Page No.:
Date:

4) Big Data
5) Sources of Big Data.

6)

Unit	Value	Example
Kilobytes	KB	1000 bytes a para of text doc
Megabytes	MB	1000 KB
Gigabytes	GB	1000 MB
Terabytes	TB	1000 GB
Petabytes	PB	1000 TB
Exabytes	EB	1000 PB
Zettabyte	ZB	1000 EB

7) Big Data Characteristics
1) • Volume
3) • Value
5) • Veracity
4) • Visualization
2) • Variety
6) • Velocity
7) • Virality

1) Volume
The name 'Big Data' itself is related to a size which is enormous.

Volume is a huge amount of data.

Page No.:
Date:

To determine the value of 'data, size of data plays a very crucial role. If the volume of data is very large, then it is actually considered as a 'Big Data'. This means whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data.

- the size and amounts of big data that companies manage and and analyze
- we measure the volume of our data in terabytes and petabytes & not gigabytes
- Hence, 'Volume' is one characteristic which needs to be considered while dealing with Big Data

2) Variety.
- Variety refers to heterogeneous sources and the nature of data, both structured, unstructured & semi-structured.
- It refers to nature of data that is structured, semi-structured & unstructured data

- Variety is basically the arrival of data from new sources that are both inside & outside of an enterprise.

- Structured data - This data is basically an organized data.
It generally refers to data that has defined the length and format of data.

- Semi-Structured data - This data is basically a semi-organised data.

It is generaly a form of data that do not conform to the formal structure of data

Unstructured data - This data basically refers to unorganized data.
It generally refers to data that doesn't fit neatly into the traditional row and column structure of the relational database.

Texts, pictures, videos etc. are the examples of unstructured data which can't be stored in the form of rows and columns.

3) Value.

- Value is an essential characteristic of big data. It is not the data that we process or store.

- It is valuable & reliable data that we store, process, & also analyze.

- The bulk of data having no value is of no good to the company, unless you turn it into something useful.

- Data in itself is of no use or importance but it needs to be converted into something valuable to extract information.

- No matter how fast the data is produced or its amount, it has to be reliable & useful. Otherwise, the data is not good enough for processing or analysis.

Research says that poor quality data can lead to almost a 20% loss in a company's revenue

- Data scientists first convert raw data into information.
Then this data set is cleaned to retrieve the most useful data.

Analysis and pattern identification is done on this data set. If the process is a success, the data can be considered to be valuable

4) Visualization
- is the process of displaying data in charts, graphs, maps & other visual forms

- process of representing data in a visual format to make it easier to understand, analyze, and derive insights from large & complex data sets

- Effective visualization helps transform raw data into meaningful information through the use of visual elements like charts, graphs, maps & dashboards.

Various software tools & platforms are available for creating data visualizations, such as Tableau, Power BI, D3.js, Google Data Studio.

- powerful means to understand & communicate complex data sets

5) ~~Variety~~ Veracity
- means how much the data is reliable.

- process of being able to handle and manage data efficiently.

- Veracity refers to uncertainty of available data

- Ex: Data in bulk could create confusion whereas less amount of data could convey half or Incomplete Information

- focuses on quality & reliability of data

- It highlights the uncertainties & inconsistencies inherent in data and addresses the degree of reliability & meaningfulness of the data being analyzed.

- High veracity means the data is accurate, precise, & reliable; while low veracity indicates data that is uncertain or inconsistent

- key aspects of veracity: data quality, data source reliability, data consistency, data completeness data integrity

6) Velocity

- is the rate at which data grows. Social media contributes a major role in the velocity of growing data

- Velocity creates the speed by which the data is created in real-time.

- This speed of data producing is also related to how fast this data is going to be processed.

This is because only after analysis and processing, the data can meet the demands of the clients / users.

- Massive amounts of data are produced from sensors, social media sites, and application logs - and all of it is continuous.

If the data-flow is not continuous, there is no point in investing time or effort on it.

7) Virality

- Virality refers to the rapid spread and sharing of information content, or data through networks, often facilitated by social media, online platforms, and digital communication tools.

- High levels of user interaction, including likes, shares, comments, and reposts, contribute to the virality of content.

- Influential individuals or entities can significantly boost the spread of content through their networks.

- Content that is humorous, emotional, relatable, or provides value is more likely to go viral

- Social media & online platforms use algorithms to promote viral content, further accelerating its spread.

8) Challenges of Conventional System. (Traditional IT systems)

- ~~1) Scalability Issues~~
~~-(i) limited Resources~~
(i) Volume of Data
(ii) Processing & Analyzing
(iii) Management of Data

1) Volume of Data
i) Exponential Growth - The volume of data generated by organizations today is exponentially larger than in the past.
- Conventional systems were not designed to handle such massive amounts of data.

ii) Storage limitations - Traditional systems often have fixed storage capacities, which can quickly become ~~inde~~ inadequate.
- Scaling up storage in these systems is typically costly and complex.

(iii) Data Variety - Modern data comes in various forms, such as structured, semi-structured, and unstructured data.
- Conventional systems struggle to store & manage this variety effectively.

→ Impact: High Costs.

2) Processing and Analyzing
i) Performance Bottlenecks - Traditional systems often face performance bottlenecks when processing large datasets.
- They may lack the necessary computing power and ~~plat~~ parallel processing capabilities.

ii) Real-Time Processing - Conventional systems struggle with real-time data processing requirements.
- They are typically designed for batch processing, which is not suitable for appl° needing immediate insights.

(iii) Complex Data Analysis - Advanced analytics, such as ML & predictive analytics, require significant computatio

resources & specialized tools that traditional systems may not support.

Impact: Slow Decision-Making, Inefficiency.

3) Management of Data

i) Data Integration - Integrating data from various sources is challenging in conventional systems.
- They often lack the flexibility to seamlessly combine & manage data from disparate systems.

ii) Data Quality - Ensuring data accuracy, consistency and completeness is a significant challenge.

(iii) Data Security - Protecting data from unauthorized access, breaches, and cyber threats is critical.
- Conventional systems may rely on outdated security protocols, making them vulnerable to attacks.

(iv) Scalability and Flexibility - Traditional data management solutions often lack the scalability & flexibility to adapt to changing business needs and data growth.

Types of Big Data:
i) Unstructured
ii) Semi-structured
iii) Structured

1) Unstructured Data
- data with unknown form or structure.
- Large size, poses challenges in processing.
Ex - Text files, images, videos, search results from Google.
- Subcategories:-
i) Human-Generated Data: Data produced by humans (e.g. social media posts).

ii) Machine-Generated Data: Data produced by machines (e.g. sensor data).

Ex - O/P returned by a Google search.

2) Structured Data
- data can be stored, accessed & processed in a fixed format.
- known format, easier to work with and extract value from, large sizes.
- Data stored in relational database management systems (RDBMS).

Example: Employee table.

Emp-Id	Emp_Name	Gender	Department	Salary
1	XYZ	Male	Finance	85000
2	ABC	Male	Admin	55000
3	PQR	Female	Sales	9900000

3) Semi-structured Data

- Contains both structured & unstructured data elements.
- Not stored in a traditional database but has tags or markers to identify data elements.

Ex:- Web application data, log files, transaction history files

Ex: Personal data stored in an XML file

```
<rec>
    <name> Prashant </name>
    <sex> Male </sex>
    <age> 35 </age>
</rec>
<rec>
    <name> Seema </name>
    <sex> Female </sex>
    <age> 41 </age>
</rec>
```

Storage of Appln :-

10) Difference :-

Structured Data	Semi-structured Data	Unstructured Data
based on Relational database table.	based on XML/ RDF (Resource Description Framework)	Based on character & binary data.
Matured transaction & various concurrency techniques.	Transaction is adapted from DBMS not matured.	No transaction management & no concurrency.
Schema dependent & less flexible.	more flexible than structured data & less than unstru. data.	It is more flexible & there is absence of schema.
It is very difficult to scale DB schema	It's scaling is simpler than structured data	It is more scalable
structured query allow complex joining	Queries over anonymous nodes are possible	Only textual queries are possible.
very robust in nature.	is fairly new tech. It is not very robust in nature.	
Less DBMS, RDF, Financial data, Relational table	significant sensor logs, sensor o/p	Large No sql, video, audio, social media, online forum

11) Difference

Traditional Data	Big Data
1) generated in enterprise level	1) generated outside the enterprise level
2) Its volume ranges from Gigabytes to Terabytes	2) Its volume ranges from Petabytes to Zettabytes or Exabytes
3) deals with structured data.	3) deals with structured, semi-structured & unstructured data.
4) is generated per hour or per day or more.	4) is generated more frequently mainly per seconds.
5) is centralized & it is managed in centralized form.	5) is distributed and it is managed in distributed form
6) Data integration is very easy	6) Data integration is very difficult.
7) Size of data is very small	7) The size is more than the traditional data size
8) Traditional database tools are required to perform any database operation.	8) Special kind of database tools are required to perform any database schema based operation.
9) Normal functions can manipulate data.	9) Special kind of functions can manipulate data
10) is stable & inter-relationship	10) is not stable and unknown relationship
11) is in manageable volume	11) is in huge volume which becomes unmanageable
12) easy to manage & manipulate	12) difficult to manage & manipulate data
13) Data sources includes ERP transaction data, CRM transaction data, financial data, organizational data.	13) Its data sources includes social media, device data, sensor data, video, images, audio etc.

12) Case Study of Big Data Solutions:-

1) Walmart

Scenario: Walmart, the world's largest retailer, uses Big Data to enhance customer experience & retention.

Solution: - It used data mining to find designs pattern that can be used to give product suggestions to client, depending on which products were brought together.

- It utilizes Hadoop and NoSQL databases to process real time data

Outcome: - Increased conversion rates & enhanced customer experience

2) Netflix

Scenario: Netflix aims to enhance user experience by predicting content preferences.

Solution: Uses Big Data to analyze viewing habits, playback interruptions, and ratings.

- Integrates Hadoop, Hive & Pig for data processing and analysis.

Outcome: Improved content recommendations and customer satisfaction, supporting Netflix's strategy as a content creator.

13) Intro to Hadoop & Big data UDM

14) Big Data UDM

15) Challenges for processing big data

c Processing big data refers to the reading, transforming, extraction, and formatting of useful info from raw information.

- Handling & storing massive amounts of data.

- Managing diverse data types (structured, semi-structured, unstructured)

- Processing data in real-time or near-real-time as it is generated

- Ensuring data accuracy, quality & trustworthiness.

- Ensuring systems can scale efficiently as data grows.

16) Tech Support for Big Data.

C

1) Infrastructure Support.

(i) Cloud Services - Utilize cloud platforms like AWS, Google cloud, and Microsoft Azure for scalable storage & processing capabilities

(ii) Distributed Systems - Implement and manage distributed file systems like Hadoop HDFS and cloud-based data lakes.

(iii) Cluster Management - Use tools like Apache Ambari or Cloudera Manager for managing and monitoring Hadoop clusters.

2) Data Storage Support

(i) NoSQL Databases - Support for databases like MongoDB, Cassandra, and HBase that can handle semi-structured and unstructured data.

3) Data Processing and Analysis Support.

(i) Big Data Frameworks - Provide support for frameworks like Apache Spark, Apache Flink, and Apache Hadoop for large-scale data processing.

(ii) Real-Time Processing - Utilize stream processing tools such as Apache Kafka and Apache Storm for real-time data analytics.

17) History of Hadoop

1) 2003-2004 : Google Influence
- Google publishes GFS and Map-Reduce papers.

2) 2005: Creation
- Doug Cutting and Mike Cafarella create Hadoop for Nutch.

3) 2006: Yahoo! Contribution
- Yahoo! supports Hadoop's development.

4) 2008: Apache Project
- Hadoop becomes an Apache project with its first official release.

5) 2009-2010: Ecosystem Expansion
- Intro of Hive, HBase, Pig, Zookeeper.

6) 2011: Hadoop 1.0 and YARN
Stable release with improved scalability: introduction of YARN.

7) 2013 Onwards: Enterprise Adoption
Major enterprises adopt Hadoop.
- commercial distributions emerge.

8) Modern Era: Continued Evolution
Integration with cloud, real-time processing, & machine learning tools.

18) Use cases of hadoop:

1) Data Storage and Processing
Facebook: stores and processes petabytes of user data to provide analytics and insights.

2) Log Analysis
Yahoo!: Analyzes massive amounts of server logs to optimize performance and user experience.

3) Recommendation Systems
Netflix: Uses Hadoop to process viewing data and generate personalized recommendations for users.

4) Healthcare Analytics
Healthcare Providers: Processes patient data to identify trends, predict outbreaks, and improve patient care.

5) Search Engine Optimization
LinkedIn: Uses Hadoop to analyze user interaction data to improve search algo & user engagement.

6) Social Media Analysis
Twitter: Uses Hadoop to store and analyze tweet data for sentiment analysis & trend prediction.

7) Retail Analytics
Walmart: Analyzes customer purchasing data to optimize inventary and personalize marketing strategies.

19)

RDBMS	Hadoop
1) primarily handles structured data.	1) Handles structured and unstructured data
2) Vertical scaling: enhance single server.	2) Horizontal scaling: add commodity hardware.
3) Licensing fees	3) opensource software.
4) is a system software for creating & managing databases that based on the relational model	4) Hadoop is a coll" of open source software that connects many computers to solve problems involving a large amount of data & computation
5) Data normalization is required in RDBMS	5) Data normalization is not required.
6) data schema is static type.	6) data schema is dynamic type.
7) data resides in tables having relational structure	7) uses key/value pairs
8) batch processing using Mapreduce or Spark	8) Interactive querying using SQL

20) When to Use Hadoop

i) Large Data Volumes - Ideal for storing and processing large datasets that exceed the capabilities of traditional databases.

ii) Variety of Data - suitable for handling various data types, including structured, semi-structured, and unstructured data.

iii) Scalability - Best used when there is a need to scale out by adding more nodes to handle increased data loads.

iv) Cost Efficiency - Economical for storing large amounts of data using commodity hardware.

v) Complex Data Analysis - Ideal for performing complex data analysis and machine learning tasks on large datasets.

21) When not to use Hadoop

i) Small Datasets - Overkill for small datasets that can be efficiently handled by traditional relational databases.

ii) Transactional Data - Inappropriate for use cases requiring ACID transactions, such as financial systems.

(iii) Simple Queries - Not ideal for applications needing simple read/write operations where a traditional SQL database would be more efficient.

21) Advantages of Hadoop
- Varied data sources
- cost-effective
- performance
- fault-tolerant
- highly available
- high throughput
- Open source
- scalable
- ease of use
- compatibility
- multiple languages supported

22) Disadvantages of Hadoop
- Issue with small files
- Security
- requires specialized knowledge & skills to set up, manage & maintain
- not designed for real-time data processing
- lacks support for ACID transactions
- ongoing maintenance and optimiz° are required to ensure performance
- job scheduling & resource management can be complex.

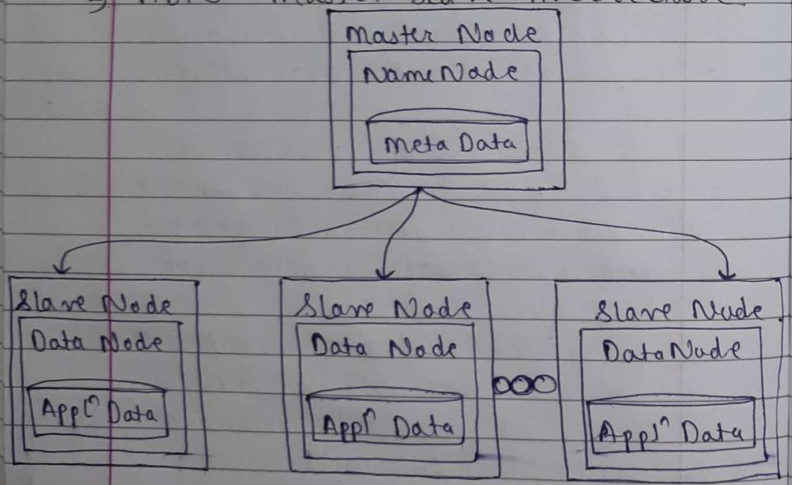
Unit II

Page No.:
Date:

1) HDFS
2) Significance of HDFS in Hadoop
3) Features of HDFS
4) Data Storage in HDFS
5) Data Replication
6) Accessing HDFS : CLI (& admin cmd)
7) Java Based Approach
8) Fault tolerance
9) Download Hadoop
10) Install^n & setup of Hadoop
11) start up & shut down process

1) HDFS - UDM notebook

2) HDFS Master-Slave Architecture



Page No.:
Date:

- HDFS Architecture comprises Slave/Master Architecture where the Master is NameNode in which metaData is stored ~~in~~ ~~the~~ and Slave ~~is~~ is the DataNode in which actual data is stored.

- This architecture can be deployed over the broad spectrum of machines which support Java.

- HDFS Architecture is a block-structured file system in which the division of file is done into the blocks having predetermined size.

- These blocks are stored on the different clusters.

- HDFS follows the master/slave architecture in which clusters comprise single NameNode referred to as Master Node and other nodes are referred to as the DataNodes or Slave Nodes.

- However, one is capable of running different DataNodes on the single machine whereas, in reality,

these DataNodes are present on the different machines

Working of HDFS Architecture

- HDFS divides the information into separate blogs and distributes those blogs to various nodes present in the cluster.
Thus, it enables efficient parallel processing.

- HDFS architecture has high fault tolerance.
- The filesystem copies or replicates every piece of data multiple times and then distributes the copies to the different nodes.

- This makes sure that, if the data present on a node crashes, it can be easily found somewhere else within the cluster.

Core Components :-

i) NameNode
ii) DataNode

i) NameNode

- It is the master daemon that maintains and manages the DataNodes (slave nodes)

- stores the metadata
ex: number of replicas, data blocks, size of the files, permissions, hierarchy.

- This metadata is present in the master memory for a quick data retrieval.

- manages and maintains the slaves nodes and is responsible for assigning tasks to them.

- It helps in executing the file system execution. For ex: opening, closing, naming the files & directories.

- It records each and every change that takes place to the file system metadata.

- If a file is deleted in HDFS, the NameNode will immediately record this in the EditLog.

- is the commodity hardware that contains the GNU/Linux operating system and the namenode software.

- does following tasks:
  - executes file system operations.
  - records each & every change that takes place to the file system metadata.
  - regularly receives a Heartbeat and a block report from all the datanodes in the cluster to ensure that the DataNodes are alive.
  - keeps a record of all the blocks in the HDFS and DataNode in which they are stored.

(ii) DataNode.
- slave daemon/process which runs on each slave machine.

- store actual data in blocks.

- A functional filesystem has more than one DataNode, with data replicated across them.

- It is also responsible for creating blocks, deleting blocks and replicating the same based

By default repl factor is 3

on the decisions taken by the NameNode.

(ii) Secondary Node.
- works concurrently with the primary NameNode as a helper daemon/process.

- It is responsible for combining the EditLogs with FsImage from the NameNode.

- Editlogs is a transaction log that records the changes in the HDFS file system or any action performed on the HDFS cluster such as addⁿ of new block, replⁿ, deletion etc. (recent modification)

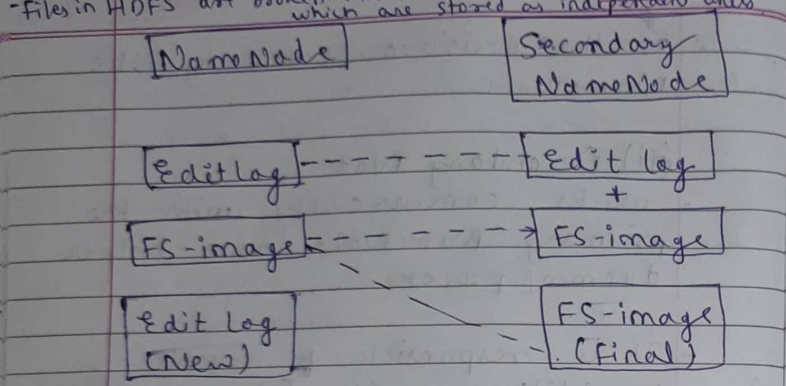
- FS-Image - is a snapshot of the file system's metadata at a certain point of time.

- Seconday node downloads Editlogs from the NameNode at regular intervals and applies to FsImage.

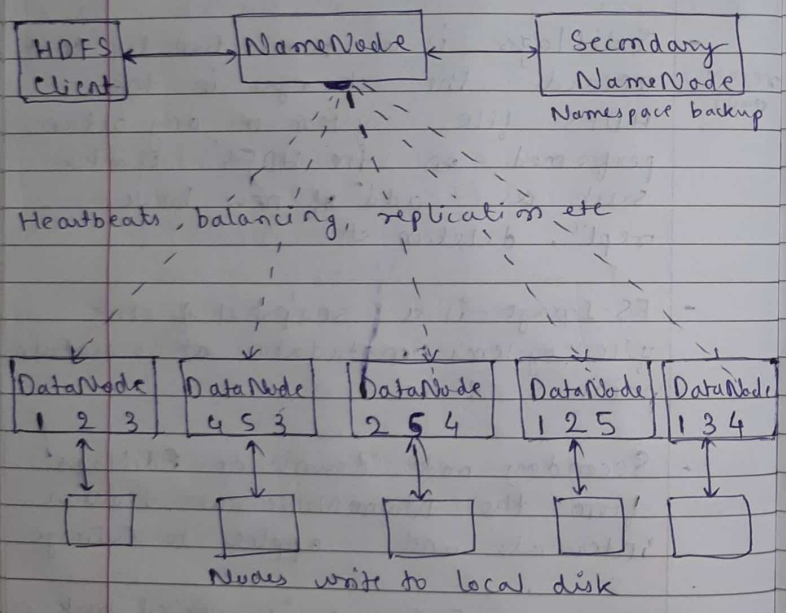
- The new FSImage is copied back to the NameNode, ~~which is~~.

- Hence, Secondary NameNode performs regular checkpoints in HDFS, hence CheckpointNode

Block - is a minimum amount of data that it can read or write.
-128 MB by default & this is configurable
-Files in HDFS are broken into block-sized chunks which are stored as independent units.



HDFS Architecture :-



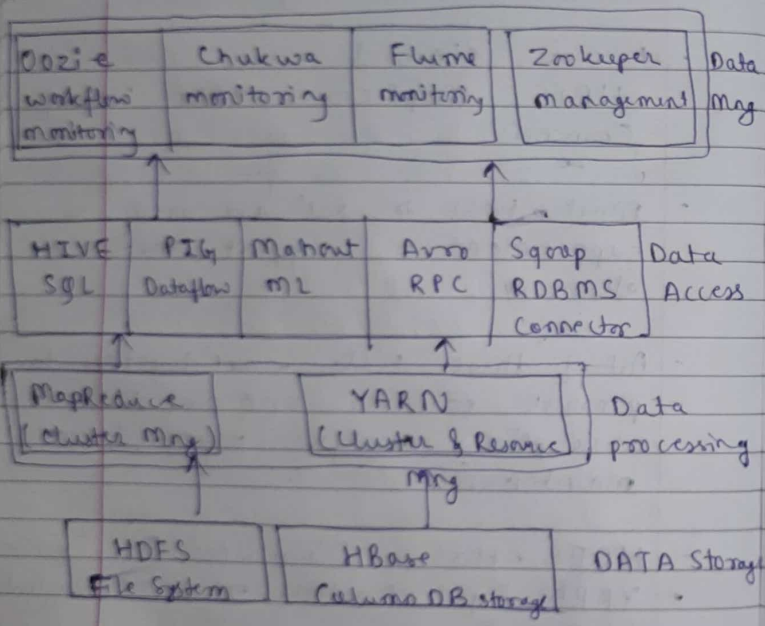
HDFS Write Architecture :- msg.

Hadoop Ecosystem.

- is a framework that can process large data sets in the form of clusters.
- is a platform that provides various services to solve big data problems
- 4 main elements
  HDFS, MapReduce, YARN, Hadoop Common.
- Most tools or sol^n are used to supplement or support these core elements.
- All of these tools work together to provide services such as data absorption, analysis, storage & maintenance

- HDFS - Hadoop Distributed File System
- YARN - yet Another resource Negotiator
- MapReduce -
- Spark
- PIG, +
- HIVE
- HBase - NoSQL database

- Mahout
- Spark MLLib
- Solar
- Lucene
- Zookeeper
- Oozie - Job Scheduling



Hadoop Ecosystem

1) HDFS
- core component / backbone of HE.
- possible to store large data sets (i.e. structured, unstructured & semi-structured)
- helps in storing our data across various nodes & maintaining log file about stored data (metadata)
- two main components
  - Namenode
  - Datanode

2) YARN -
- brain
- performs all your processing activities by allocating resources & scheduling tasks
- helps manage all resources in the cluster
- 3 main components -

i) Resource Manager - has the right to allocate resources for the appl^n

ii) Node Manager - is responsible for allocating resources such as CPU, memory for each machine

iii) Applic^n Manager - acts as an interface between resource managers and the node manager

Page No.:
Date:

3). Map Reduce

- core component of processing in a Hadoop Ecosystem as it provides the logic of processing.

- 2 function:-
Map() function performs actions like filtering, grouping and sorting.

Reduce() function aggregates and summarizes the result produced by map function

4) Pig

- developed by Yahoo,
- uses Pig Latin language which is query-based language similar to SQL
- platform for structuring data flows, processing and analyzing massive data sets.
- responsible for executing commands and processing all MapReduce activities in the background.
- helps simplify programming & optimization.

Page No.:
Date:

5) Hive

- highly scalable, because it supports real-time processing and batch processing.

- with the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query language)

HIVE + SQL = HQL

- Hive supports all SQL data types, making query processing easier.

6) Mahout

- allows automatic learning of systems or appl^n

- provides various libraries or functions such as collaborative filtering, clustering, and classification.

7) Apache Spark

- is a framework for real time data analytics in a distributed computing environment.

- The Spark is written in Scala &

originally developed at the University of California.

- better for real time processing, while Hadoop is designed to store unstructured data and perform batch processing on it.

- 100x faster than Hadoop for large scale data processing

- When we combine capabilities of Apache Spark with low-cost operations of Hadoop, we get best results.
So, many companies use Spark & Hadoop together to process & analyze big data stored in HDFS.

8) Apache HBase

- open source non-relational distributed database (NoSQL db)

- supports all types of data, which is why it can handle anything in the Hadoop ecosystem

- based on Google's Big Table; which is a distributed storage system designed to handle large data sets

- fault tolerant
- written in java

9) Zookeeper

10) Oozie

performs task of a scheduler, thus scheduling jobs and binding them together as a single unit.

- 2 kinds of job
1) Oozie workflow is a job that need to be executed in a sequentially ordered manner

2) Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

- uses divide & conquer approach to process large volume of data.
- distributed data processing algo
- can process big data in parallel on multiple nodes

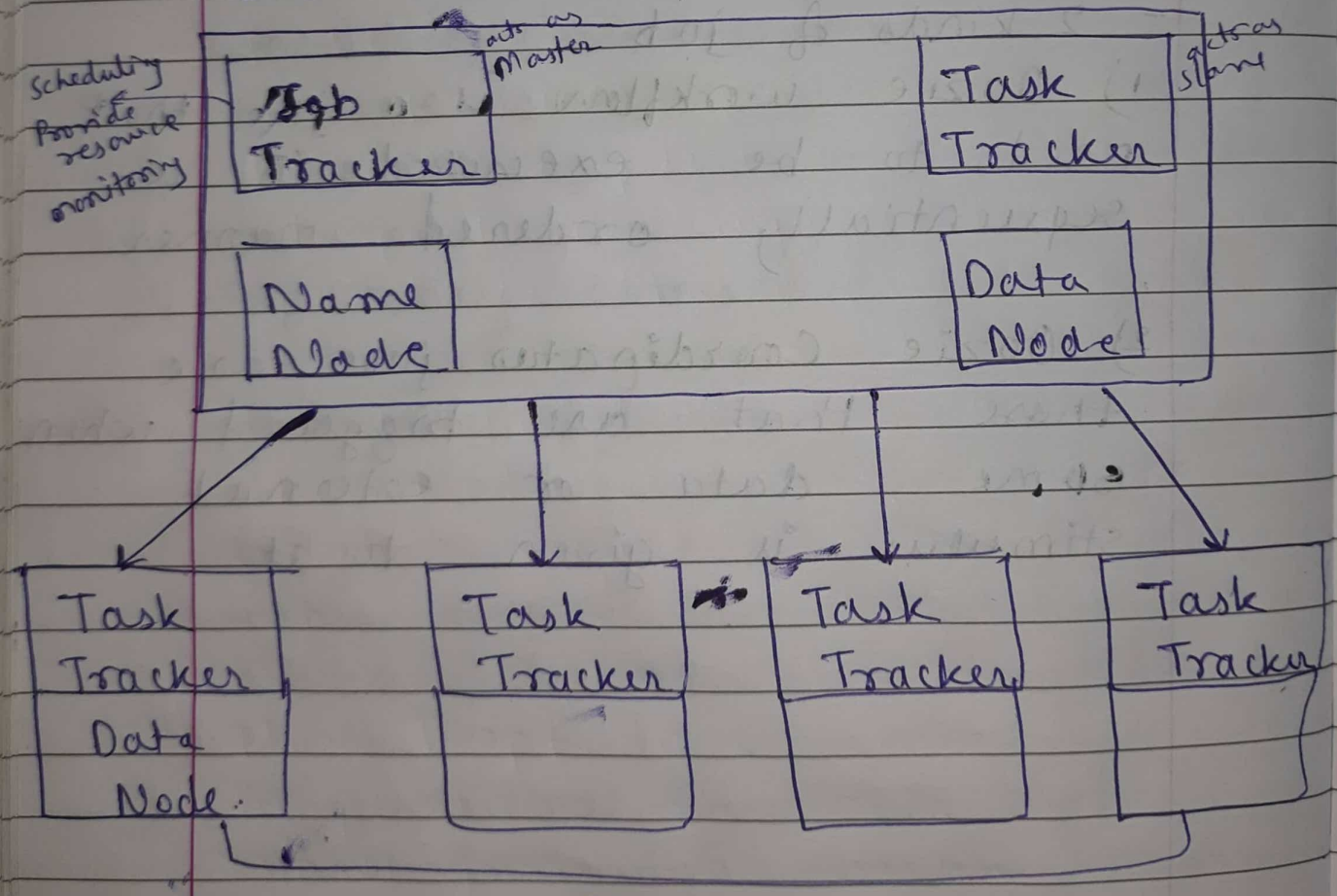
Page No.:
Date:

1) MapReduce

- performs the processing of large data sets in a distributed & parallel manner.

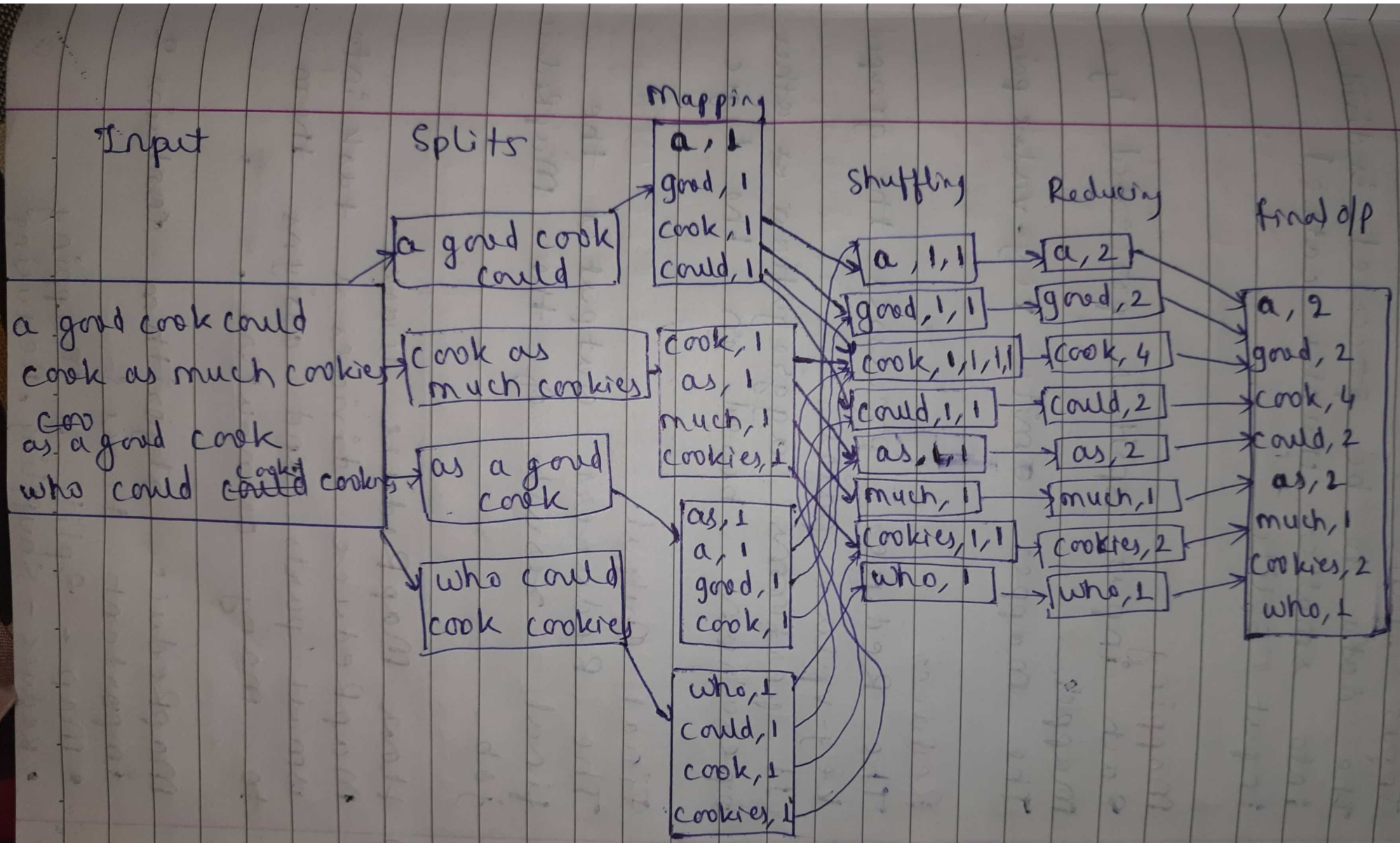
- consists of 2 distinct tasks - Map and Reduce

- Two essential daemons - Job Tracker & Task Tracker



- The data is first split & then combined to produce final result

2) Map Reduce Architecture



1) Input Splits -
The data to be processed is divided into smaller chunks called input splits.

2) Mapping -
- Each input split is processed by a Mapper
- The mapper emits key-value pair.

3) Reducer :
- The Reducer processes the grouped key-value pairs.
- It performs aggregation or other operations based on the logic.

4) Final Output
The Reducer's output is the final result of the MapReduce job.

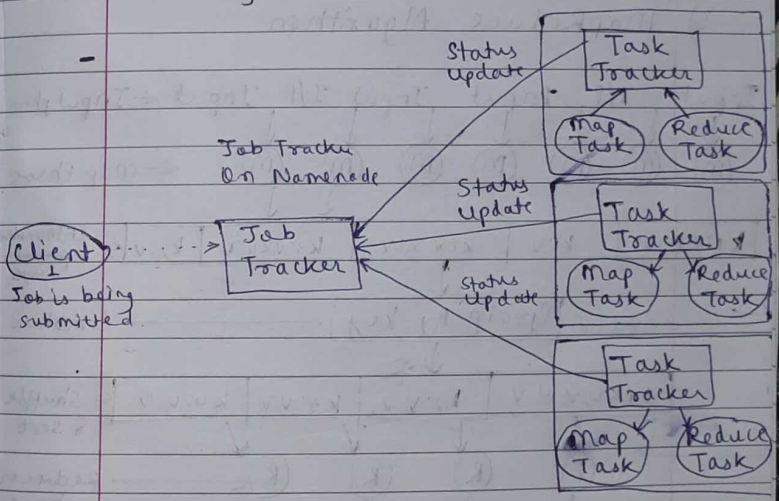
3) How MapReduce Works
MapReduce divides a task into small parts and assigns them to many computers.

MapReduce algorithm contains 2 important tasks :
- Map - Split & Mapping
- Reduce - Shuffling, Reducing

- The complete execution process is controlled by two types of entities called :-

- Job Tracker : Acts like a master (responsible for complete execution of submitted job)

- Multiple Task Trackers : Acts like Slaves, each of them performing the job.



- A job is divided into multiple tasks which are then run onto multiple data nodes in a cluster.

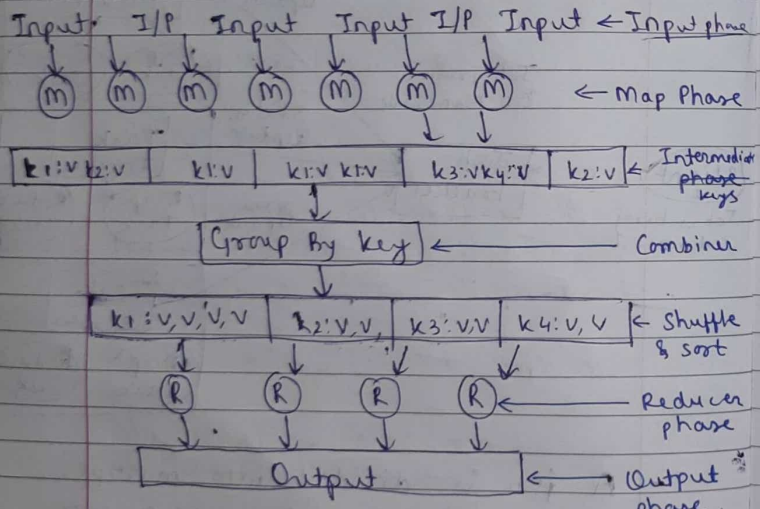
- It is the responsibility of job tracker

to coordinate the activity by scheduling tasks to run on different data nodes.

Task tracker executes the job & send progress report to the job tracker.

Job tracker keeps track of the overall progress of each job.

4) MapReduce Algorithm



1) Input Phase -
Record Reader translates each record in an input file and sends the parsed data to the mapper.

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

3) Intermediate keys
key value pairs generated by the mapper are known as intermediate keys

4) Combiner
- not a part of main MapReduce algo (optional)
- groups similar data from map phase

5) Shuffle and Sort
Reducer task starts with the Shuffle & Sort step.

downloads the grouped key value pairs onto the local machine.

The individual key-value pairs are sorted by key

6) Reducer

~~data can~~ takes grouped key-value paired data as input & runs a Reducer function on each one of them.

Data can be aggregated, filtered & combined in number of ways.

Once the execution is over, it gives zero or more key-value pairs to the final step.

7) Output Phase

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

Map Reduce Feature

- Scalability
- Flexibility
- Security & Authentication
- Cost Effective Sol?
- Fast