

Big Data and Hadoop Ecosystems



Prepared by: Sunil Kumar

- **What is Data ?? Data and its storage pattern in computing environment.**
- **Data generation sources.**
- **Challenges in Data Management and data accessing.**
- **What is Big data?**
- **Apache Hadoop as a solution.**
- **History of Hadoop.**
- **Features of Hadoop .**
- **Core components of Hadoop and its versions V.1,V2.0,V3.0**
- **HDFS Components (Name Node, Data Node)**
- **Read and Write in HDFS**
- **Lab Work on HDFS commands**
- **Summary of session**

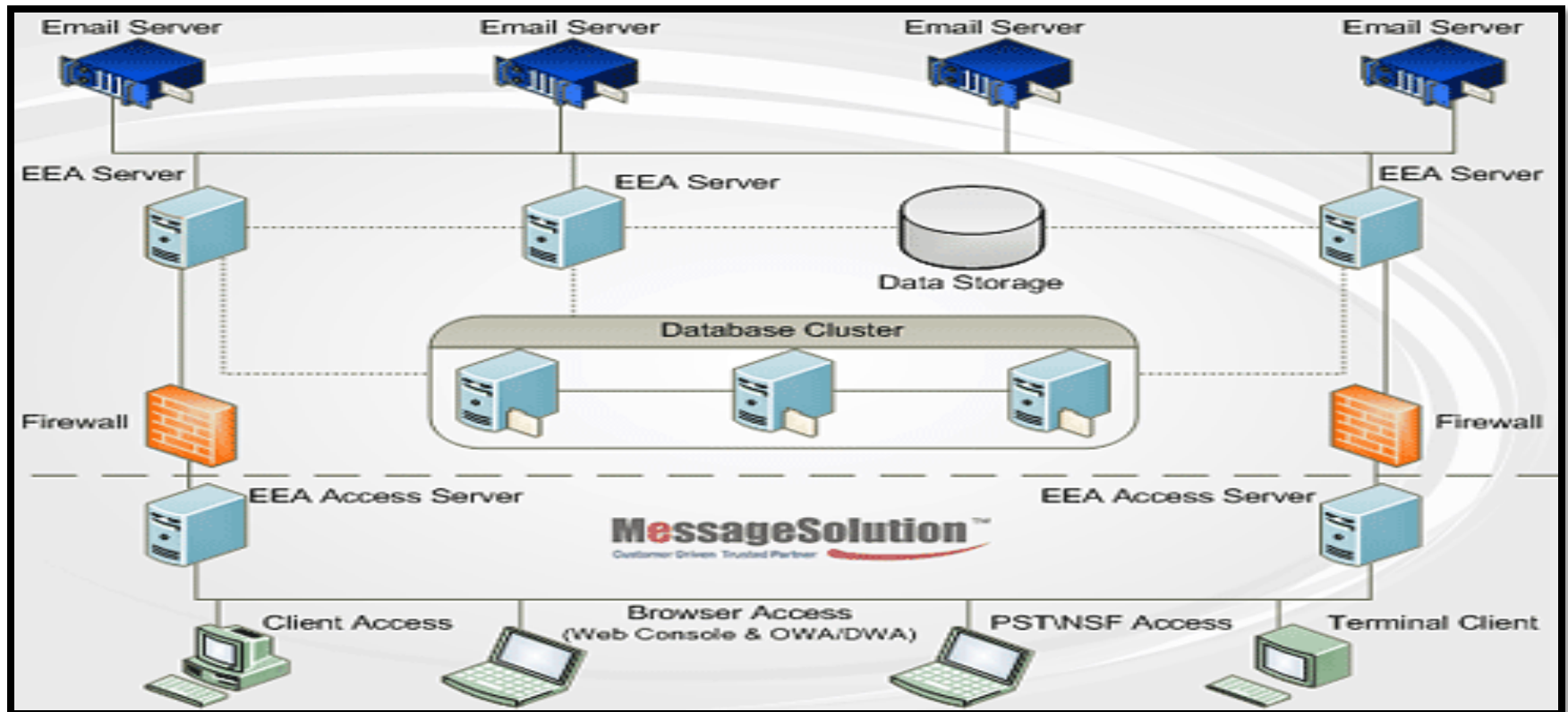
- Data is a collection of discrete values that convey information, describing quantity, quality, fact, statistics and measurements etc.
- Data can be collected by using various techniques such as measurement, observation, query, or analysis, and typically represented as numbers or characters which may be further processed.
- Data can be organized in graphs or charts for analysis etc.





Data storage in computing environment

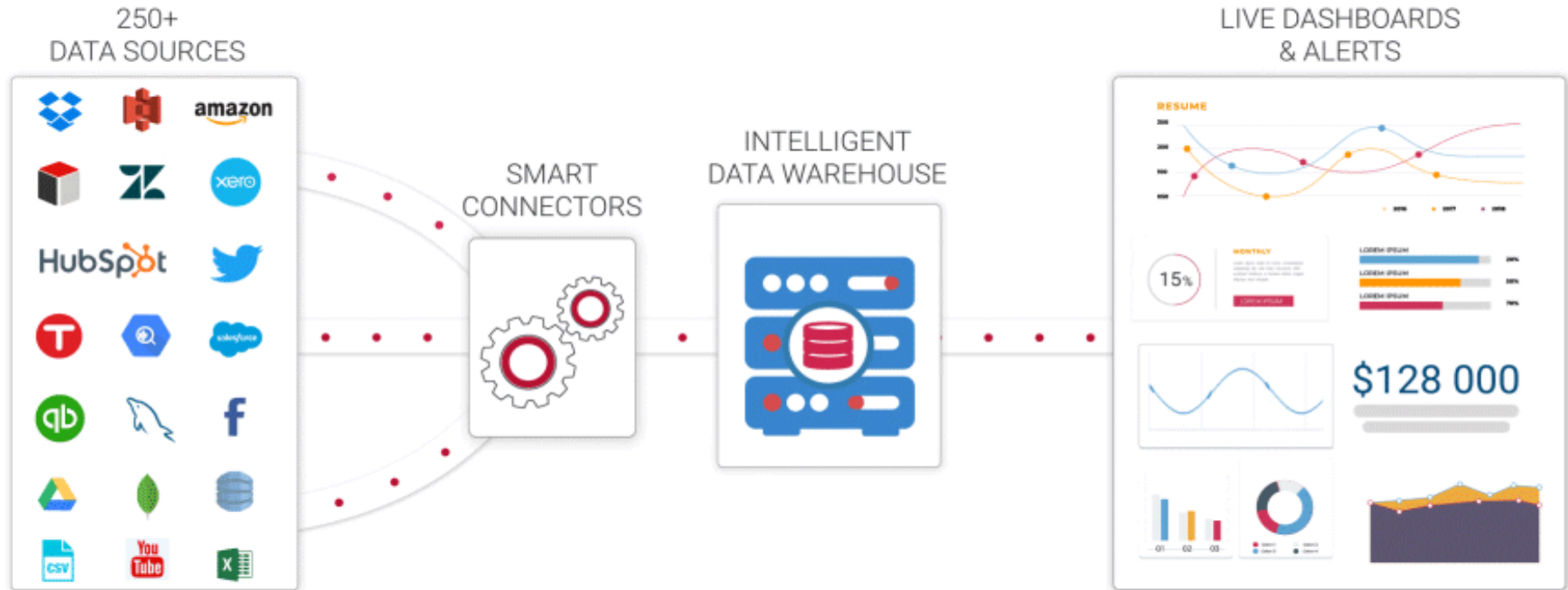
Large volume of data has been generating everyday through various web applications.

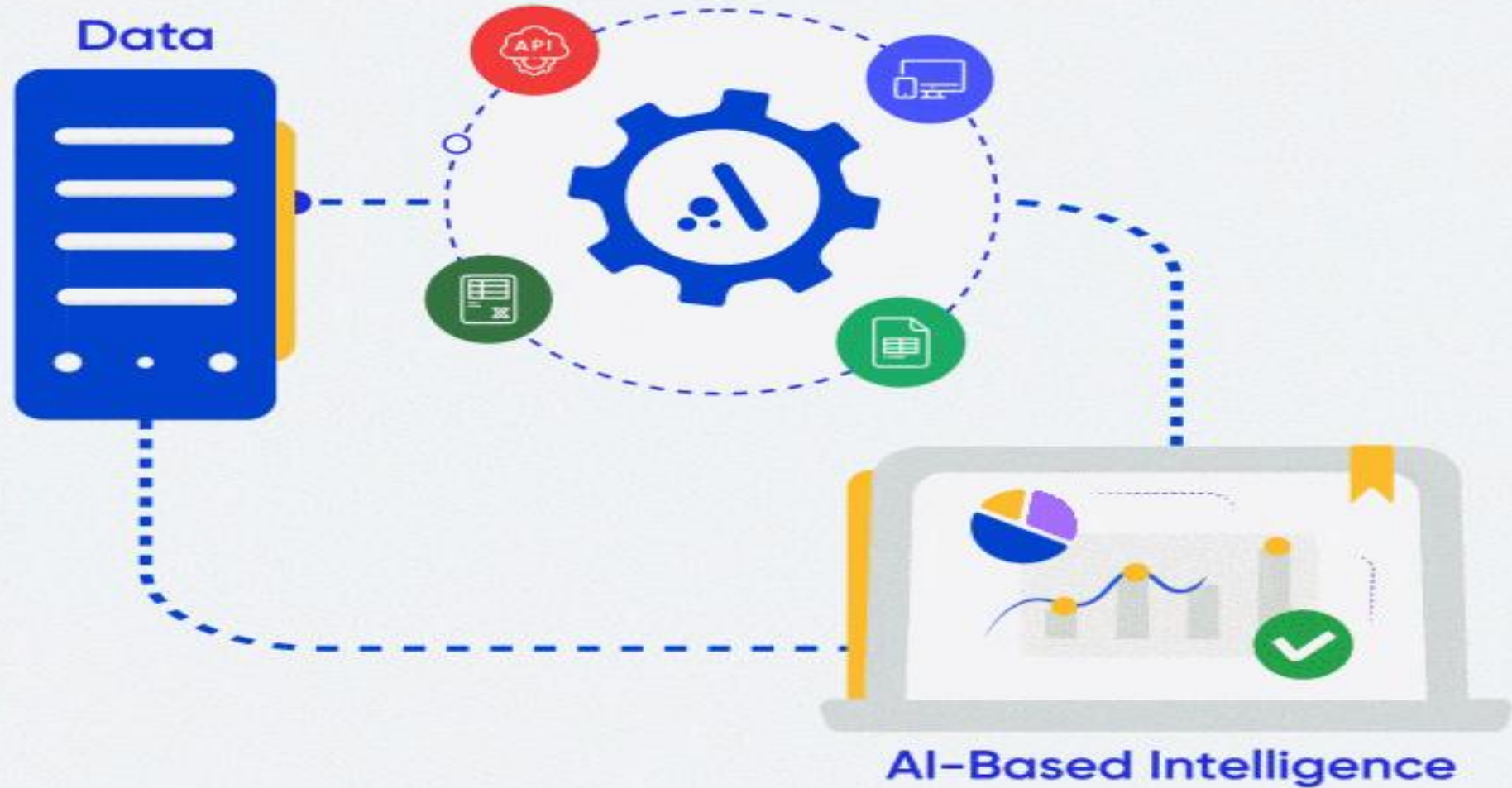


Data generation sources



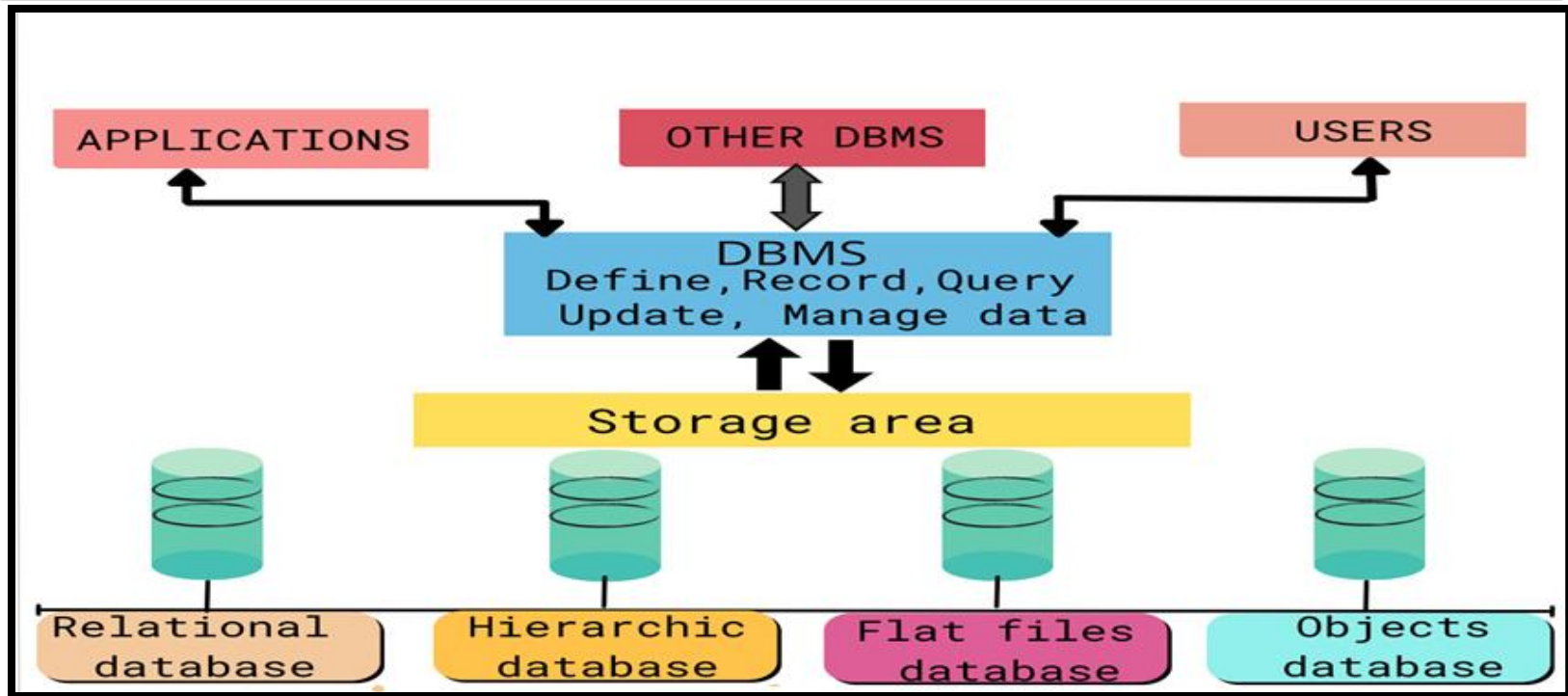
Data Coming from Various Sources in Real time





Challenges arises in data management and data accessing

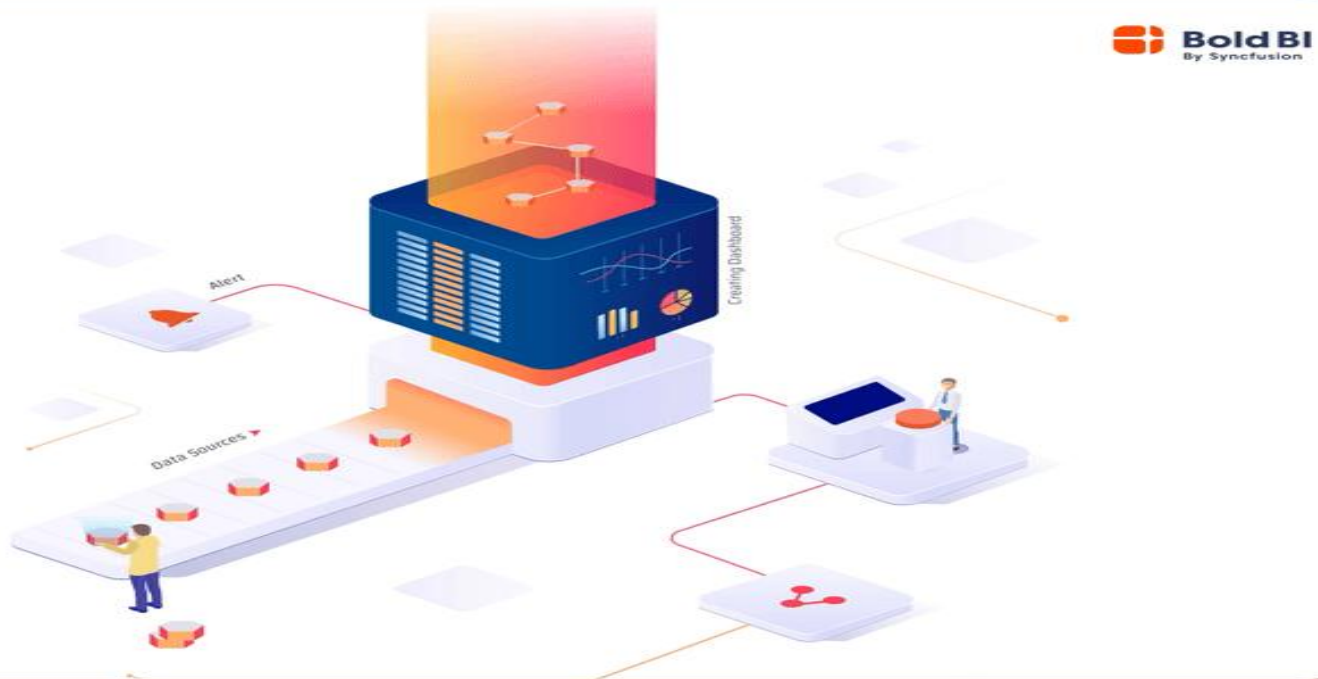
A. Data Base management and accessing method.



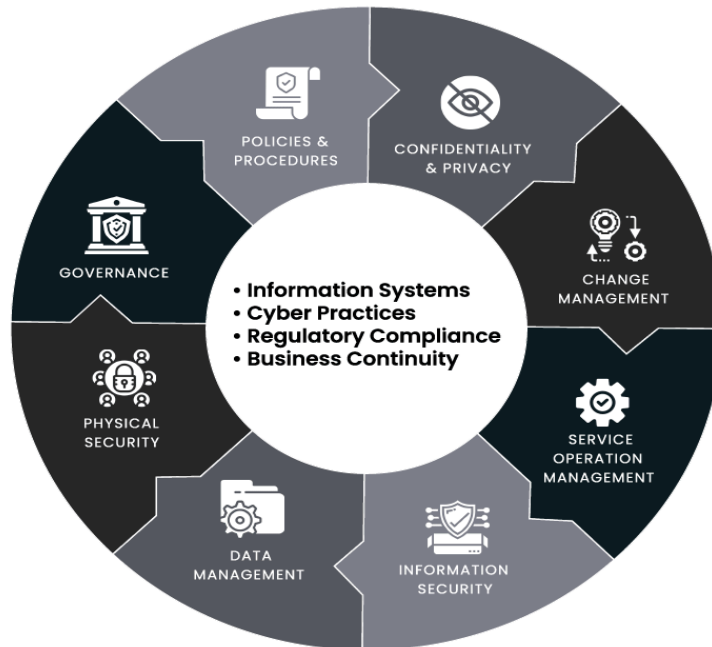
B. Data mining to retrieve required data from historical data.



C. How quickly result shall be processed to user



D. Privacy and Security of data.



E. Need Quick response time

HOW DO CREDIT CARDS WORK?



What is Big data?

➤ Big data refers to data sets that are too large in volume and complex to handle with by traditional data-processing application software.



Big data(Cont...)

- Big data is a revolution in the field of Information Technology.
- Big data has the characteristics of high variety, volume, and velocity etc.
- Big data involves the use of analytics techniques like machine learning, data mining, natural language processing, and statistics.
- Using big data tools Can store large amount of data, process data , analyze the data and visualize the data.

Real-time Benefits of Big Data Analytics:

The use of Big Data analytics is very flexible. With the use of big data a lot there has been an enormous growth in multiple industries. Some of them are-

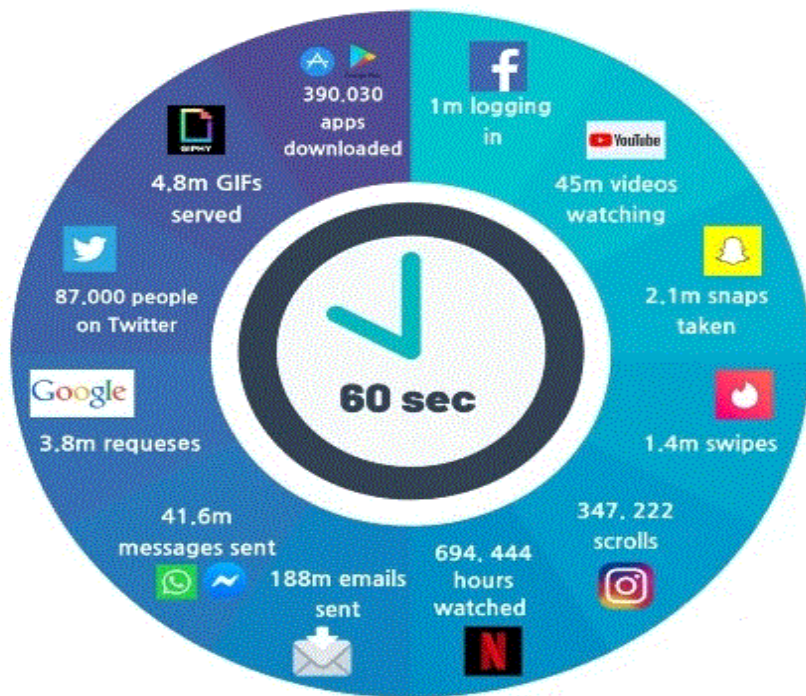
- **Banking**
- **Technology**
- **Consumer**
- **Manufacturing**

Example of Big Data?

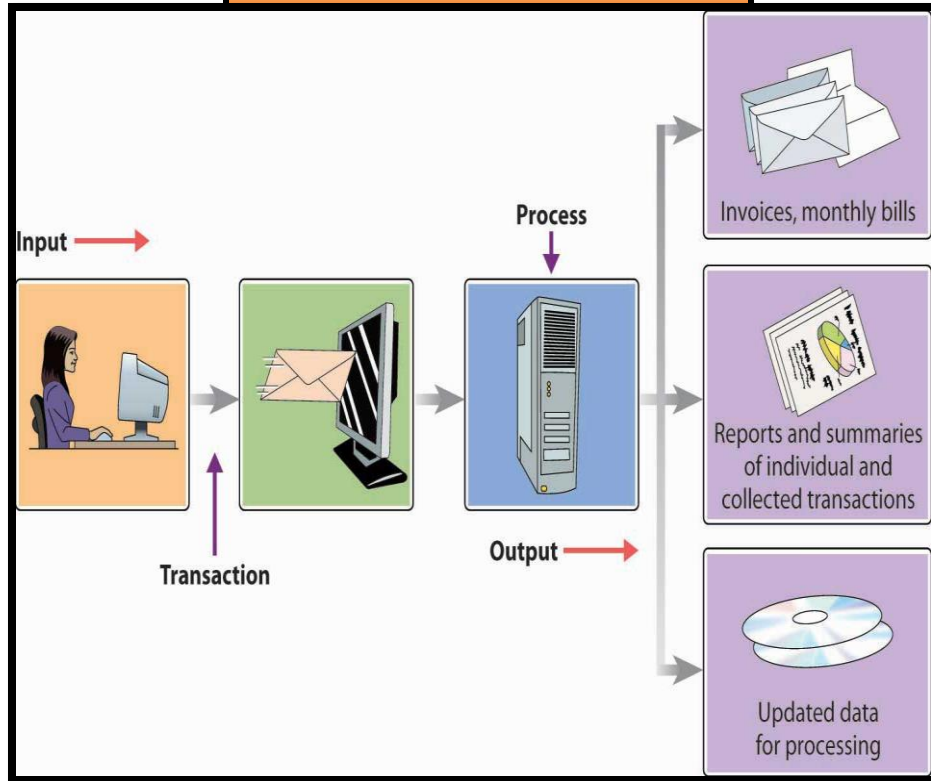
- The **New York Stock Exchange** is an example of Big Data that generates about **1- terabyte** of new trade data per day.
- The statistic shows that **500+terabytes** of new data get ingested into the databases of social media site **Facebook**, every day. This data is mainly generated in terms of photo and video uploads, message exchanges, putting comments etc.
- A single **Jet engine** can generate **10+terabytes** of data in **30 minutes** of flight time. With many thousand flights per day, generation of data reaches up to many **Petabytes**.

Examples of Bigdata

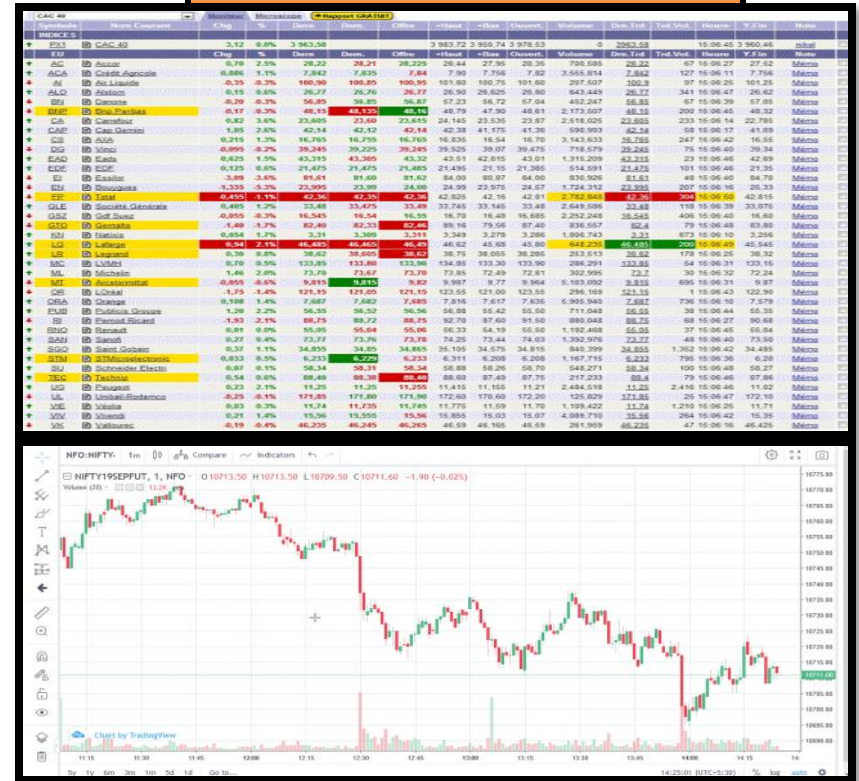
1.Social Media sites /Network- Approximately 500+TB data has been generating through social media everyday.



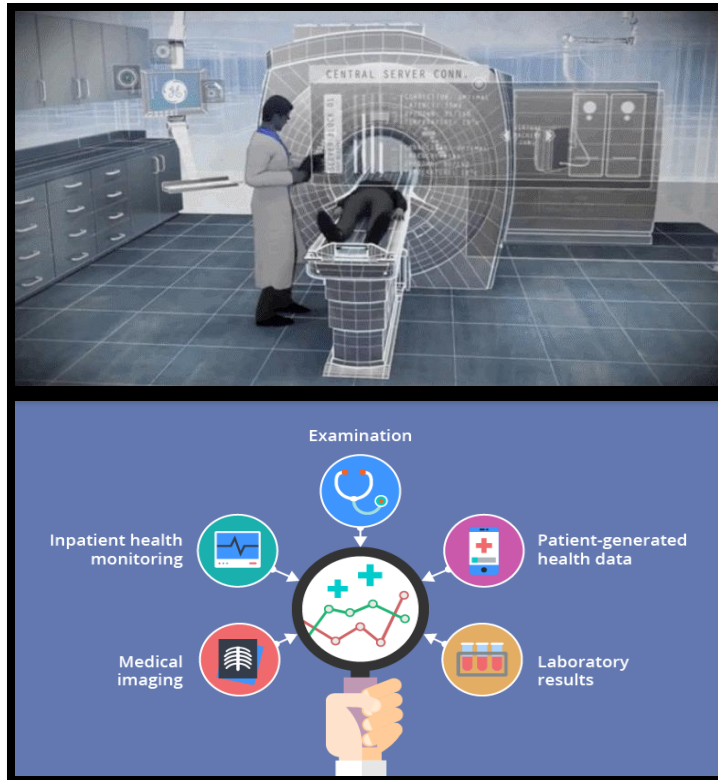
2. Transactional Data



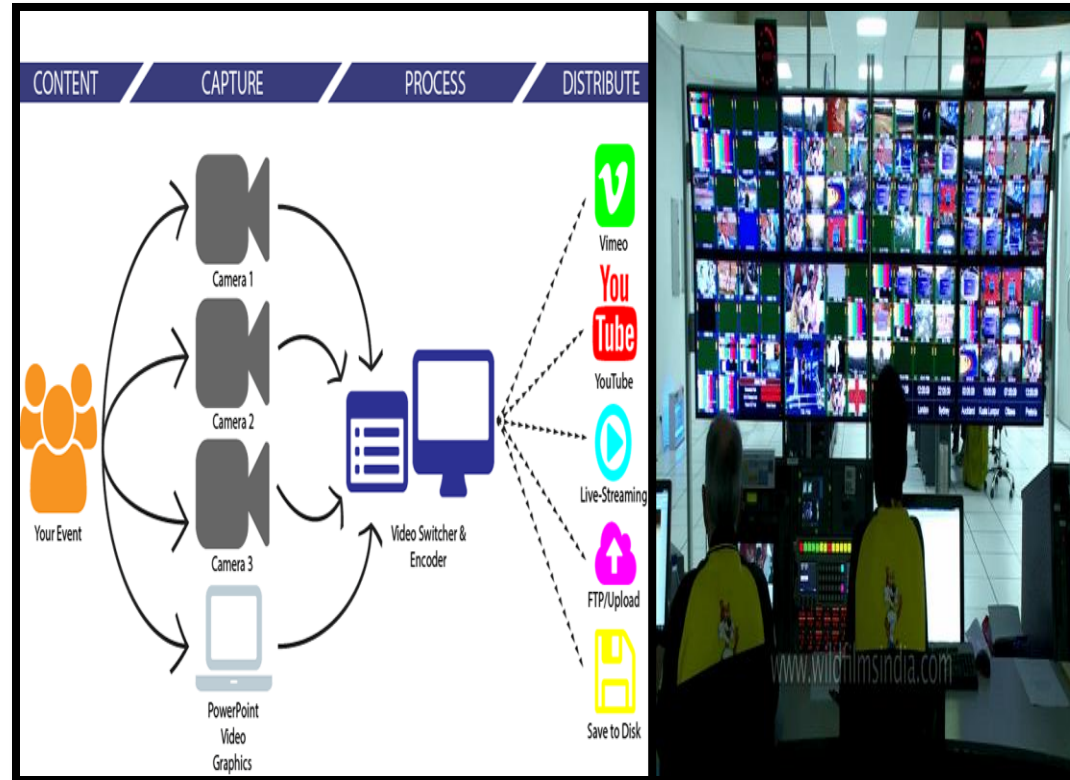
3. Stock Exchange Data



4. Healthcare Data

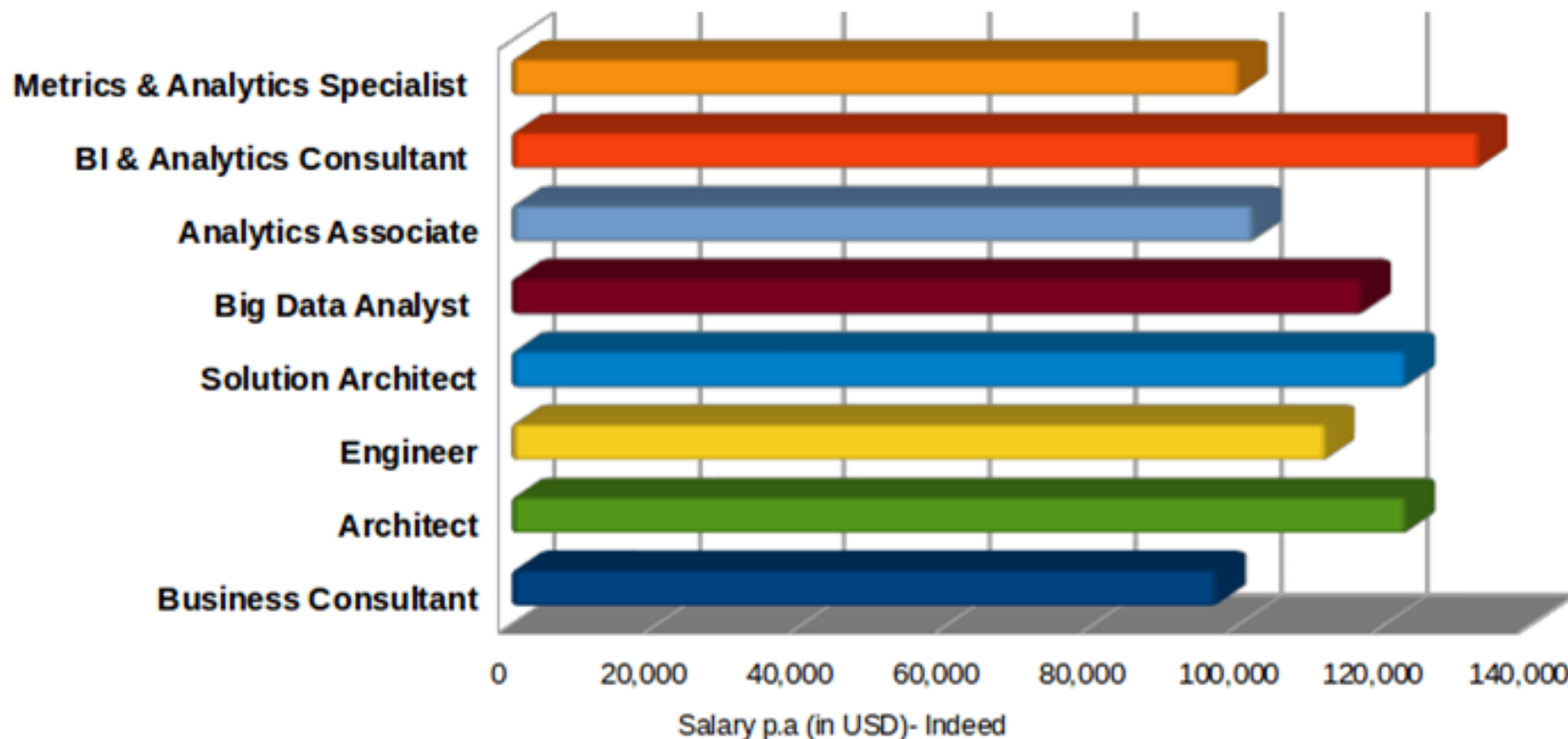


5. Live Broadcasting



Opportunities in Big Data Analytics:

Big Data Analytics Job Titles & Salaries



3 -Types of Big data

1.Structured Data – Used a specific Data model to store or create data i.e **relational database model**.(Ex- SQL data store in table).

2.Unstructured Data- No specific data mode to store data used NoSQL methodology i.e no relational database like- **MongoDB** (Ex- pdf file, .exe file, image, video etc.)

3.Semi-structured data- Used RDF(Resource Description framework) methodology to store data not relational data base methodology. Ex-**XML**



STRUCTURED

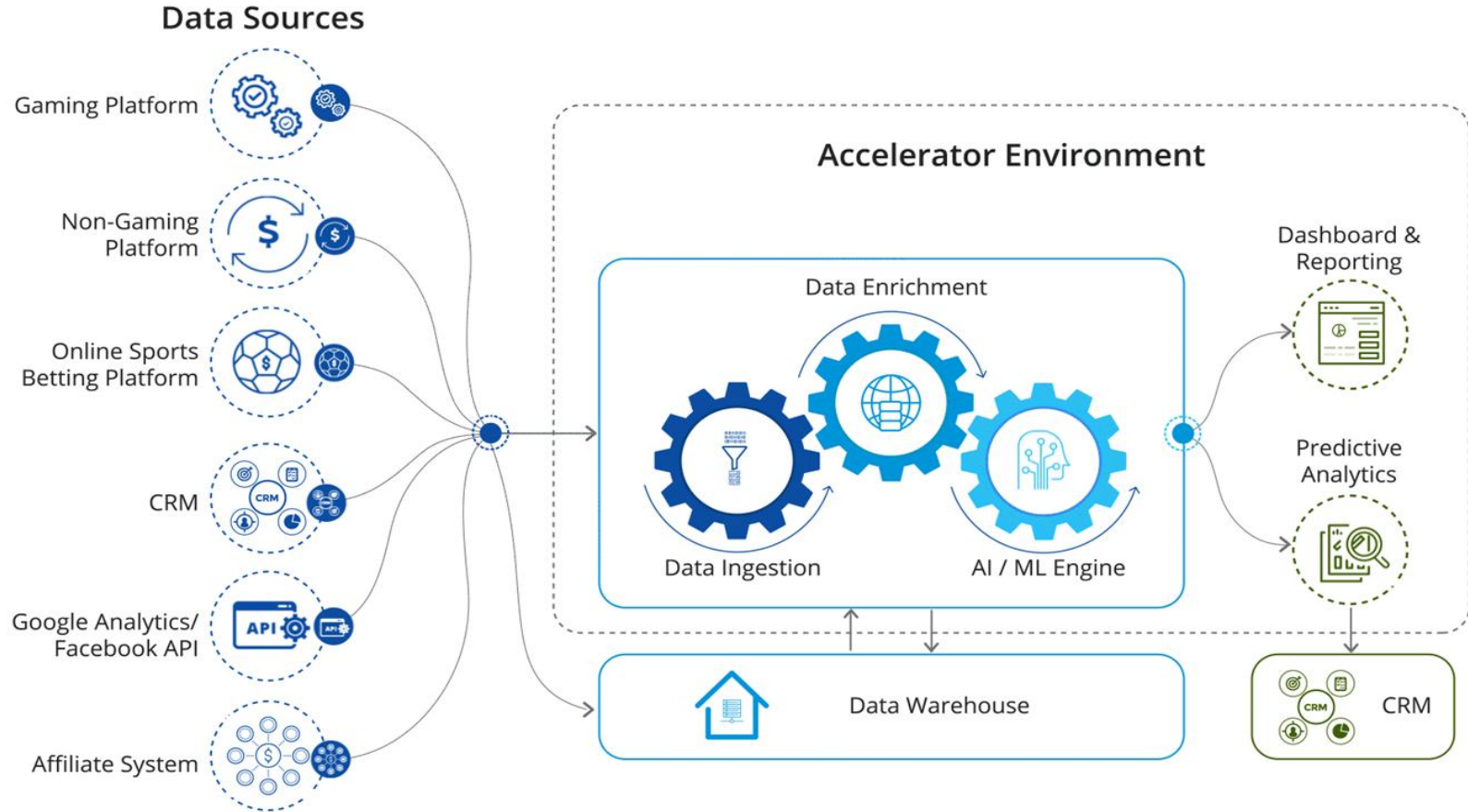


SEMI-STRUCTURED



UN-STRUCTURED

Processing of Various form of Data using Data Accelerator tools in cloud computing



5 VS of Bigdata

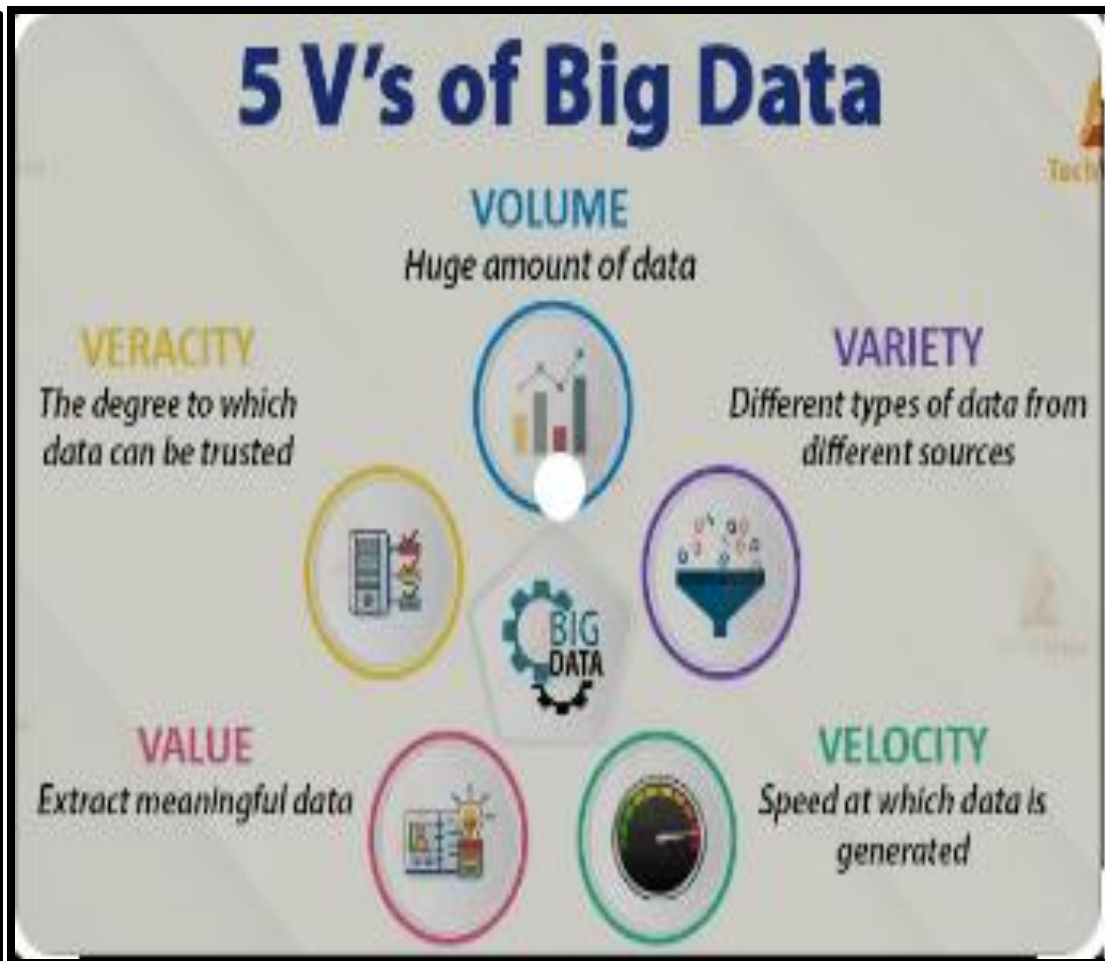
1.Volume - The volume of data refers to the size of the data sets that need to be analyzed and processed.

2.Variety – Big Data comes from a great variety of sources in form of structured, semi structured and unstructured data.

3.Velocity - Velocity refers to the speed with which data is generated.

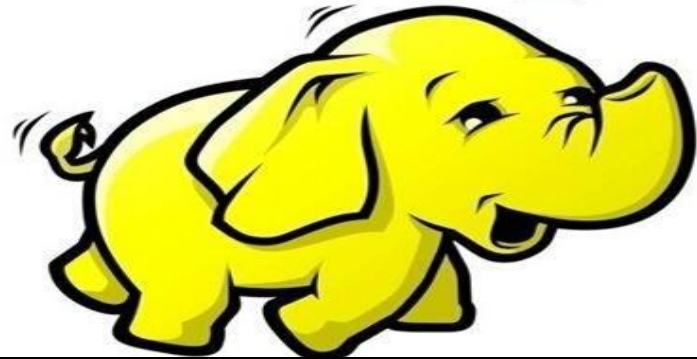
4.Veracity - Veracity refers to the quality of the data.

5. Value – Useful data extraction and their valubility from the data ware house.



Apache Hadoop as a solution

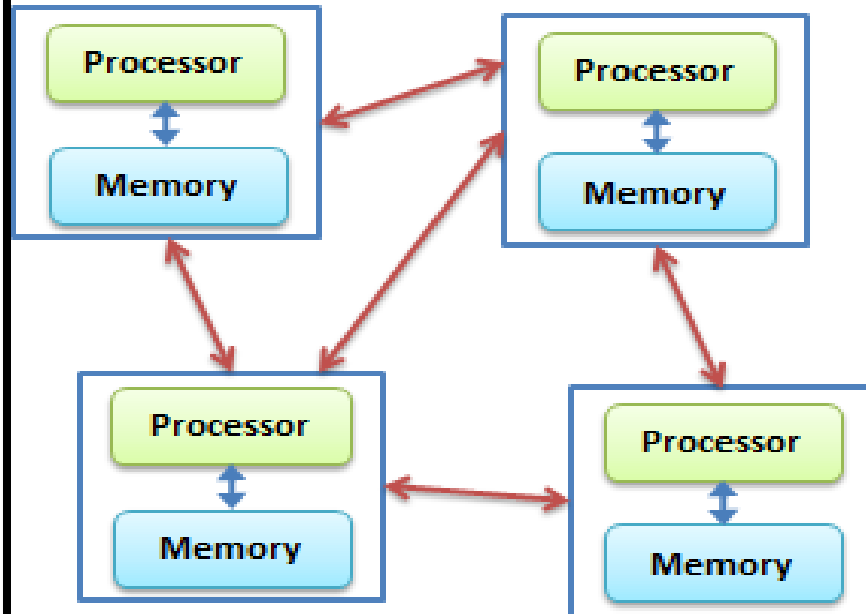
hadoop



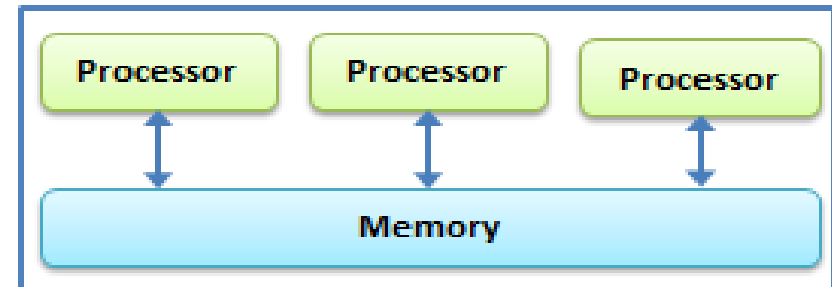
Parallel vs Distributed Computing

Parallel computing allows multiple processors to execute tasks simultaneously while distributed computing divides a single task between multiple computers to achieve a common goal.

Distributed Computing

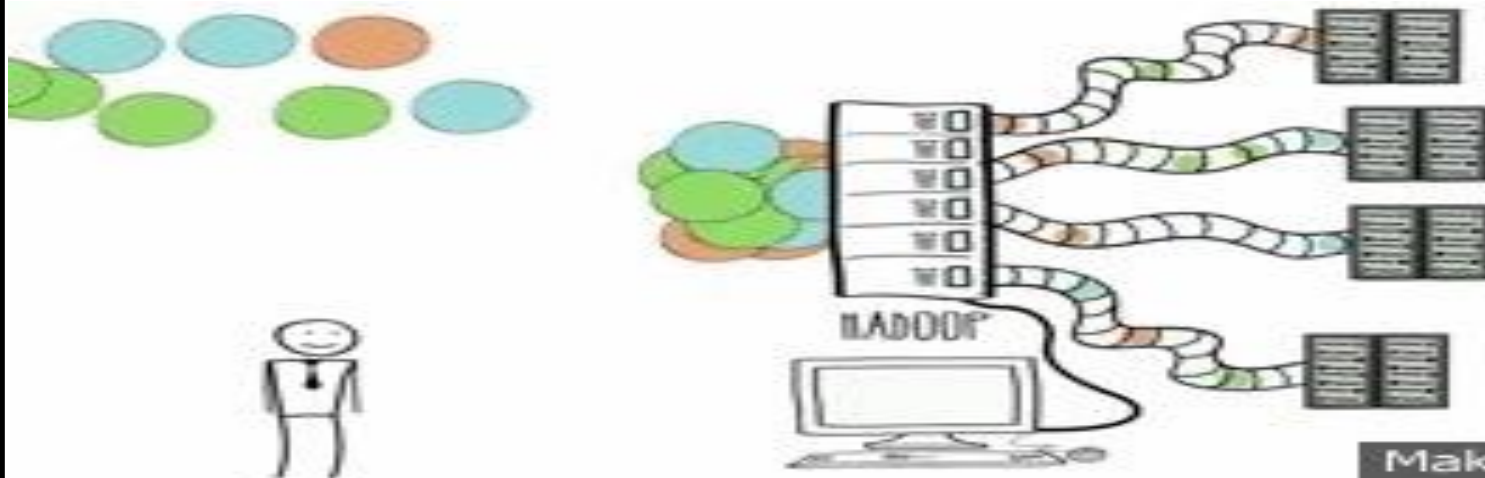


Parallel Computing

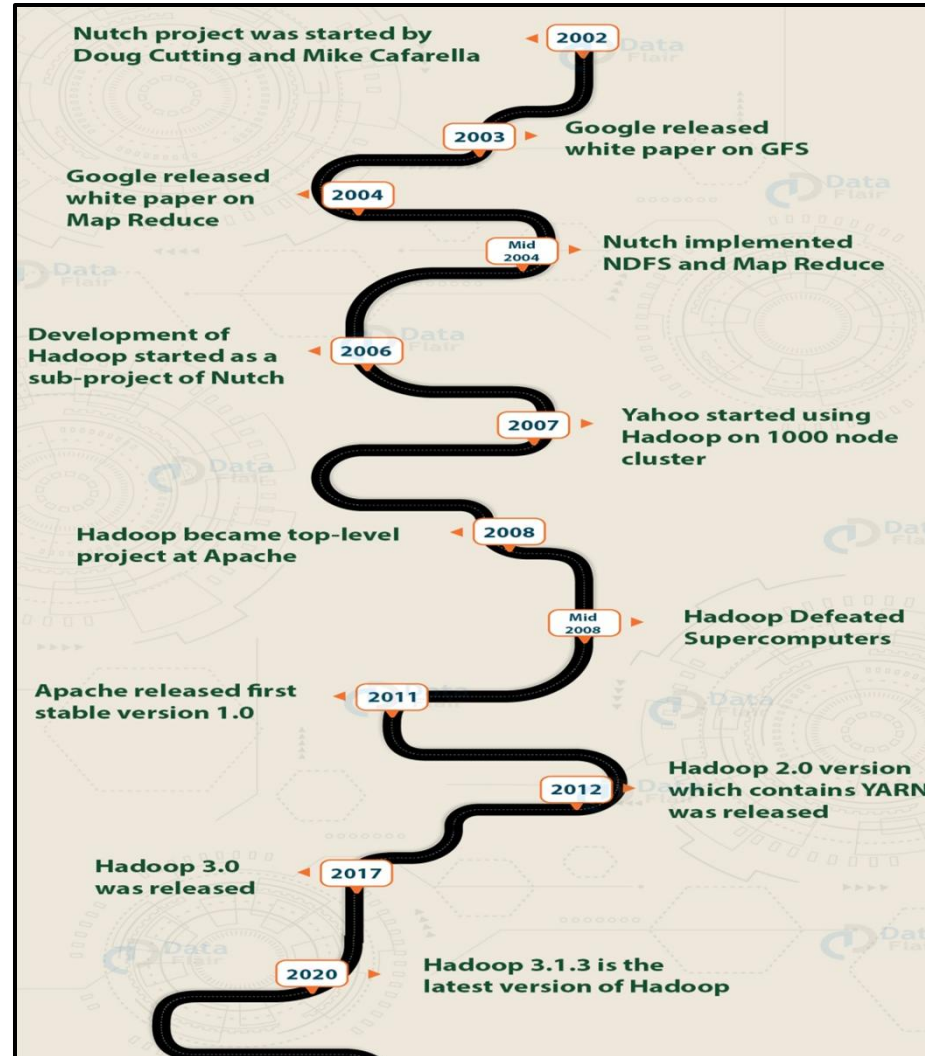


What is Hadoop ?

- Hadoop is an open source framework that is used to efficiently store and process large datasets.
- Instead of using one large computer to store and process the data, Hadoop allows clustering multiple computers to analyze massive datasets in parallel more quickly.



History and Timelines Evaluation of Hadoop



Tools to handle the Bigdata

- There are many tools that can be use to manage the Bigdata. Some of them are as follow –
 1. Apache Hadoop
 2. Apache Spark
 3. Google Big Query
 4. ApacheStorm
 5. Atlas.ti
 6. Apache **Cassandra**
 7. Hive
 8. Kaggle

Versions of Hadoop

HADOOP 1

MapReduce
(Resource Management +
Data Processing)

HDFS

HADOOP 2

MapReduce +
Other Types of Jobs

YARN
(Resource Management)

HDFS

Hadoop 3

Intel® Manager for Hadoop Software
Deployment, Configuration, Monitoring, Alerting and Security

Sqoop
Data Exchange

Flume
Log Collector

ZooKeeper
Coordination

Pig
Scripting

Hive
SQL-Like Query

HBase
Columnar Storage

MapReduce
Distributed Processing Framework

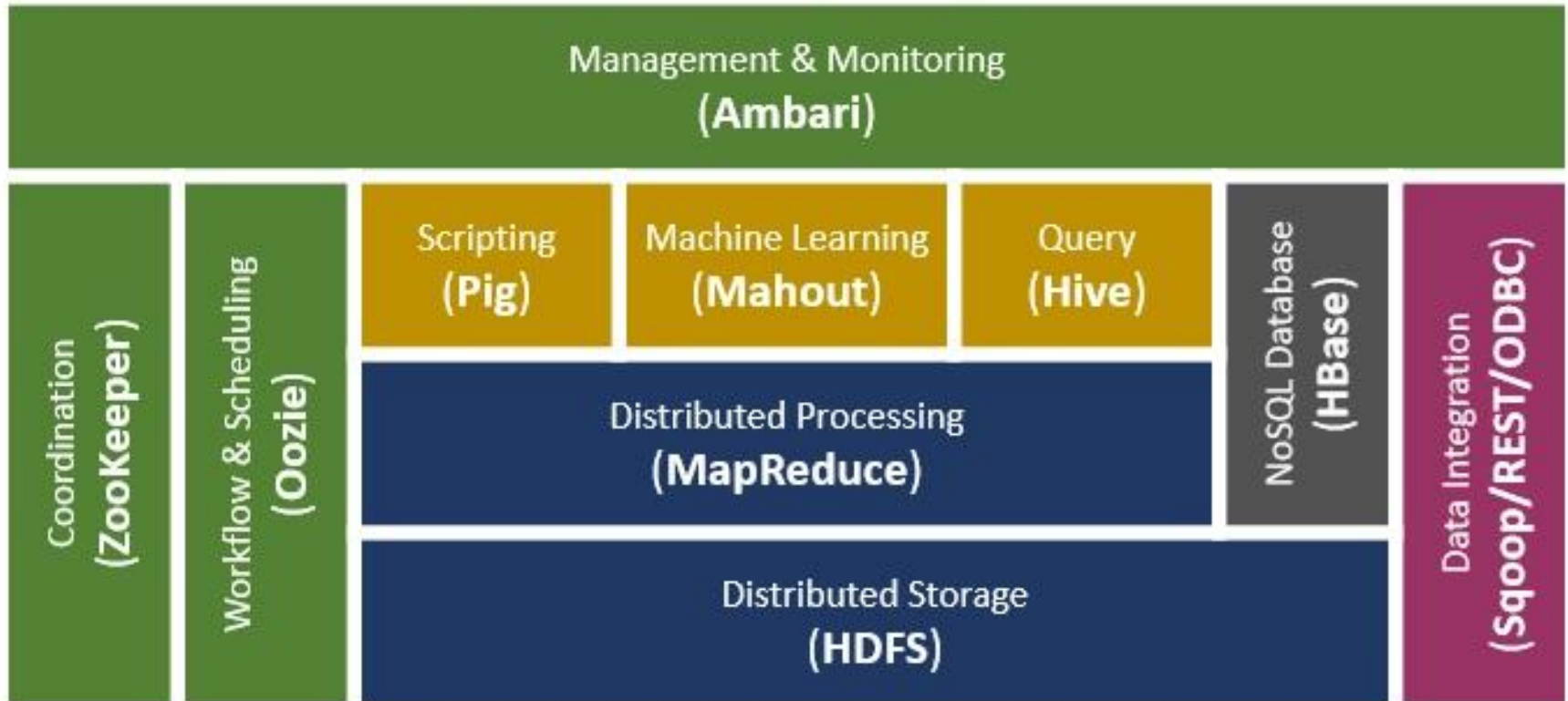
HDFS
Hadoop Distributed File System

Main differences between Hadoop 1 and Hadoop 2.

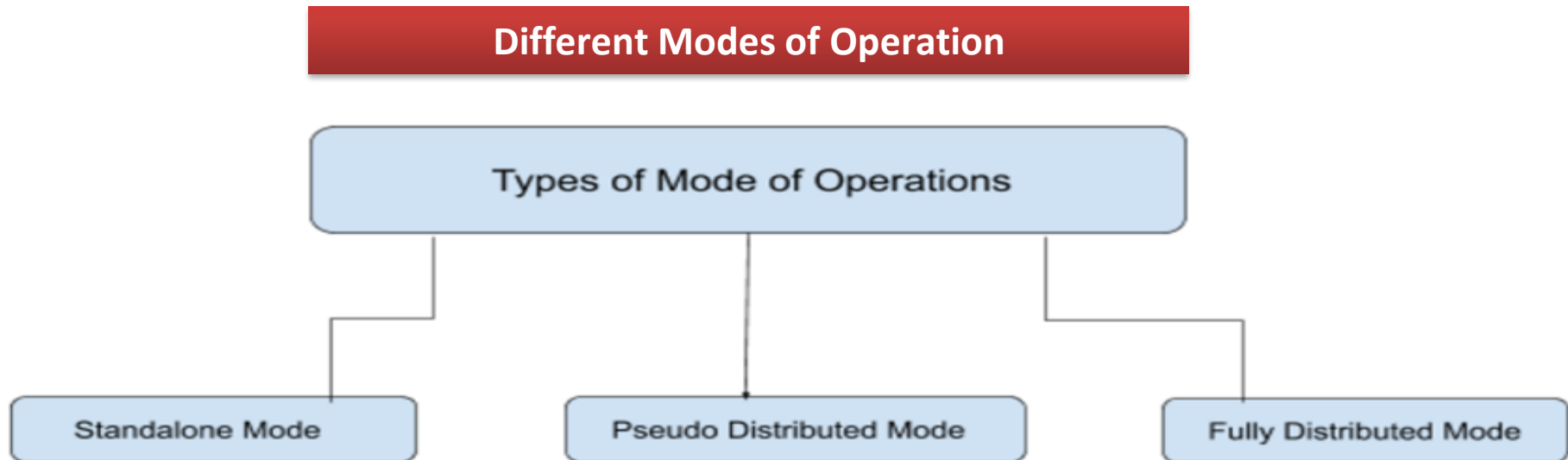
Sr. No.	Key	Hadoop 1	Hadoop 2
1	New Components and API	As Hadoop 1 introduced prior to Hadoop 2 so has some less components and APIs as compare to that of Hadoop 2.	On other hand Hadoop 2 introduced after Hadoop 1 so has more components and APIs as compare to Hadoop 1 such as YARN API, YARN FRAMEWORK, and enhanced Resource Manager.
2	Support	Hadoop 1 only supports MapReduce processing model in its architecture and it does not support non MapReduce tools.	On other hand Hadoop 2 allows to work in MapReducer model as well as other distributed computing models like Spark, Hama, Giraph, Message Passing Interface) MPI & HBase coprocessors.
3	Resource Management	Map reducer in Hadoop 1 is responsible for processing and cluster-resource management.	On other hand in case of Hadoop 2 for cluster resource management YARN is used while processing management is done using different processing models.
4	Scalability	As Hadoop 1 is prior to Hadoop 2 so comparatively less scalable than Hadoop 2 and in context of scaling of nodes it is limited to 4000 nodes per cluster	On other hand Hadoop 2 has better scalability than Hadoop 1 and is scalable up to 10000 nodes per cluster.
5	Implementation	Hadoop 1 is implemented as it follows the concepts of slots which can be used to run a Map task or a Reduce task only.	On other hand Hadoop 2 follows concepts of containers that can be used to run generic tasks.
	Windows Support	Initially in Hadoop 1 there is no support	On other hand with an advancement in version of Hadoop

Hadoop Architecture

Apache Hadoop Ecosystem



- Hadoop is an open-source framework which is mainly used for storage purpose and maintaining and analyzing a large amount of data or datasets on the clusters of commodity hardware.
- Actually a data management tool. Hadoop also possesses a scale-out storage property, which means that we can scale up or scale down the number of nodes as per a requirement in the future which is really a cool feature.

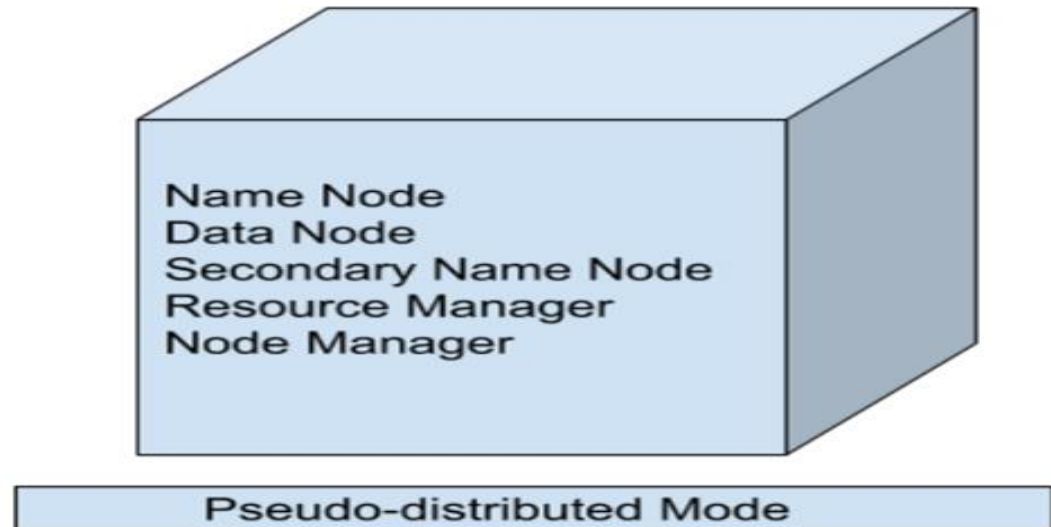


1. Standalone Mode

- Standalone Mode also means that we are installing Hadoop only in a single system.
- By default, Hadoop is made to run in this Standalone Mode or we can also call it as the *Local mode*.
- When Hadoop works in this mode there is no need to configure the files – *hdfs-site.xml*, *mapred-site.xml*, *core-site.xml* for Hadoop environment.
- In this Mode, all of your Processes will run on a single JVM(Java Virtual Machine) and this mode can only be used for small development purposes.
- Mainly in this Mode use Hadoop will be used for the Purpose of Learning, testing, and debugging.

2. Pseudo Distributed Mode (Single Node Cluster)

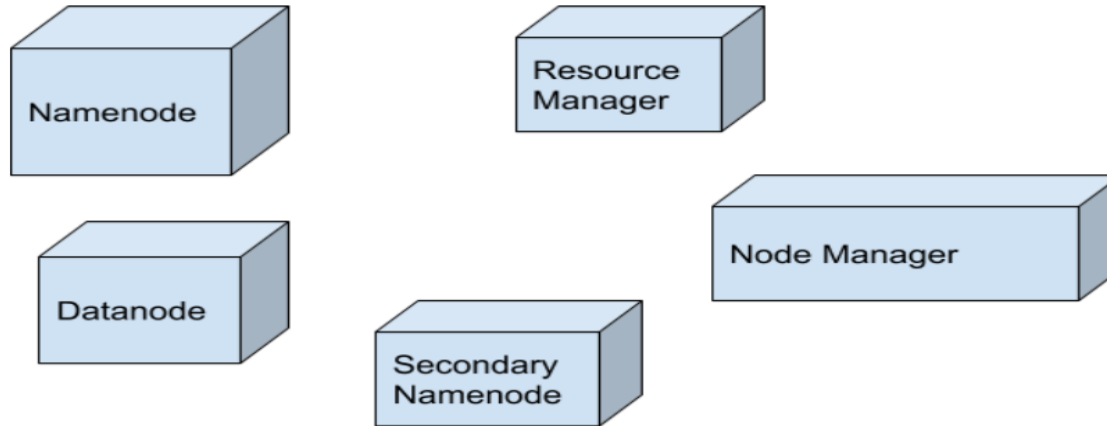
- In Pseudo-distributed Mode we also use only a single node, but the main thing is that the cluster is simulated, which means that all the processes inside the cluster will run independently to each other.
- All the daemons Namenode, Datanode, Secondary Name node, Resource Manager, Node Manager, etc. will be running as a separate process on separate JVM.
- we are using only the single node set up so all the Master and Slave processes are handled by the single system



- A secondary name node is also used as a Master. The purpose of the Secondary Name node is to just keep the hourly based backup of the Name node.
- In this Mode, Hadoop is used for development and for debugging purposes both. Our HDFS (Hadoop Distributed File System) is utilized for managing the Input and Output processes.
- Need to change the configuration files *mapred-site.xml*, *core-site.xml*, *hdfs-site.xml* for setting up the environment.

3. Fully Distributed Mode (Multi-Node Cluster)

- This is the most important mode in which multiple nodes are used few of them run on the **Master Daemon's** that are **Namenode** and **Resource Manager** and the rest of them run the **Slave Daemon's** that are **DataNode** and **Node Manager**. Hadoop will run on the clusters of Machine or nodes.
- Here the data that is used is distributed across different nodes. This is actually the *Production Mode* of Hadoop let's clarify or understand this Mode in a better way in Physical Terminology.



Features of Hadoop

- Hadoop is an open source software framework.
- Hadoop supports distributed storage and processing of huge amount of data set.

Key features of Hadoop-

1. **Fault tolerance** – Ability of a system to work continue in case of any components failure condition.
2. **Reliability** – HDFS store data in cluster and then divide the data in blocks. Hadoop framework store these block on nodes.
3. **High Availability**- Hadoop ensures the availability of the Hadoop cluster without any downtime.
4. **Hadoop is Open Source** –
5. **Cost-Effective** – Can run on commodity hardware.
6. **Faster in Data Processing** – Processed data in distributed nodes on a cluster.

Hadoop Ecosystem & its Components

- Hadoop Ecosystem is **a platform or a suite which provides various services to solve the big data problems**. It includes Apache projects and various commercial tools and solutions.

Hadoop consists of three main modules:

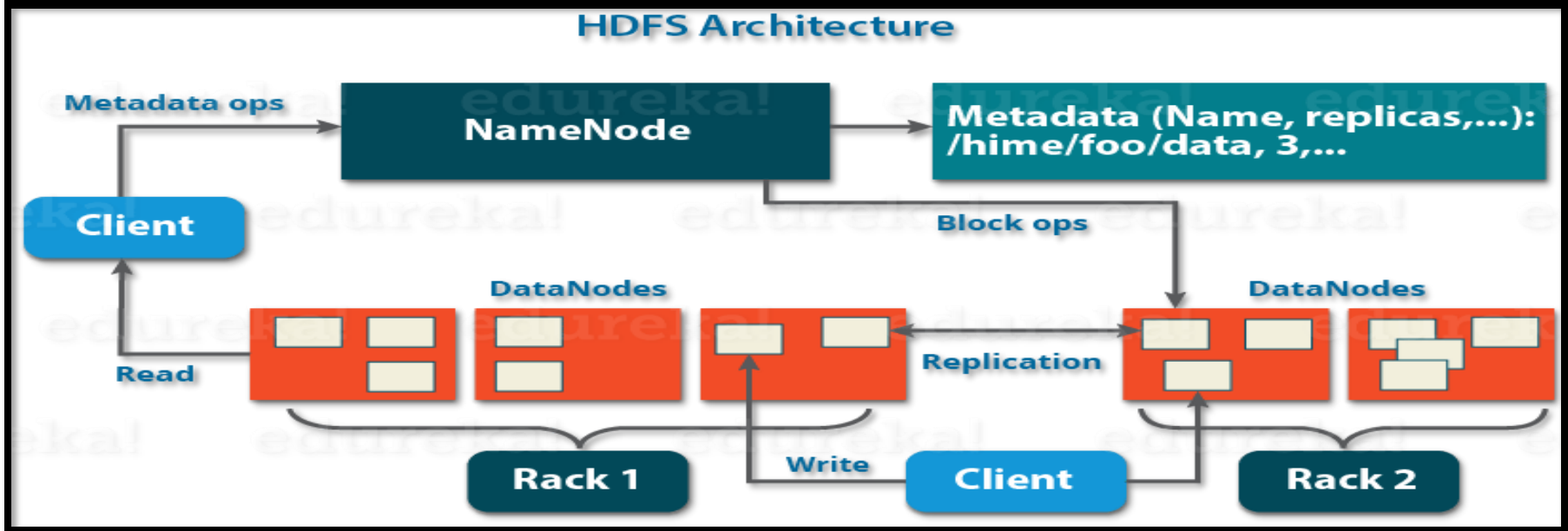
- 1. Hadoop Distributed File System (HDFS)**
- 2. Yet Another Resource Negotiator (YARN)**
- 3. Map Reduce**

1. HDFS

(Hadoop Distributed File System)

Is HDFS a database?

- No, it is just a **storage model**. HDFS store files which is used for processing.
- HDFS is a distributed file system that handles large data sets running on commodity hardware.
- HDFS follows the **master-slave** architecture.



HDFS

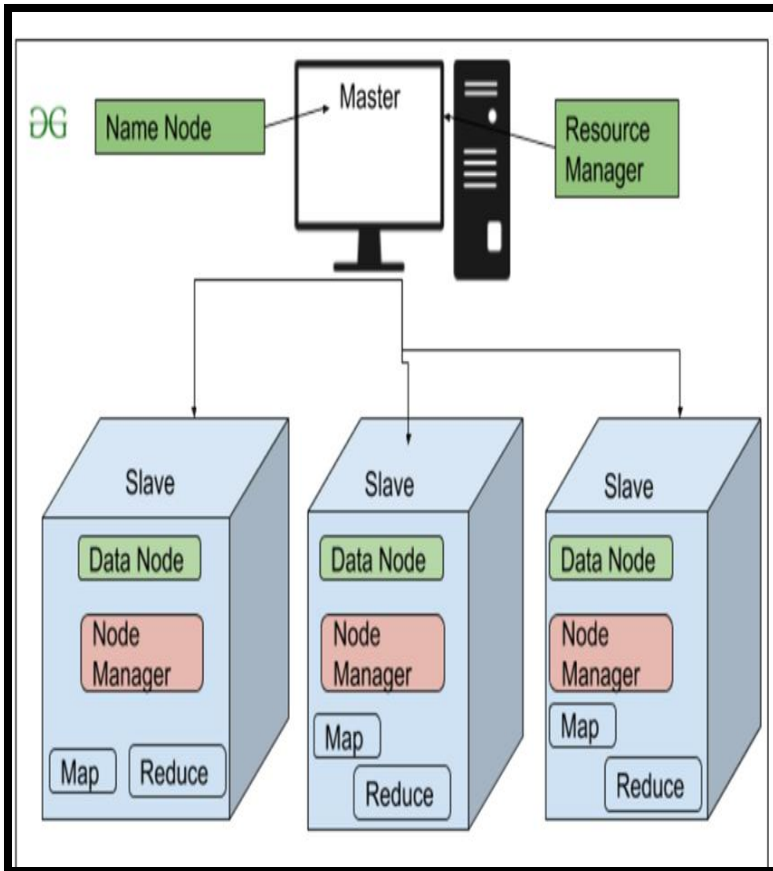
Master-Slave Architecture

Master-(Name Node)

The Master (Name Node) manages the file system namespace operations like opening, closing, and renaming files and directories and determines the mapping of blocks to Data Nodes along with regulating access to files by clients

Slave: {Data node}

Slaves (Data Nodes) are responsible for serving read and write requests from the file system's clients along with perform block creation, deletion, and replication upon instruction from the Master (Name Node).



Secondary Name Node(cont..)

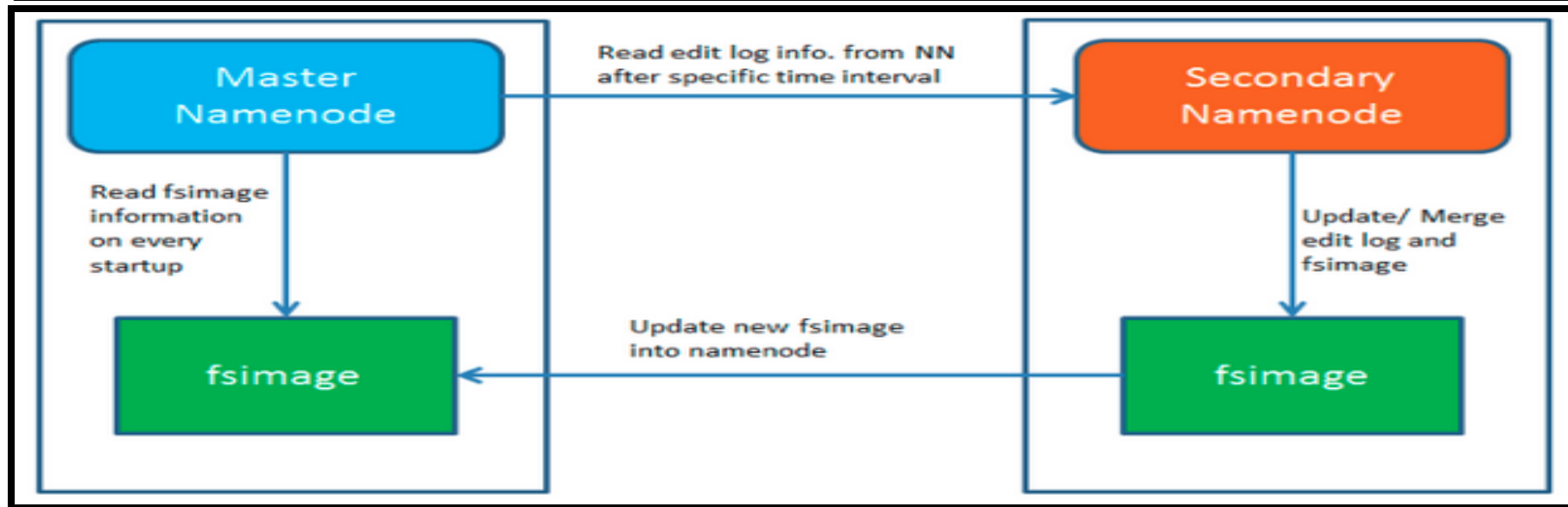
- **Secondary Name Node** – The main function is to observe the checkpoints of the file system metadata present on **name node**.
- It just checkpoints name node's file system namespace.
(fsimage and edit log files)
- When name node down, secondary node will be online but this node only has the **read access** to the fsimage and edit log files and **don't have the write** access to them.

Functions of Secondary Name Node-

1. Stores a copy of FsImage file and edits log.
2. Periodically applies edits log records to FsImage file & refreshes the edits log.
3. Check pointing of File system metadata is performed.

Secondary Name Node

- If **NameNode** fails, the **entire Hadoop cluster will fail**. Actually, there will be no data loss, only the cluster job will be shut down because Name Node is just the point of contact for all Data Nodes and if the Name Node fails then all communication will stop.
- The **secondary Name Node** merges the **fsimage** and the edits **log files** periodically and keeps edits log size within a limit.

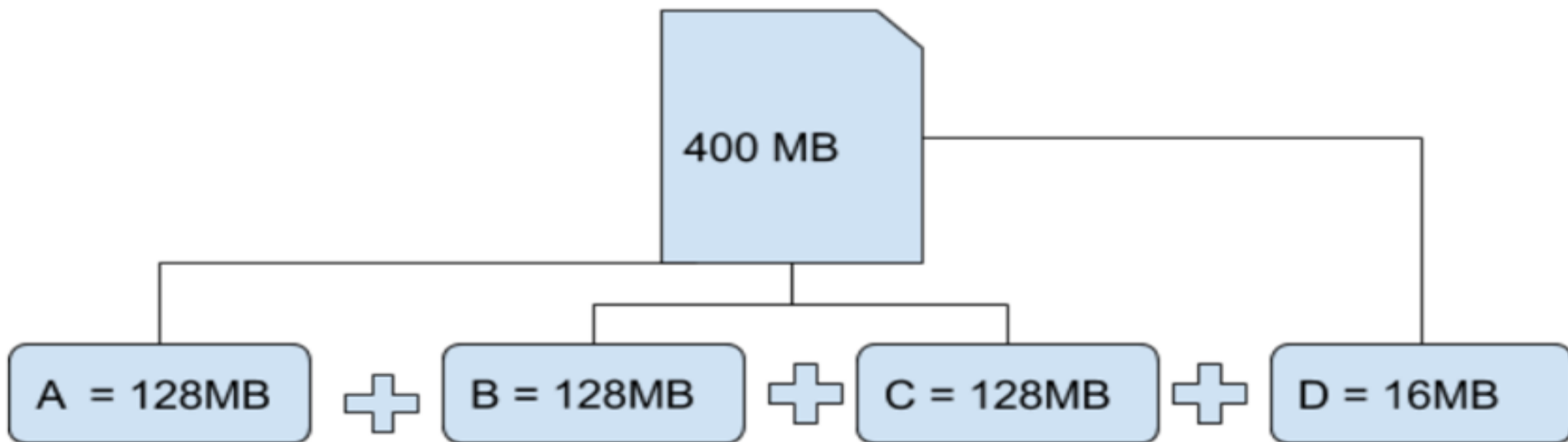


HDFS -Terminology to Store Data

File Block In HDFS-

- Data in HDFS is always stored in terms of **blocks**.
- Single block of data is divided into multiple blocks of size **128MB** which is **default**.

Data Blocks In Hadoop HDFS



Example- HDFS Terminology to Store Data

File Block In HDFS-

- Data in HDFS is always stored in terms of **blocks**.
- Single block of data is divided into multiple blocks of size **128MB** which is **default**.

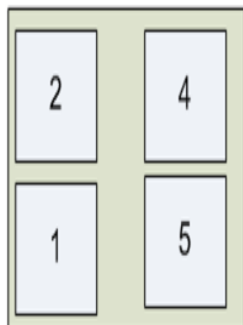
Files broken into blocks and replicated:

METADATA:

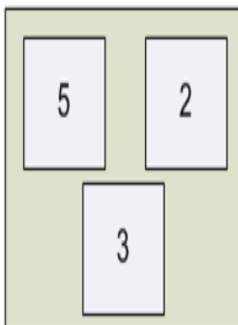
File 'F1' -> Blocks 1,2,3

File 'F2' -> Block 4,5

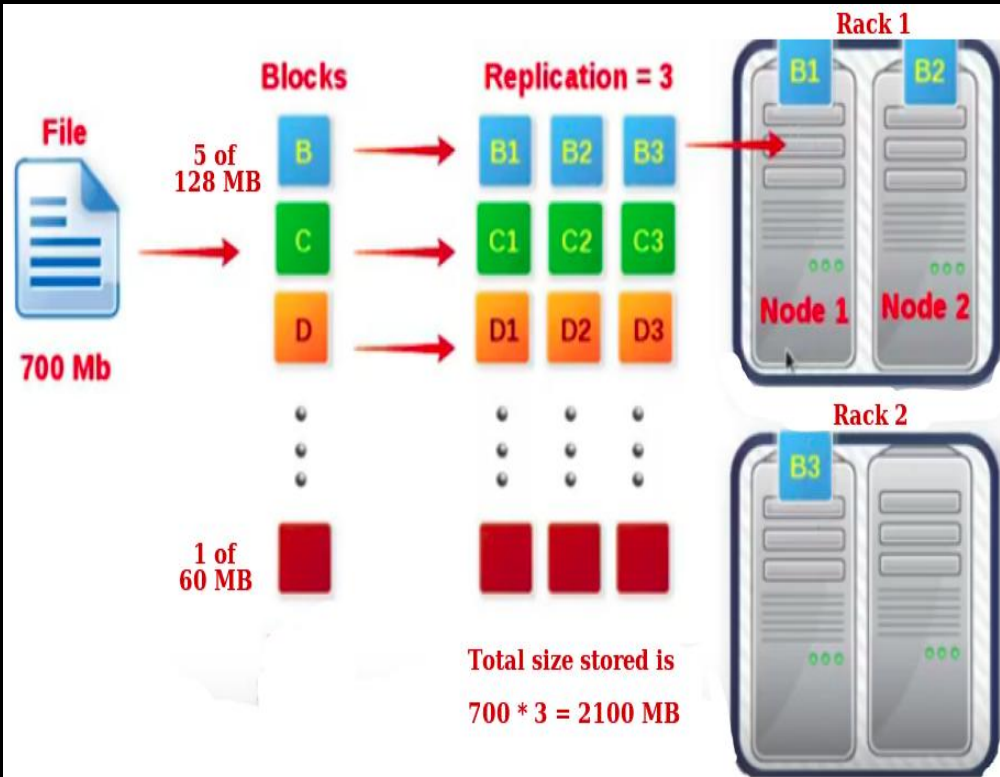
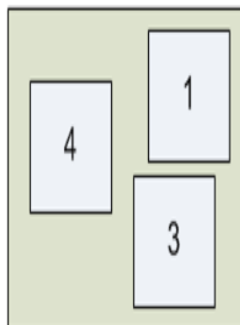
Datanode1



Datanode2



Datanode3



Replication and Rack awareness

Replication in HDFS - Ensures the availability of the data. Replication is making a copy of data node.

- By default, the **Replication Factor for Hadoop is set to 3** which can be configured or change as per our requirements manually .

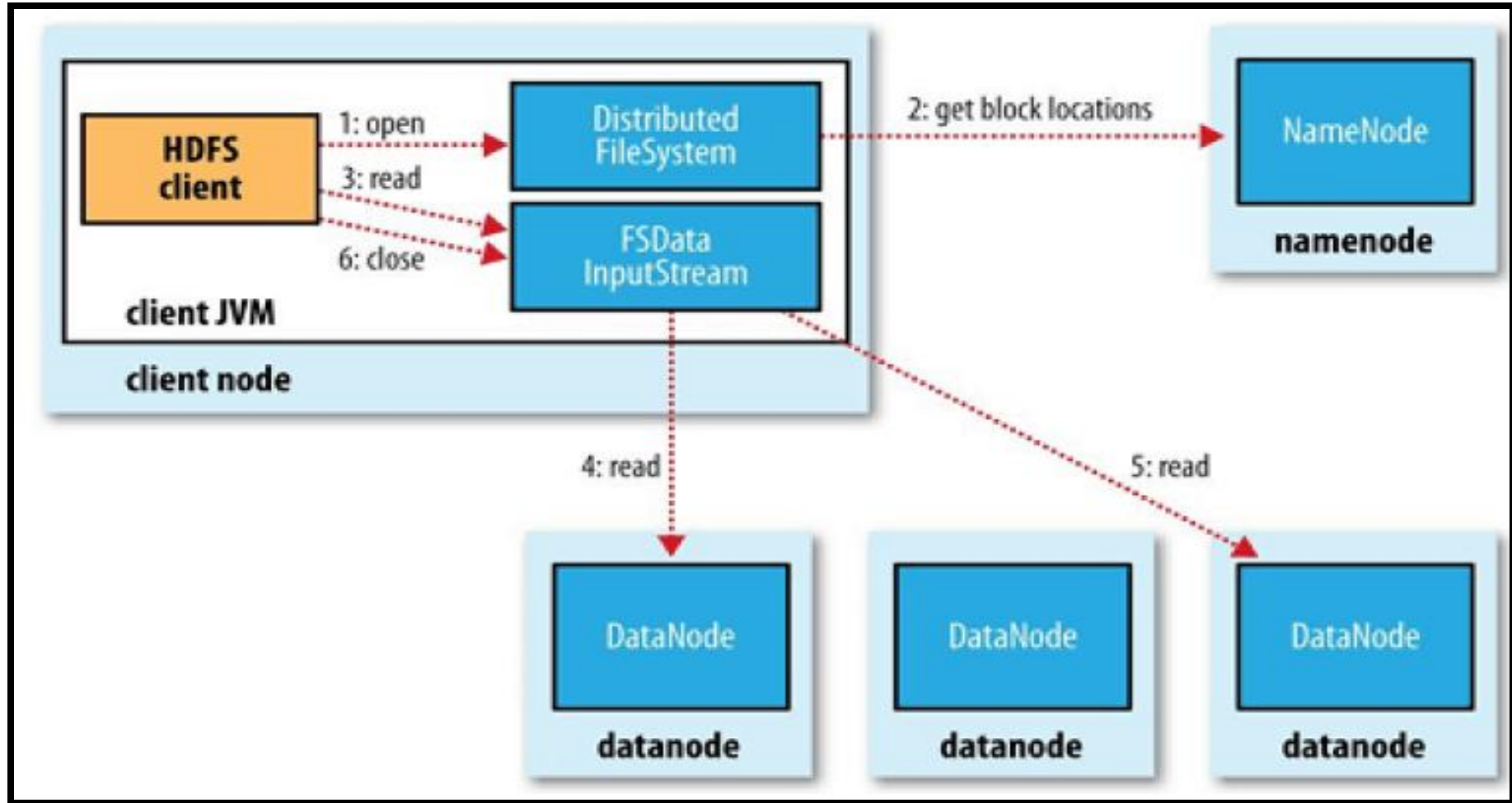
Example- if We have 4 file blocks which means that 3 Replica or copy of each file block is made means total of $4 \times 3 = 12$ blocks are made for the backup purpose.

Rack Awareness

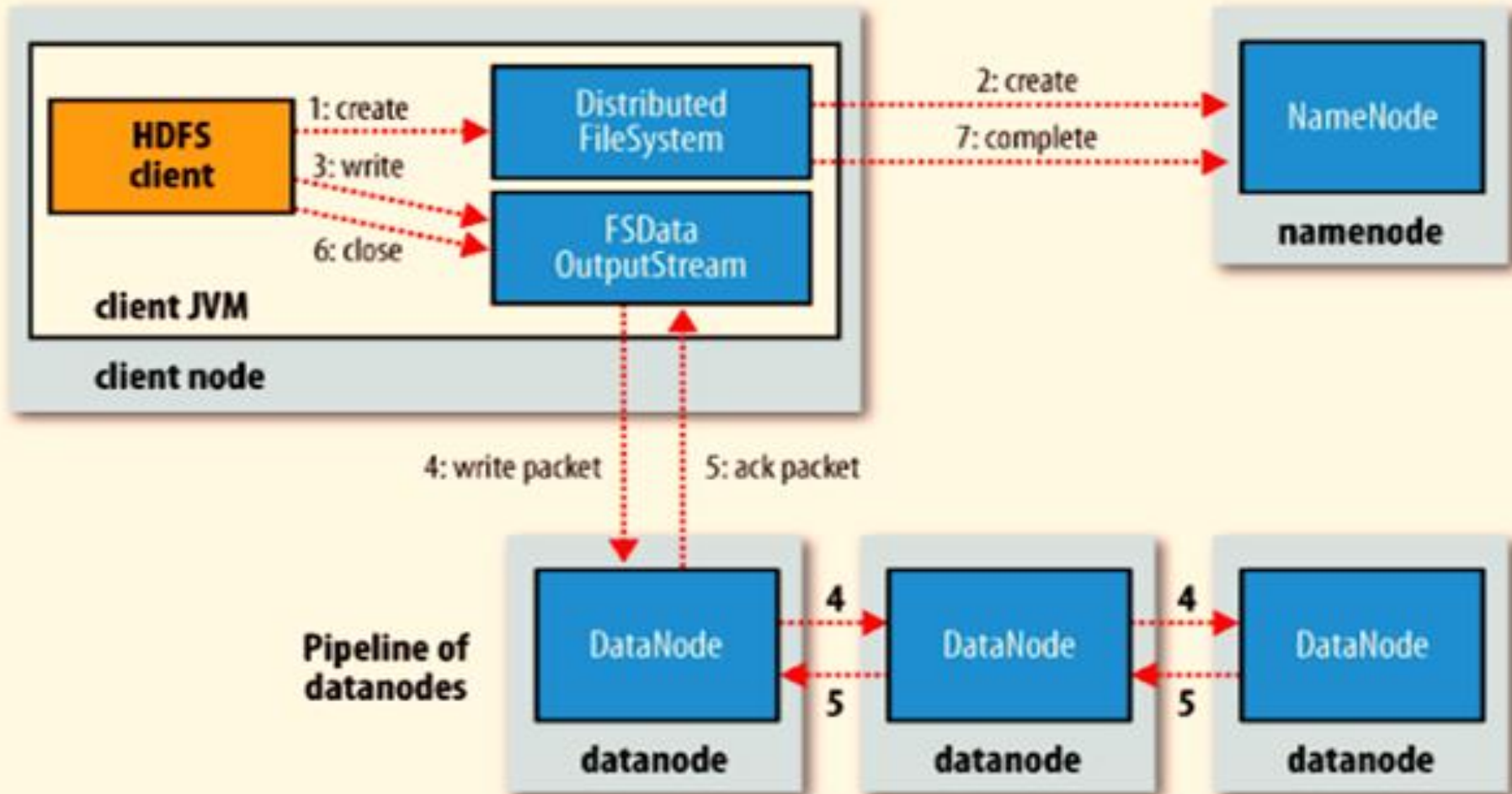
The rack is nothing but just the physical collection of nodes in our Hadoop cluster.

- A large Hadoop cluster is consists of so many Racks. with the help of this Racks information, Name node chooses the closest Data node to achieve the maximum performance while performing the read/write information which reduces the Network Traffic.

Read Operation in HDFS



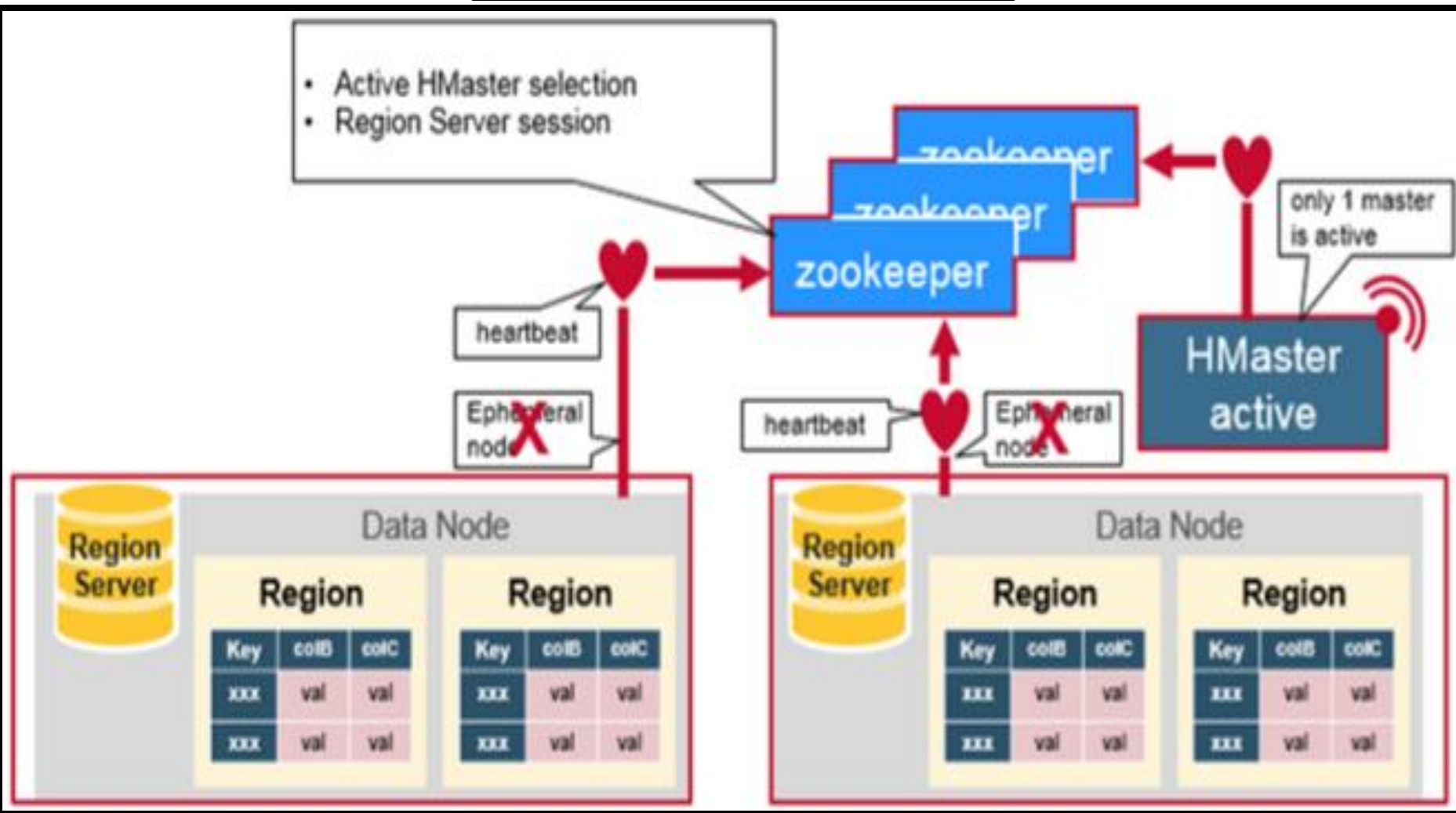
Write Operation in HDFS



What is Heartbeat in HDFS?

- In Hadoop framework heartbeat is a signal that is sent by Data node to Name node. Data node sends the signal to Name Node periodically indicating that it is alive.
- **Default interval for the heartbeat is 3 seconds in the Hadoop framework. If the Data node does not send the heartbeat signal to Name Node for more than 10 minutes, it is considered to be dead or unavailable. “This condition is known as ‘Heartbeat Lost’ condition”.**
- Name node knows about this condition by finding out that Heartbeat message is absent. Name Node marks the data node as dead and does not forward any Input/output request to it.
- **Data is replicated in HDFS in one of the below conditions-**
 - A. Data node becomes Unavailable
 - B. HDFS Replica of data becoming corrupted
 - C. Failure of Hard Disk in Data node.
 - D. Increase of replication factor in the settings

Heartbeat Broken - Example



Benefits of HDFS

- **Fault tolerance**-Detect faults and automatically recover quickly ensuring continuity and reliability.
- **Speed**- because of its cluster architecture, it can maintain 2 GB of data per second.
- **Access to more types of data**- specifically Streaming data because of its design to handle large amounts of data for batch processing.
- **Compatibility and Portability**-HDFS is designed to run on variety of hardware setups and compatible with several underlying operating systems Scalable.
- **Scalable**- You can scale resources according to the size of your file system. HDFS includes vertical and horizontal scalability mechanisms.

HDFS COMMANDS

1. **sudo jps** - command to see all Hadoop daemons.
2. **hadoop fs** - List all the Hadoop file system shell commands
3. **hadoop fs -help** ask for help!
4. **hadoop version** - Print the Hadoop version
5. **hadoop fsck / -files** – blocks will list the blocks that make up each file in the filesystem
hadoop fs -ls / - List the contents of the root directory in HDFS
6. **hadoop fs -df hdfs:/** - Report the amount of space used and available on currently mounted filesystem.
7. **hadoop fs -count hdfs:/** - Count the number of directories, files and bytes under the paths that match the specified file pattern
8. **hadoop fs -mkdir /sample** - Create a directory in hdfs named as sample.
9. **hadoop fs -put /home/cloudera/sample.txt /sample** - Add a sample text file from the local directory named as "sample.txt" to the new directory you created in HDFS.

10.hadoop fs -ls /sample - List the contents of this new directory in HDFS.

11.hadoop fs -put /home/cloudera/abc.txt hdfs:/ -Command to put a file from local file system directly in hdfs.

12. Hadoop fs -cat /abc.txt - command to see the contents of file.

13. Hadoop fs -rm /abc.txt - command to remove/delete a file from hdfs.

14. Hadoop fs -rm -r /sample - command to remove a directory.

15. Hadoop fs -get /abc.txt /home/cloudera/Desktop - command to put a file from hdfs to local file system.

16. Hadoop fs -mv /b.txt /sample - command to move one file from one location to another in hdfs. (both locations should be in hdfs)

17 du Hadoop fs -du /sampletest - will display all the storage space taken by all the files in a directory.

HDFS COMMANDS (Cont.....)

18 Hadoop fs -df - displays current disk usage by Hadoop distributed file system

19 cp Hadoop fs -cp /source location /destination location - copy files from one hdfs dir to other.

20 Hadoop fs -tail /home/cloudera/ - will show the content of the file whose path we give

21 -touchz : Used to create an empty file at the specified location.

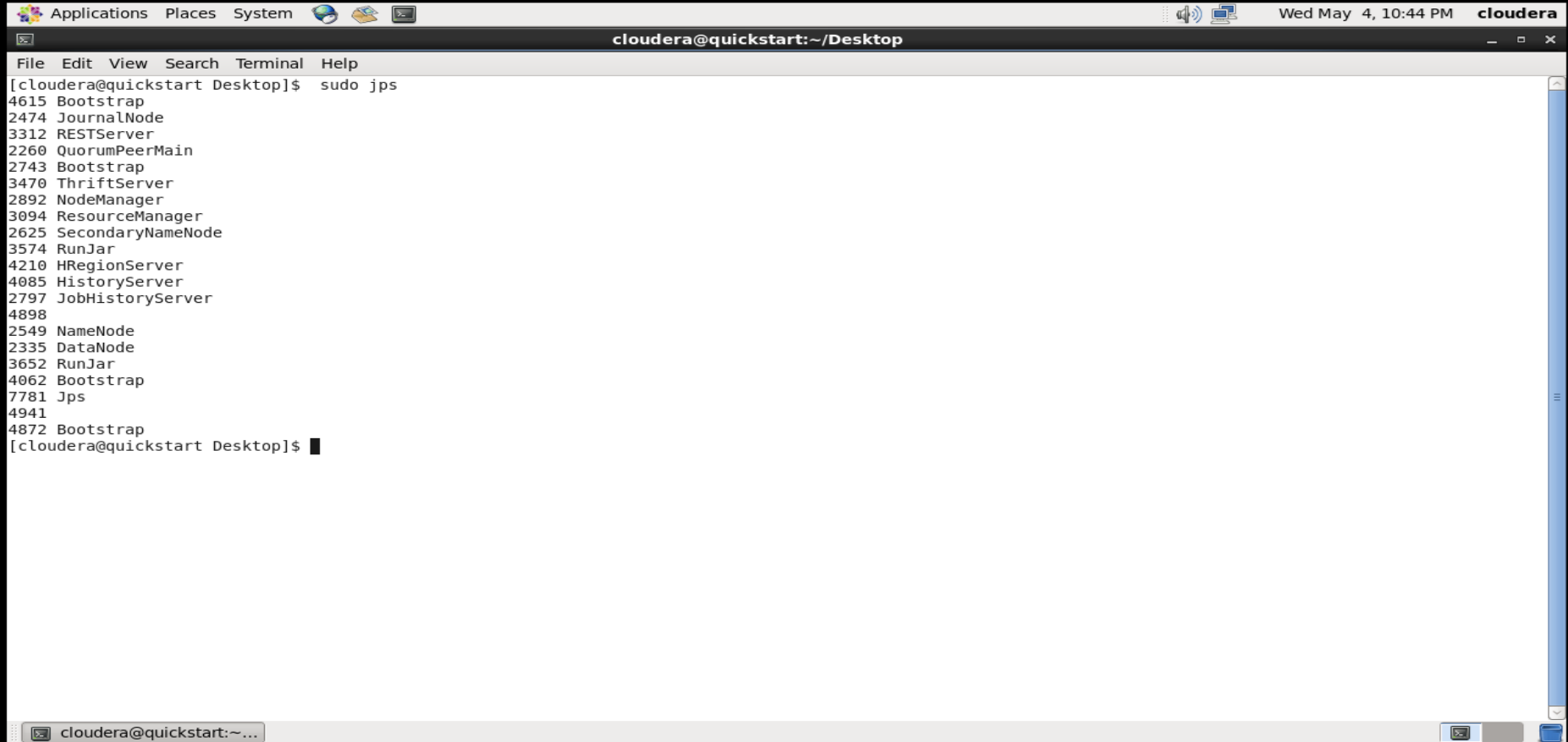
hadoop fs -touchz /HDFS_location_of_directory/empty_filename

hadoop fs -touchz /empty_filename

22. Hadoop fs - copyFromLocal /source_at_local /destination_at_hdfs

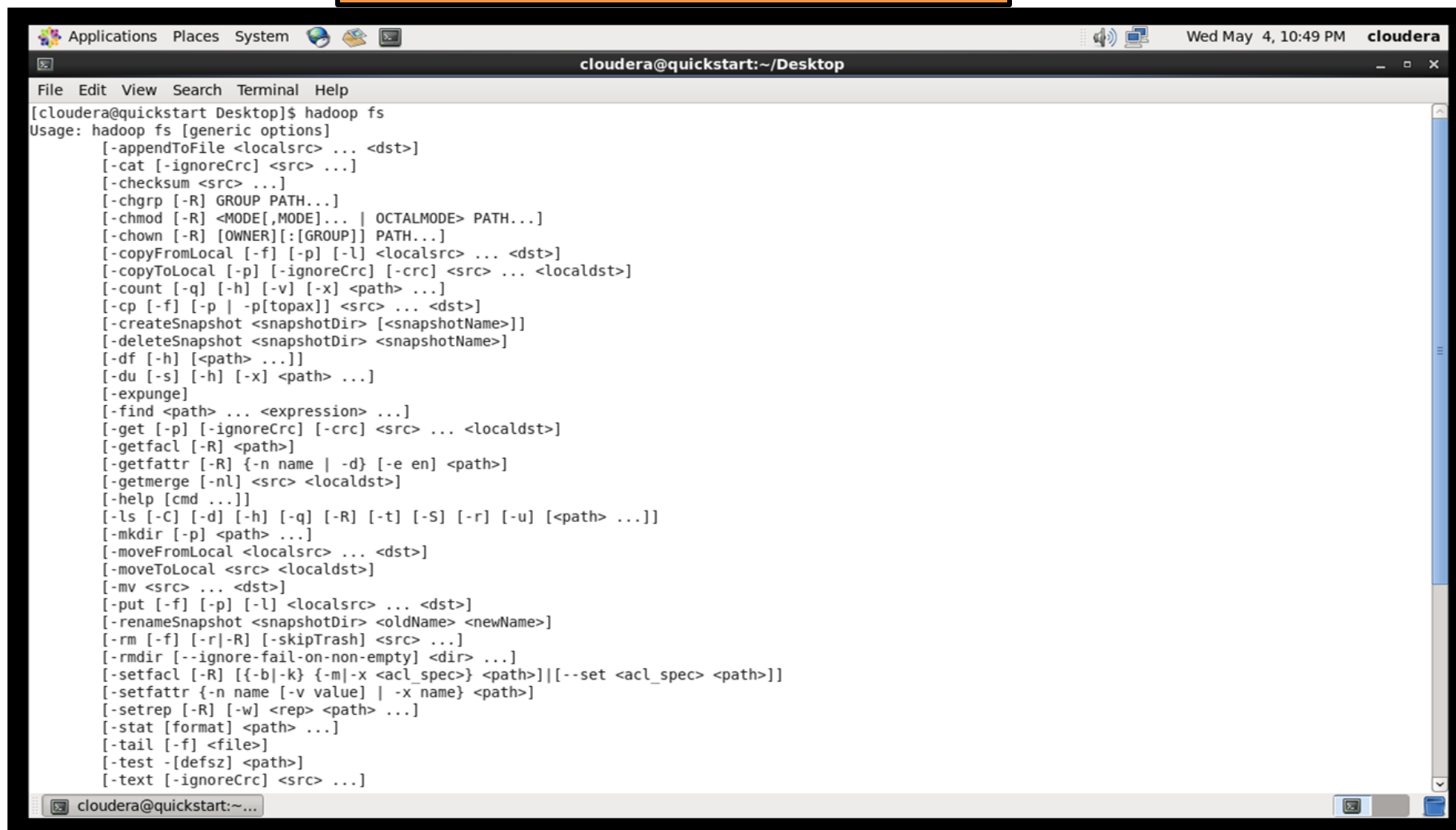
Example - HDFS Commands

1. Sudo jps - Command to see all Hadoop daemons



```
cloudera@quickstart:~/Desktop
File Edit View Search Terminal Help
[cloudera@quickstart Desktop]$ sudo jps
4615 Bootstrap
2474 JournalNode
3312 RESTServer
2260 QuorumPeerMain
2743 Bootstrap
3470 ThriftServer
2892 NodeManager
3094 ResourceManager
2625 SecondaryNameNode
3574 RunJar
4210 HRegionServer
4085 HistoryServer
2797 JobHistoryServer
4898
2549 NameNode
2335 DataNode
3652 RunJar
4062 Bootstrap
7781 Jps
4941
4872 Bootstrap
[cloudera@quickstart Desktop]$
```

2. Hadoop fs - List all the Hadoop file system shell commands



The screenshot shows a terminal window titled "cloudera@quickstart:~/Desktop". The terminal displays the command "hadoop fs" and its usage information. The usage information lists various Hadoop fs commands and their options, including file operations, permissions, and directory management.

```
cloudera@quickstart Desktop]$ hadoop fs
Usage: hadoop fs [generic options]
    [-appendToFile <localsrc> ... <dst>]
    [-cat [-ignoreCrc] <src> ...]
    [-checksum <src> ...]
    [-chgrp [-R] GROUP PATH...]
    [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
    [-chown [-R] [OWNER][[:GROUP]] PATH...]
    [-copyFromLocal [-f] [-p] [-l] <localsrc> ... <dst>]
    [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
    [-count [-q] [-h] [-v] [-x] <path> ...]
    [-cp [-f] [-p | -p[topax]] <src> ... <dst>]
    [-createSnapshot <snapshotDir> [<snapshotName>]]
    [-deleteSnapshot <snapshotDir> <snapshotName>]
    [-df [-h] [<path> ...]]
    [-du [-s] [-h] [-x] <path> ...]
    [-expunge]
    [-find <path> ... <expression> ...]
    [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
    [-getfacl [-R] <path>]
    [-getfattr [-R] {-n name | -d} [-e en] <path>]
    [-getmerge [-nl] <src> <localdst>]
    [-help [cmd ...]]
    [-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [<path> ...]]
    [-mkdir [-p] <path> ...]
    [-moveFromLocal <localsrc> ... <dst>]
    [-moveToLocal <src> <localdst>]
    [-mv <src> ... <dst>]
    [-put [-f] [-p] [-l] <localsrc> ... <dst>]
    [-renameSnapshot <snapshotDir> <oldName> <newName>]
    [-rm [-f] [-r|-R] [-skipTrash] <src> ...]
    [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
    [-setfacl [-R] [{-b|-k} {-m|-x acl_spec} <path>]|[--set acl_spec <path>]]
    [-setfattr {-n name [-v value] | -x name} <path>]
    [-setrep [-R] [-w] <rep> <path> ...]
    [-stat [format] <path> ...]
    [-tail [-f] <file>]
    [-test -[defsz] <path>]
    [-text [-ignoreCrc] <src> ...]
```


Thank You

*Any
Questions?*