22MCA412 – Big Data Analytics

Fourth Semester MCA Degree Examination, June / July 2025

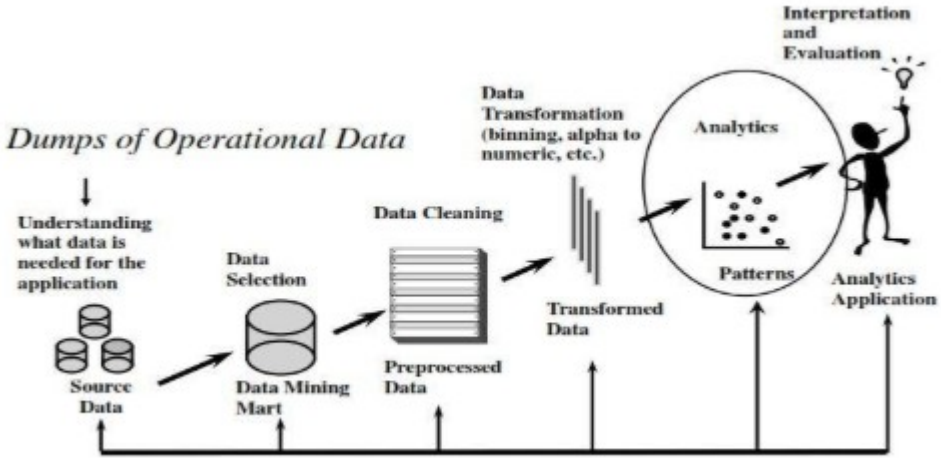
Max. Marks: 100 Time: 3 Hrs

Module – 1

Q.1 a. Explain the various steps involved in an Analytical Process model with a neat diagram. (10 marks)

1. Define the business problems to be solved using analytics.
2. All source-data need to be identified that could be of potential interest. (Select of data will have a deterministic impact on the analytical model).
3. All data will be gathered in a staging area which could be data mart/ data warehouse. Basic exploratory analysis will be considered, for example: OLAP (Online Analytical Processing) facilities for multi-dimensional data analysis.
4. Data Cleaning steps to get rid of all inconsistencies – like missing data / values, outliers and duplicate data. Additional transformations may also be considered, such as binning, alphanumeric to numeric coding, geographical aggregation etc.
5. In the analytics steps, an analytical model will be estimated on the pre-processed and transformed data. Once the model is built, it will be interpreted and evaluated by the business experts. Many trivial patterns will be detected by the model.

Knowledge Pattern: Unexpected yet interesting and actionable patterns (referred to as knowledge pattern). Once analytical model has been appropriately validated and approved, it can be put into production as an analytical application.



b. Define Big Data Analytics. Discuss the different types of data elements with example for each. (10 marks)

Big data is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software.

It is important to appropriately consider the different types of data elements at the start of the analysis. The different types of data elements can be considered:

- Continuous
- Categorical

Continuous: These are data elements that are defined on an interval that can be limited or unlimited. Examples include income, sales, RFM (recency, frequency, monetary).

Categorical: The categorical data elements are differentiated as follows:

Nominal: These are data elements that can only take on a limited let of values with no meaningful ordering in between. Examples: marital status, profession, purpose of loan.

Ordinal: These are data elements that can only take on a limited set of values with a meaningful ordering in between. Examples: credit rating; age coded as young, middle aged, and old.

Binary: These are data elements that can only take on two values. Example: gender, employment status.

Appropriately distinguishing between these different data elements is of key importance to start the analysis when importing the data into an analytics tool. For example, if marital status were to be incorrectly specified as a continuous data element, then the software would calculate its mean, standard deviation, and so on, this is obviously meaningless.

Q.2 a. Discuss the applications of Big Data Analytics. (10 marks)

Analytics is everywhere and strongly embedded in our daily lives. The relevance, importance and impact of analytics are now bigger than ever before and, given that more and more data are being collected and that there is strategic value in knowing what is hidden in data, analytics will continue to grow. Physical mail box: a catalogue sent to us through mail most probably as a result of a response modeling analytical exercise that indicated, given my characteristics and previous purchase behavior, we are likely to buy one or more products from it.

Behavioral Scoring Model: Checking account balance of the customer from the past 12 months and credit payments during that period, together with other kinds of information available to the bank, to predict whether a customer will default on the loan during the next

year. Social Media: As we logged on to my Facebook page, the social ads appearing there were based on analyzing all information (posts, pictures, my friends and their behavior, etc.) available to Facebook. Twitter posts will be analyzed (possibly in real time) by social media analytics to understand both the subject tweets and the sentiment of them.

Marketing	Risk Management	Government	Web	Logistics	Other
Response modeling	Credit risk modeling	Tax avoidance	Web analytics	Demand forecasting	Text analytics
Net lift modeling	Market risk modeling	Social security fraud	Social media analytics	Supply chain analytics	Business process analytics
Retention modeling	Operational risk modeling	Money laundering	Multivariate testing		
Market basket analysis	Fraud detection	Terrorism detection			
Recommender systems					
Customer segmentation					

b. Analyze the given data to detect any outliers, if present in the following data, where mean = 40, standard deviation = 10. Data: 30, 50, 10, 40, 60, 80. (10 marks)

Data Value	Z-Score Calculation	Z-Score	Outlier Status
30	$(30 - 40) \div 10 = -10 \div 10$	−1	Not an outlier
50	$(50 - 40) \div 10 = 10 \div 10$	+1	Not an outlier
10	$(10 - 40) \div 10 = -30 \div 10$	−3	Borderline Outlier
40	$(40 - 40) \div 10 = 0 \div 10$	0	Not an outlier
60	$(60 - 40) \div 10 = 20 \div 10$	+2	Not an outlier
80	$(80 - 40) \div 10 = 40 \div 10$	+4	Clear Outlier

Module – 2

Q.3 a. Write a note on Mobile Business Intelligence and Big Data. (10 marks)

Analytics on mobile devices is what some refer to as putting BI in your pocket. Mobile drives straight to the heart of simplicity and ease of use that has been a major barrier to BI adoption since day one. Kerzner explains his view on this topic: They have been working on Mobile BI for a while but the iPad was the inflection point where I think it started to become mainstream. I have seen customers over the past decade who focused on the mobile space generally and mobile applications in particular. One client in particular told me that he felt like he was pushing a boulder up a hill until he introduced mobility to enhance productivity. Once the new smart phones and tablets arrived, his phone was ringing off the

hook and he was trying to figure out which project to say yes to, because he couldn't say yes to everyone who suddenly wanted mobile analytics in the enterprise.

Ease of Mobile Application Deployment

Three elements that have impacted the viability of mobile BI:

Location—the GPS component and location . . . know where you are in time as well as the movement.
It's not just about pushing data; you can transact with your smart phone based on information you get.
Multimedia functionality allows the visualization pieces to really come into play.

Three challenges with mobile BI include:

1. Managing standards for rolling out these devices.
2. Managing security (always a big challenge).
3. Managing "bring your own device," where you have devices both owned by the company and devices owned by the individual, both contributing to productivity.

b. Describe the Inter and Trans firewall analytics with the help of a neat diagram. (10 marks)

Over the last 100 years, supply chain has evolved to connect multiple companies and enable them to collaborate to create enormous value to the end-consumer through concepts like CPFR (collaborative planning, forecasting and replenishment—a collection of business practices that leverage the Internet and electronic data interchange to reduce inventories and expenses while improving customer service), VMI (vendor-managed inventory—a technique used by customers in which manufacturers receive sales data to forecast consumer demand more accurately), etc.

Decision sciences will witness a similar trend as enterprises begin to collaborate on insights across the value chain. For instance, in the healthcare industry, rich consumer insights can be generated by collaborating on data and insights from the health insurance provider, pharmacy delivering the drugs and the drug manufacturer. In fact, this is not necessarily limited to companies within the traditional demand-supply chain.

There are instances where a retailer and a social media company can come together to share insights on consumer behaviour that will benefit both concerns. Some of the more progressive companies will take this a step further and work on leveraging the large volumes of data outside the firewall such as social data, location data, etc.

In other words, it will not be long before internal data and insights from within the enterprise firewall is no longer a differentiator. We call this trend the move from intra- to inter- and trans-firewall analytics. Yesterday, companies were doing functional silo-based analytics. Today they are doing intra- firewall analytics with data within the firewall. Tomorrow they

will be collaborating on insights with other companies to do inter-firewall analytics as well as leveraging the public domain to do trans- firewall analytics.

Figure given below depicts, setting up inter- and trans-firewall analytics will add significant value. However, it does present some challenges. They are: As one moves outside the firewall, the information-to-noise ratio increases putting additional requirements on analytical methods and technology requirements Organizations are often limited by a fear of collaboration and overreliance on proprietary information The fear of collaboration is mostly driven by competitive fears, data concerns, and proprietary orientation that limits opportunities for cross-organizational learning and innovation.While it is clear that the transition to an inter- and trans-firewall paradigm is not easy, we feel it will continue to grow and at some point it will become a key weapon, available for decision scientists to drive value and efficiencies.

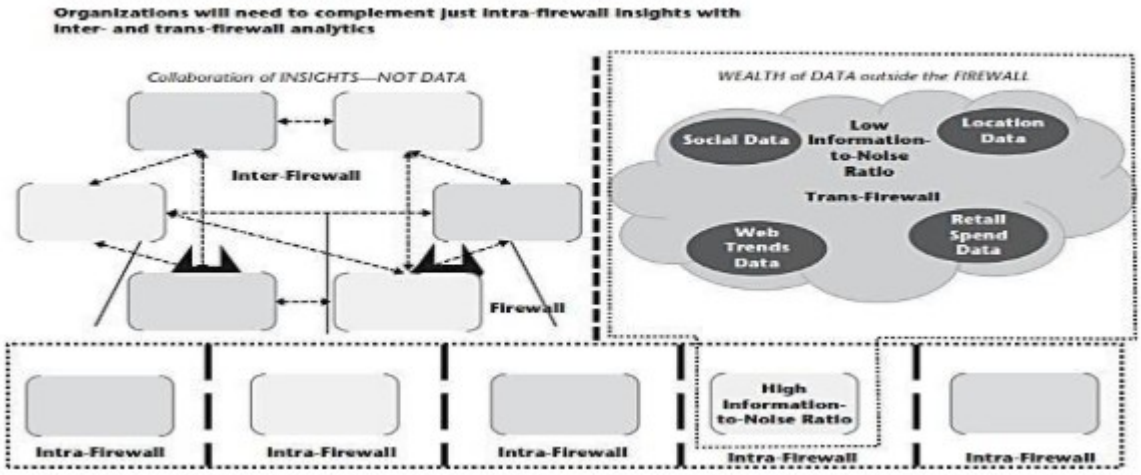


Figure: Value Chain for Inter-Firewall and Trans-Firewall Analytics

OR

Q.4 a. What is Predictive Analysis? Why are they required? Discuss the leading trends of Predictive Analysis. (10 marks)

To master analytics, enterprises will move from being in reactive positions (business intelligence) to forward leaning positions (predictive analytics). Using all the data available- traditional internal data sources combined with new rich external data sources- will make the predictions more accurate and meaningful. Because analytics are contextual, enterprises can build confidence in the analytics and the trust will result in using analytics insight to trigger business events.

Leading trends that are making their way to the forefront of businesses today:

- Recommendation engines similar to those used in Netflix and Amazon that use past purchases and buying behavior to recommend new purchases.
- Risk engines for a wide variety of business areas, including market and credit risk, catastrophic risk, and portfolio risk.
- Innovation engines for new product innovation, drug discovery, and consumer and fashion trends to predict potential new product formulations and discoveries.
- Customer insight engines that integrate a wide variety of customer- related info, including sentiment, behavior, and even emotions. Customer insight engines will be the backbone in online and set-top box advertisement targeting, customer loyalty programs to maximize customer lifetime value, optimizing marketing campaigns for revenue lift, and targeting individuals or companies at the right time to maximize their spend.
- Optimization engines that optimize complex interrelated operations and decisions that are too overwhelming for people to systematically handle at scales, such as when, where, and how to seek natural resources to maximize output while reducing operational costs or what potential competitive strategies should be used in a global business that takes into account the various political, economic, and competitive pressures along with both internal and external operational capabilities.

b. Define Crowd Sourcing. Explain the different types of crowd sourcing. (10 marks)
Crowdsourcing is a great way to capitalize on the resources that can build algorithms and predictive models

Kaggle: Kaggle describes itself as "an innovative solution for statistical/analytics outsourcing." That's a very formal way of saying that Kaggle manages competitions among the world's best data scientists. Here's how it works: Corporations, governments, and research laboratories are confronted with complex statistical challenges. They describe the problems to Kaggle and provide data sets. Kaggle converts the problems and the data into contests that are posted on its web site. The contests feature cash prizes ranging in value from $100 to $3 million. Kaggle's clients range in size from tiny start-ups to multinational corporations such as Ford Motor Company and government agencies such as NASA.

As per Anthony Goldbloom, Kaggle's founder and CEO: The idea is that someone comes to us with a problem, we put it up on our website, and then people from all over the world can compete to see who can produce the best solution." Kaggle's approach is that it is truly a win-win scenario—contestants get access to real-world data (that has been carefully "anonymized" to eliminate privacy concerns) and prize sponsors reap the benefits of the contestants' creativity.

Crowdsourcing is a disruptive business model whose roots are in technology but is extending beyond technology to other areas. There are various types of crowd sourcing, such as crowd voting, crowd purchasing, wisdom of crowds, crowd funding, and contests.

Take for example:

99designs.com/, which does crowdsourcing of graphic design agentanything.com/, which posts "missions" where agents vie for to run errands

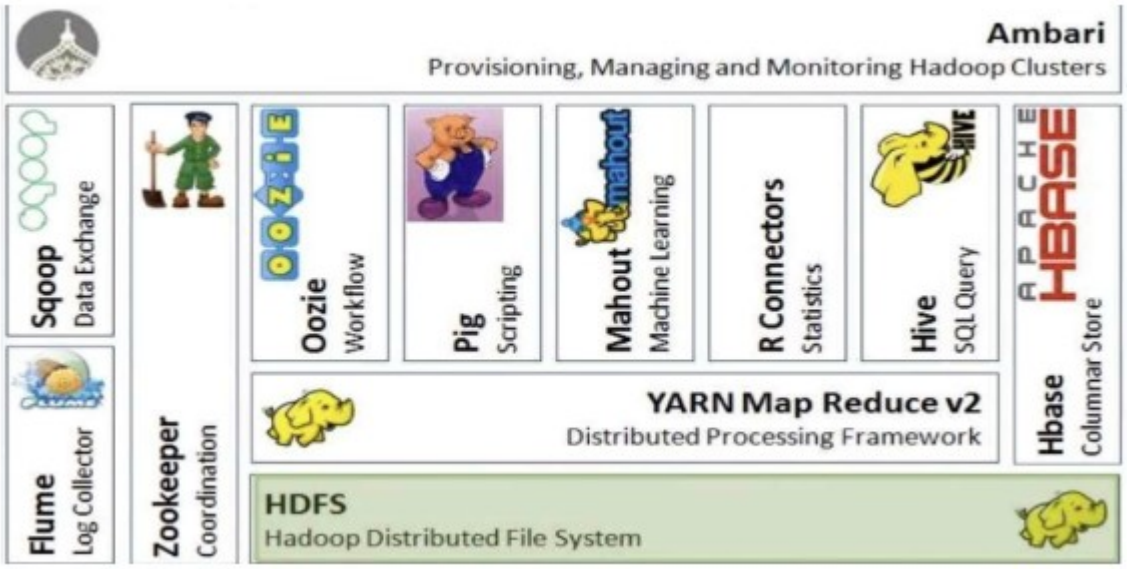
33needs.com/, which allows people to contribute to charitable programs that make a social impact

Module – 3

Q.5a. Explain the working of Hadoop ecosystem with a neat diagram. (10 marks)

Hadoop Ecosystem is neither a programming language nor a service, it is a platform or framework which solves big data problems. You can consider it as a suite which encompasses a number of services (ingesting, storing, analyzing and maintaining) inside it.

The figure below describes the typical view of Hadoop Ecosystem:



The components of Hadoop 1.0 and Hadoop 2.0 are listed as follows:

Common – a set of components and interfaces for filesystems and I/O.

Avro – a serialization system for RPC and persistent data storage.

MapReduce – a distributed data processing model.

YARN - Yet Another Resource Negotiator

HDFS – a distributed filesystem running on large clusters of machines.

Pig – a data flow language and execution environment for large datasets.

Hive – a distributed data warehouse providing SQL-like query language.

HBase – a distributed, column-oriented database.

Mahout, Spark MLlib - Machine Learning

Apache Drill -SQL on Hadoop

ZooKeeper – a distributed, highly available coordination service.

Sqoop – a tool for efficiently moving data between relational DB and HDFS.

Oozie - Job Scheduling

Flume- Data Ingesting Services

Solr & Lucene - Searching & Indexing

Ambari - Provision, Monitor and Maintain cluster

b. Illustrate the disabilities of implementing the storage and analysis support for Big Data. (10 marks)

Storage Challenges

- Volume Overload – Traditional storage systems cannot efficiently handle petabytes/exabytes of data.
- Scalability Issues – Scaling storage infrastructure (servers, disks, cloud) is costly and complex.
- Data Variety Handling – Storing structured, semi-structured, and unstructured data together is challenging.
- Data Redundancy – Replication for fault tolerance consumes extra space.
- Latency – Real-time data storage (e.g., IoT streams) may not sync fast enough.

Analysis Challenges

- High Computational Demand – Big Data analytics require massive processing power, which increases cost.
- Complexity of Algorithms – Running ML/AI models at scale over huge datasets is non-trivial.
- Data Integration – Combining heterogeneous sources (logs, videos, transactions) into one analytical pipeline is tough.
- Real-time Analytics Difficulty – Continuous streaming analysis (e.g., fraud detection, clickstreams) is resource-intensive.
- Accuracy and Noise – Large datasets often have missing, inconsistent, or noisy values, lowering analysis quality.

Cost & Resource Limitations

- Infrastructure Cost – Servers, clusters, and cloud usage bills are very high.
- Skilled Manpower – Requires data engineers, analysts, and domain experts—hard to hire and retain.
- Maintenance Overhead – Monitoring distributed systems adds complexity.
- Security and Privacy Concerns
- Data Security Risks – Sensitive information spread across multiple storage nodes.
- Access Control Difficulty – Managing permissions for distributed systems is complex.
- Regulatory Compliance – Meeting GDPR, HIPAA, etc., is harder with massive, diverse datasets.

OR

Q.6 a. List the differences between RDBMS and Map Reduce. (10 marks)

#	MapReduce	RDBMS
1	Good fit for problems that analyze the whole data set in a batch fashion	RDBMS is good for point queries or updates, where the data set has been ordered to deliver low latency retrieval
2	It suits well for applications where the data is written once and read many times	Relational database is good for data that are continuously updated
3	Works on semi-structured or unstructured data	Operates on structured data
4	Ex: spreadsheets, images, text etc..	Ex: database tables, XML docs etc..
5	It is designed to interpret the data at processing time (Schema on Read)	It is designed to interpret the data run time (Schema on Write)
6	Normalization creates a problem in Hadoop because, it reading a record is non-local operation, instead Hadoop makes it possible to perform streaming reads and writes	RDBMS data is often normalized to avoid redundancy and to retain integrity.
7	MapReduce can process the data in parallel	Parallel processing is not true for SQL RDBMS queries

b. Write short notes on: (10 marks)

i) Volunteer Computing projects work by breaking the problem they are trying to solve into small chunks called work units, which are sent to computers around the world to analyse.

Example: SETI@home: It works as follows:

- It sends a work unit of 0.35 MB of radio telescope data and takes hours or days to analyse on a typical home computer. When the analysis is completed, results are sent back to the server.
- SETI, the Search for Extra-Terrestrial Intelligence, runs a project called SETI@home in which volunteers donate CPU time from their otherwise idle computers to analyze radio telescope data for signs of intelligent life outside earth.
- others include the Great Internet Mersenne Prime Search (to search for large prime numbers)
- and Folding@home (to understand protein folding and how it relates to disease).
- Volunteer computing projects work by breaking the problem they are trying to solve into chunks called *work units*, which are sent to computers around the world to be analyzed.
- For example, a SETI@home work unit is about 0.35 MB of radio telescope data, and takes hours or days to analyze on a typical home computer.
- When the analysis is completed, the results are sent back to the server, and the client gets another work unit.
- As a precaution to combat cheating, each work unit is sent to three different machines and needs at least two results to agree to be accepted.

ii) Grid Computing

Grid computing is the collection of **computer** resources from multiple locations to reach a common goal. The **grid** can be thought of as a distributed system with non-interactive workloads that involve a large number of files.
In **grid computing**, the **computers** on the network can work on a task together, thus functioning as a supercomputer.

- The High Performance Computing (HPC) and Grid Computing communities have been doing large- scale data processing for years, using such APIs as Message Passing Interface (MPI).
- HPC is to distribute the work across a cluster of machines, which access a shared filesystem, hosted by a SAN.
- HPC works well for predominantly compute-intensive jobs, but becomes a problem when nodes need to access larger data volumes (hundreds of gigabytes) since the network bandwidth is the bottleneck and compute nodes become idle.
- MapReduce tries to collocate the data with the compute node, so data access is fast since it is local. *Data locality*, is at the heart of MapReduce and is the reason for its good performance. Recognizing that network bandwidth is the most precious resource in a data center environment (it is easy to saturate network links by copying data around), MapReduce implementations go to great lengths to conserve it by explicitly modelling network topology.
- MPI gives great control to the programmer, but requires that he or she explicitly handle the mechanics of the data flow.

- MapReduce operates only at the higher level: the programmer thinks in terms of functions of key and value pairs, and the data flow is implicit.

Module – 4

Q.7 a. What is a memory block in HDFS? Explain block report, replication factor and rack awareness with respect to data node. (10 marks)

A disk has a block size, which is the minimum amount of data that it can read or write. Filesystems for a single disk build on this by dealing with data in blocks, which are an integral multiple of the disk block size. Filesystem blocks are typically a few kilobytes in size, while disk blocks are normally 512 bytes.

HDFS too has the concept of a block, but it is a much larger unit of 64 MB or 128MB by default. Like in a filesystem for a single disk, files in HDFS are broken into block-sized chunks, which are stored as independent units. Unlike a filesystem for a single disk, a file in HDFS that is smaller than a single block does not occupy a full blocks worth of underlying storage. There are tools to perform filesystem maintenance, such as *df* and *fsck*, that operate on the filesystem block level.

Benefits of using block abstraction

A file can be larger than any single disk in the network: There's nothing that requires the blocks from a file to be stored on the same disk, so they can take advantage of any of the disks in the cluster. In fact, it would be possible, if unusual, to store a single file on an HDFS cluster whose blocks filled all the disks in the cluster. Making the unit of abstraction a block rather than a file simplifies the storage subsystem: Simplicity is something to strive for all in all systems, but is especially important for a distributed system in which the failure modes are so varied.

Blocks are just a chunk of data to be stored—file metadata such as permissions information does not need to be stored with the blocks, so another system can handle metadata separately. Blocks fit well with replication for providing fault tolerance and availability. To insure against corrupted blocks and disk and machine failure, each block is replicated to a small number of physically separate machines (typically three). If a block becomes unavailable, a copy can be read from another location in a way that is transparent to the client.

Definition of Replication Factor: The replication factor is a property that can be set in the HDFS configuration file that will allow you to adjust the global replication factor for the entire cluster. A block that is no longer available due to corruption or machine failure can be replicated from its alternative locations to other live machines to bring the replication factor back to the normal level. Similarly, some applications may choose to set a high replication factor for the blocks in a popular file to spread the read load on the cluster. Like its disk filesystem cousin, HDFS's fsck command understands blocks. For example, running:

% hadoop fsck / -files –blocks will list the blocks that make up each file in the filesystem.

In Hadoop, most of the components like NameNode, DataNode etc are rack-aware. It **means** they have the information about the rack on which they exist. The main use of **rack awareness** is in implementing fault-tolerance. Hadoop components are rack-aware. For example, HDFS block placement will use rack awareness for fault tolerance by placing one block replica on a different rack. This provides data availability in the event of a network switch failure or partition within the cluster.

b. Discuss any five HDFS commands. (10 marks)

%hadoop fs -help <<command>>

– to get detailed help on every command.

% hadoop fs -copyFromLocal input/docs/quangle.txt hdfs://localhost/user/tom/quangle.txt

% hadoop fs -copyFromLocal input/docs/quangle.txt /user/tom/quangle.txt

% hadoop fs -copyFromLocal input/docs/quangle.txt quangle.txt

- Start by copying a file from the local filesystem to HDFS

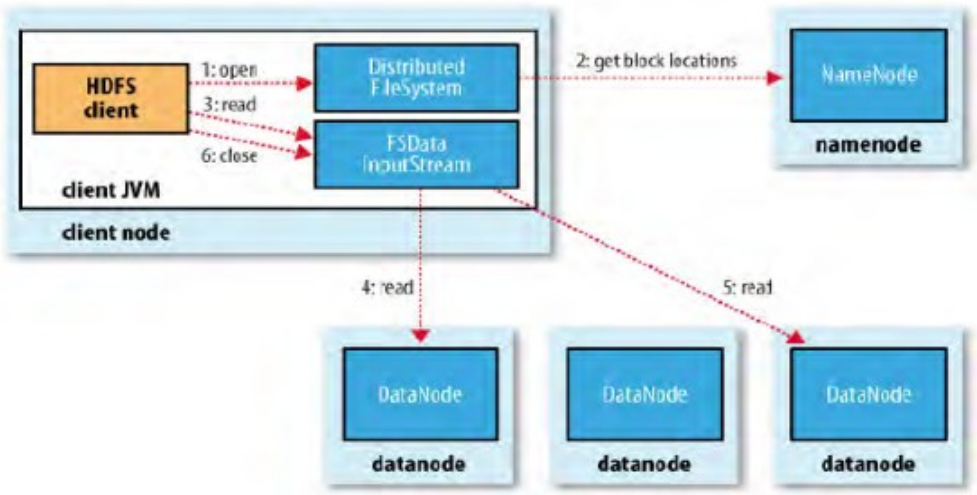
% Hadoop fs -copyToLocal hdfs://Buangle.txt c:/input/docs/Buangle.txt

- Start by copying a file from the HDFS to local filesystem

OR

Q.8 a. With a neat diagram, explain the anatomy of reading data from a file in HDFS. (10 marks)

To get an idea of how data flows between the client interacting with HDFS, the name-node and the datanodes, consider the below Figure, which shows the main sequence of events when reading a file.



The client opens the file it wishes to read by calling open() on the FileSystem object, which for HDFS is an instance of DistributedFileSystem (step 1 in Figure). DistributedFileSystem calls the namenode, using RPC, to determine the locations of the blocks for the first few blocks in the file (step 2). For each block, the namenode returns the addresses of the datanodes that have a copy of that block. The DistributedFileSystem returns an FSDataInputStream (an input stream that supports file seeks) to the client for it to read data from. FSDataInputStream in turn wraps a DFSInputStream, which manages the datanode and namenode I/O.

The client then calls read() on the stream (step 3). DFSInputStream, which has stored the datanode addresses for the first few blocks in the file, then connects to the first (closest) datanode for the first block in the file. Data is streamed from the datanode back to the client, which calls read() repeatedly on the stream (step 4). When the end of the block is reached, DFSInputStream will close the connection to the datanode, then find the best datanode for the next block (step 5). This happens transparently to the client, which from its point of view is just reading a continuous stream.

Blocks are read in order with the DFSInputStream opening new connections to datanodes as the client reads through the stream. It will also call the namenode to retrieve the datanode locations for the next batch of blocks as needed. When the client has finished reading, it calls close() on the FSDataInputStream (step 6).During reading, if the DFSInputStream encounters an error while communicating with a datanode, then it will try the next closest one for that block. It will also remember datanodes that have failed so that it doesn't needlessly retry them for later blocks. The DFSInputStream also verifies checksums for the data transferred to it from the datanode. If a corrupted block is found, it is reported to the namenode before the DFSInput Stream attempts to read a replica of the block from another datanode.

One important aspect of this design is that the client contacts datanodes directly to retrieve data and is guided by the namenode to the best datanode for each block. This design allows HDFS to scale to a large number of concurrent clients, since the data traffic is spread across all the datanodes in the cluster. The namenode meanwhile merely has to service block location requests

(which it stores in memory, making them very efficient) and does not, for example, serve data, which would quickly become a bottleneck as the number of clients grew.

b. Explain the architecture changes that are needed while active name node with standby name node. (10 marks)

The architectural changes are needed to replace active name node with standby name node:

i. The namenodes must use highly-available shared storage to share the edit log. When a standby namenode comes up it reads up to the end of the shared edit log to synchronize its state with the active namenode, and then continues to read new entries as they are written by the active namenode.

ii. Datanodes must send block reports to both namenodes since the block mappings are stored in a namenode's memory, and not on disk.Clients must be configured to handle namenode failover, which uses a mechanism that is transparent to users.

iii. If the active namenode fails, then the standby can take over very quickly (in a few tens of seconds) since it has the latest state available in memory: both the latest edit log entries, and an up-to-date block mapping. The actual observed failover time will be longer in practice (around a minute or so), since the system needs to be conservative in deciding that the active namenode has failed.

iv. In the unlikely event of the standby being down when the active fails, the administrator can still start the standby from cold. This is no worse than the non- HA case, and from an operational point of view it's an improvement, since the process is a standard operational procedure built into Hadoop.

Module – 5

Q.9a. Explain the following: (10 marks)

i) Mapper class

The **Mapper class** in Hadoop is responsible for the first stage of the MapReduce framework. Its primary function is to take the raw input data, process it, and generate intermediate key–value pairs. Each Mapper works on a specific portion of the input data, known as an input split, and processes it independently in parallel across multiple nodes. This parallel execution significantly improves the speed and scalability of big data processing. For example, in a word count program, the Mapper reads lines of text, splits them into words, and for each word, it emits a key–value pair such as (word, 1). Thus, the Mapper class essentially performs the tasks of filtering, parsing, and transforming raw data into a structured form that can be aggregated later.

ii) Reducer class

The **Reducer class**, on the other hand, is responsible for the second stage of the MapReduce framework. It takes the intermediate key–value pairs produced by the Mappers, groups them by key, and applies aggregation operations to generate the final result. During this stage, all values associated with a particular key are collected into a list, and the Reducer processes this list to produce a single output for each key. For instance, continuing with the word count example, the

Reducer takes input in the form of (word, [1,1,1...]) and sums the values to produce a final output such as (word, count). The Reducer class is crucial for summarizing, filtering, and combining data to produce meaningful insights from large volumes of raw input.

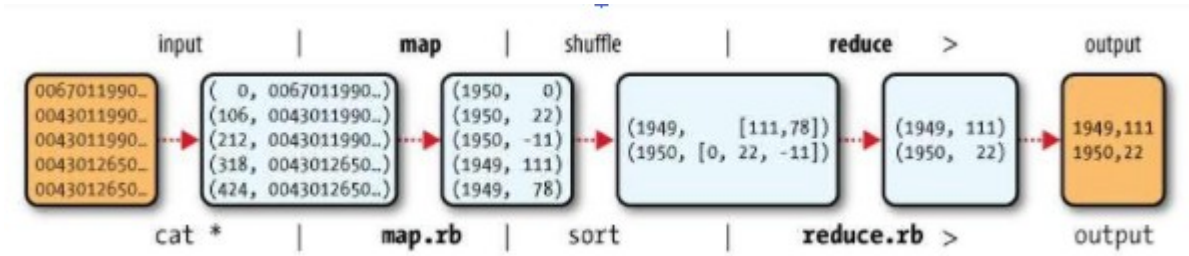
b. Describe Map Reduce Execution steps with a neat diagram and Replication factor. (10 marks)

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

- The Map task takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key-value pairs).
- The Reduce task takes the output from the Map as an input and combines those data tuples (key- value pairs) into a smaller set of tuples.

The reduce task is always performed after the map job.

Let us now take a close look at each of the phases and try to understand their significance.



Input Phase − Here we have a Record Reader that translates each record in an input file and sends the parsed data to the mapper in the form of key-value pairs.

Map − Map is a user-defined function, which takes a series of key-value pairs and processes each one of them to generate zero or more key-value pairs.

Intermediate Keys − They key-value pairs generated by the mapper are known as intermediate keys.

Combiner − A combiner is a type of local Reducer that groups similar data from the map phase into identifiable sets. It takes the intermediate keys from the mapper as input and applies a user-defined code to aggregate the values in a small scope of one mapper. It is not a part of the main MapReduce algorithm; it is optional.

Shuffle and Sort − The Reducer task starts with the Shuffle and Sort step. It downloads the grouped key-value pairs onto the local machine, where the Reducer is running. The individual key-value pairs are sorted by key into a larger data list. The data list groups the equivalent keys together so that their values can be iterated easily in the Reducer task.

Reducer − The Reducer takes the grouped key-value paired data as input and runs a Reducer function on each one of them. Here, the data can be aggregated, filtered, and combined in a number of ways, and it requires a wide range of processing. Once the execution is over, it gives zero or more key-value pairs to the final step.

Output Phase − In the output phase, we have an output formatter that translates the final key-value pairs from the Reducer function and writes them onto a file using a record

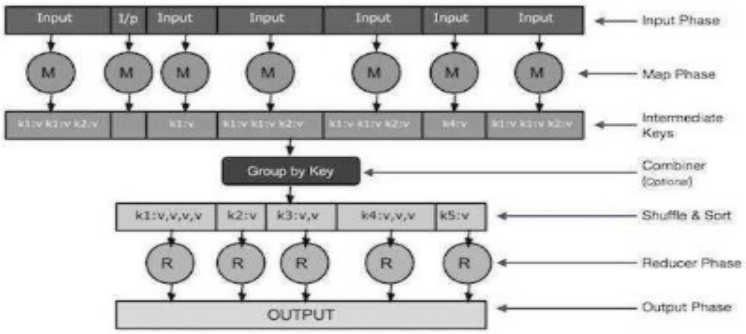
OR

Q.10 a. How Map Reduce job works with classic Java stream. (10 marks)

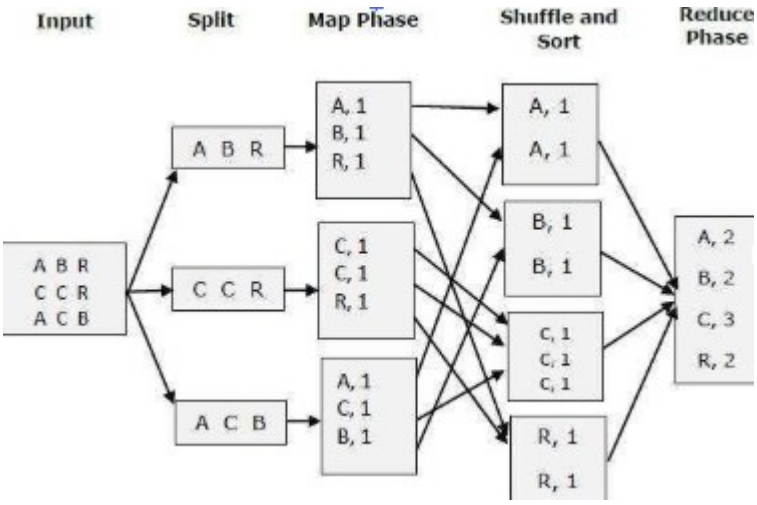
A MapReduce job works in a sequence of well-defined stages to process large-scale data in a distributed manner. First, the input data is divided into smaller chunks called splits, and each split is processed independently by a Mapper. The Mapper transforms the raw input into intermediate key–value pairs, such as emitting (word, 1) for each word encountered in a text file. Once all Mappers finish their work, the framework automatically performs a shuffle and sort operation, grouping values that share the same key. These grouped results are then sent to the Reducer, which aggregates the values for each key to produce the final output, such as summing the counts of words in a word count program. Finally, the output is written back to storage, typically in HDFS. This model is powerful because it distributes computation across many nodes, allowing it to process massive datasets efficiently.

In classic Java Streams, the concepts of map and reduce are implemented in a simpler, in-memory fashion. The map() function in Java Streams transforms each element of a collection into another form, much like the Mapper converts input records into intermediate results. Similarly, the reduce() function in Java Streams combines elements of a stream into a single result, just as the Reducer aggregates values for each key in Hadoop. The key difference lies in scale and execution: Java Streams operate within a single JVM, optionally using multiple threads for parallelism, whereas MapReduce distributes data and computation across a cluster of machines.

b. With a neat diagram, explain Map Reduce programming model. (10 marks)



Let us try to understand the two tasks Map &f Reduce with the help of a small diagram – Word Count Example.



The input to our map phase is the raw NCDC data. We choose a text input format that gives us each line in the dataset as a text value. The key is the offset of the beginning of the line from the beginning of the file, but as we have no need for this, we ignore it. Our map function is simple. We pull out the year and the air temperature, since these are the only fields we are interested in. In this case, the map function is just a data preparation phase, setting up the data in such a way that the reducer function can do its work on it: finding the maximum temperature for each year.

To visualize the way the map works, consider the following sample lines of input data

```
0067011990999991950051507004...9999999N9+00001+99999999999...
0043011990999991950051512004...9999999N9+00221+99999999999...
0043011990999991950051518004...9999999N9-00111+99999999999...
0043012650999991949032412004...0500001N9+01111+99999999999...
0043012650999991949032418004...0500001N9+00781+99999999999...
```

These lines are presented to the map function as the key-value pairs:

```
(0, 0067011990999991950051507004...9999999N9+00001+99999999999...)
(106, 0043011990999991950051512004...9999999N9+00221+99999999999...)
(212, 0043011990999991950051518004...9999999N9-00111+99999999999...)
(318, 0043012650999991949032412004...0500001N9+01111+99999999999...)
(424, 0043012650999991949032418004...0500001N9+00781+99999999999...)
```

These lines are presented to the map function as the key-value pairs:The keys are the line offsets within the file, which we ignore in our map function. The map function merely extracts the year and the air temperature (indicated in bold text), and emits them as its output (the temperature values have been interpreted as integers): The output from the map function is processed by the MapReduce framework before being sent to the reduce function. This processing sorts and groups the key-value pairs by key. So, continuing the example, our reduce function sees the following input: Each year appears with a list of all its air temperature readings. All the reduce function has to do now is iterate through the list and pick up the maximum reading: This is the final output: the maximum global temperature recorded in each year.