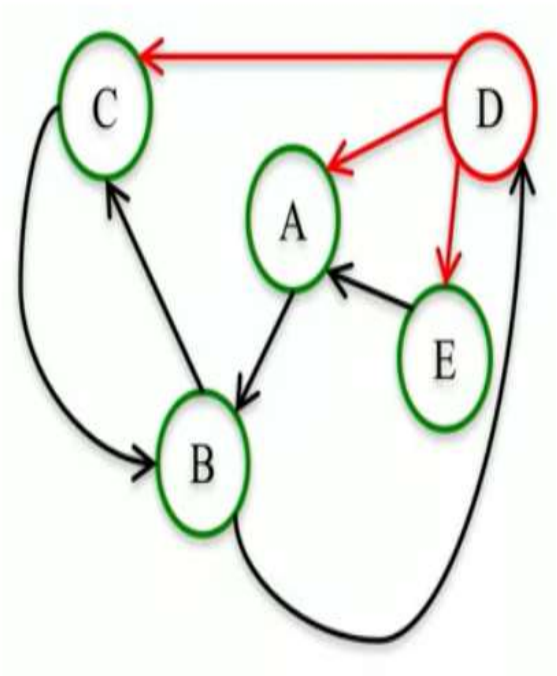
10 b. Explain the parameters in social graph network topological analysis using centralities and page rank

The Social Network Analysis is simply a set of objects, which we call nodes that have some relationships between each other, which we call edges. The first reason to study networks, is because networks are everywhere. In social networks the nodes are people, and the connections between the nodes represent some type relationship between the people in the network. There are many metrics to estimate the importance or better, the centrality of a node in a network such as **Closeness centrality** and **Betweenness centrality**. The page rank was developed by the Google founders when they were thinking about how to measure the importance of webpages using the hyperlink network structure of the web. And the basic idea, is that PageRank will assign a score of importance to every single node. And the assumption that it makes, is that important nodes are those that have many in-links from important pages or important other nodes.

The Page Rank can be also used on any type of network, for example, the web or social networks, but it really works better for networks that have **directed edges**. In fact, the important pages are those that have many **in-links** from more important pages.

At first, we start with every node having a PageRank value of 1/n. And then what we give all of its PageRank to all the nodes that it points to, and then we do this over and over again performing these **Basic PageRank Updating Rules** k times. And then, the new value of PageRank for every node is going to be simply the sum of all the PageRank that are received from all the nodes that point to it.

Page Rank (k = 1)					
	A	B	C	D	E
Old	1/5	1/5	1/5	1/5	1/5
New	4/15	2/5	1/6	1/10	1/15

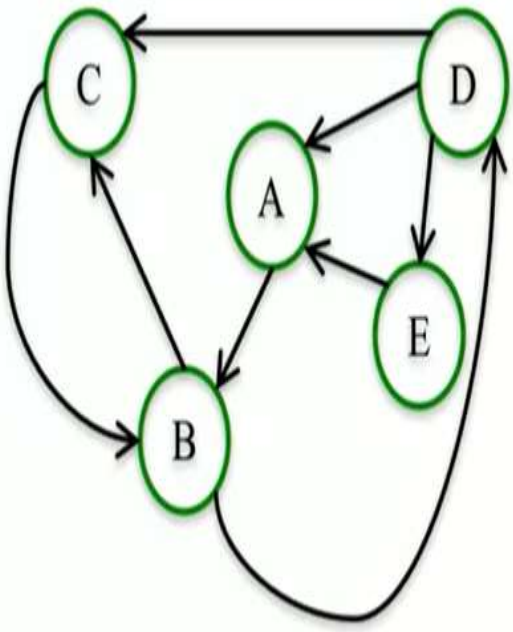


A: (1/3)*(1/5) + 1/5 = 4/15
B: 1/5 + 1/5 = 2/5
C: (1/3)*(1/5) + (1/2)*(1/5) = 5/30 = 1/6
D: (1/2)*(1/5) = 1/10
E: (1/3)*(1/5) = 1/15

At this point, we assign every node a PageRank of 1/n. looking at the figure above, every node will have a PageRank value of 1/5. And now we're going to start applying the Update Rule. So we're going to have to keep track of the old values of PageRank, and what the new value of PageRank of each node is. Let's start with node A. Nodes D and E point to node A, so A is going to get PageRank from D and E. Now let's think about how much **PageRank A** is going to receive from each one of those two nodes. So if we look at **D**, D has three edges, that points to three different nodes, C, A, and E. So A is going to receive 1/3 of the current page rank that D has. D currently has 1/ 5 PageRank, and so A is going to get 1/3 of that 1/5 PageRank that D has. Now A is also going to get PageRank from node **E**, and because E only points to A, then it's going to give all of its PageRank to node A. And so A is going to get 1/5 PageRank from node E. And so, in total, A is going to get 4/15 PageRank from those two nodes. And so the new value PageRank of node A is 4/15. Now, let's think of node B. We use the same approach for each nodes on the network. The figure above represents the **PageRank at Step 1**.

Crucially, we do the exact same thing again to get the second step of PageRank, k equals 2 in order to obtain the **PageRank at Step 2** as shown in the figure below.

	Page Rank				
	A	B	C	D	E
k=2	1/10	13/30	7/30	2/10	1/30
k=2	.1	.43	.23	.20	.03

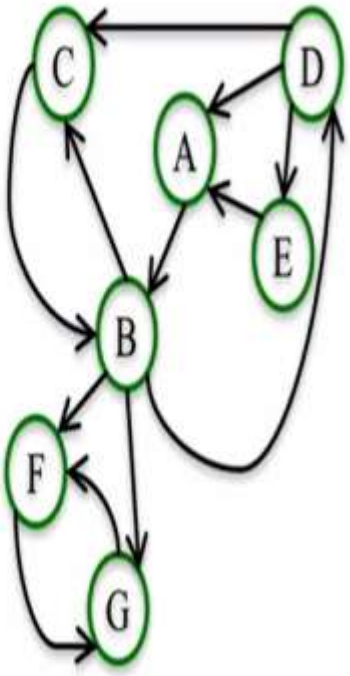


From the figure above, after two steps we find that node B has the highest PageRank (0.43), followed by node C, then node D, node A, and E. This point, this is suggesting that B is the most important node in this network.
If we continue with more steps we might notice that the values change a little bit, but they still have the same order and B is sitll the highest PageRank node.

The PageRank value of a node after k steps can be interpreted as the probability that a random walker lands on that node after taking k random steps. Random walk means we would start on a random node, and then we choose outgoing edges at random, and follow those edges to the next node. And then you're going to repeat this k times. We simply randomly choose edges and walk along in the network and the value of PageRank of each node is the probability that you would land on that node after k steps. If you repeat this for a lot of steps, say k equals infinity. These are the values that you can eventually approach, these are the values that you converges to.

Lets make a small change to this network adding two nodes, F and G, where B points to both of those nodes, and then they point to each other as depicted in the figure below.

Scaled PageRank ($\alpha = .8, k$ large)						
A	B	C	D	E	F	G
.08	.17	.1	.08	.05	.27	.25



We want you to look at this network and try and figure out what the PageRank of each node is after taking a lot of PageRank **steps** for example **4k**. For large enough k, F and G are going to have a PageRank value of about one half. And all the other nodes are going to have a PageRank value of 0. That is due to the fact that whenever the random walk lands on F or G, then they're going to stock on F and G because there are no edges to go to. There is no way to get back from G and F to any of the other notes. The way we fix this problem is by introducing a new parameter to the PageRank computation called this **damping** parameter **alpha** and we make the random woalk with the with probability **1- alpha** and by doing this we get unstuck whenever we actually choose a random node. Crucially, for most networks as k gets larger, the **Scaled PageRank** converges to a unique value. Using Scaled PageRank now we have F and G still with a very high PageRank compared to the other notes. But the other nodes don't have a PageRank value of zero. It is important to note that this damping parameter works better for **large network**s like the **web** or very large **social networks**.