BIG DATA ANALYTICS - 18CS72

1. Raj Kamal and Preeti Saxena, “Big Data Analytics Introduction to Hadoop, Spark, and Machine-Learning”, McGraw Hill Education, 2018 ISBN: 9789353164966, 9353164966

2. Douglas Eadline, "Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem", 1st Edition, Pearson Education, 2016. ISBN13: 978-9332570351

Dr. Vaneeta M
Associate Professor
Dept. of CSE

Module 2
Chapter 2
Introduction to Hadoop

- **CO 2: Construct Hadoop framework components and Hadoop Distributed File system.**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision

- **“To create competent professionals in Computer Science and Engineering with adequate skills to drive the IT industry”.**

Mission

- **Impart sound technical knowledge and quest for continuous learning.**
- **To equip students to furnish Computer Applications for the society through experiential learning and research with professional ethics.**
- **Encourage team work through inter-disciplinary project and evolve as leaders with social concerns.**

Introduction

- A programming model is **centralised** computing of data in which the data is transferred from multiple distributed data sources to a **central server**.

- Analyzing reporting, visualizing, business-intelligence **tasks compute centrally. Data are inputs to the central server.**

- An enterprise collects and analyzes data at the **enterprise level**. The computations are at an enterprise server or data warehouse integrated with the applications.

- Applications running at the server does the following analysis:

1.Suggests a strategy for filling the machines at minimum cost of logistics

2.Finds locations of high sales such a gardens, playgrounds etc.

3. Finds days or periods of high sales such as Xmas etc.

4 Finds children's preferences for specific chocolate flavors

5 Finds the potential region of future growth

6 Identifies unprofitable machines

7. Identifies need of replicating the number of machines at specific locations.

- Another programming model is **distributed computing** that uses the databases at multiple computing nodes with data sharing between the nodes during computation.
- **Distributed computing** in this model requires
  - ***Cooperation (sharing) between the DBs in a transparent manner*** *- Each user within the system may access all the data within all databases as if they were a single database.*
  - ***Location independence-*** *Results should be independent of geographical locations.*

■ **EXAMPLE 2.1**

Consider a jigsaw Puzzle Ravensburger Beneath the Sea (5000 pieces) Children above 14 years of age will assemble the pieces in order to solve the puzzle. What will be the effect on time intervals for solution in three situations, when 4, 100 and 200 children simultaneously attempt the solution.

SOLUTION 1:

- Assume 4 children sit together and solve the puzzle by dividing the tasks.
- Each child assembles one-fourth part of the picture for which they pick the pieces from a common basket
- **Distributed computing and centralized data model**

Solution 2:

- Each child assembles one-fourth part of the picture for which the pieces are distributed in four baskets.
- The child in case does not find a piece in his/her basket, then searches for it in another basket
- **Distributed databases and distributed computing tasks with data sharing model**

- Partitioning of assembling jobs into four has an issue of time.
- A child may **complete his/her part much late** than the remaining children. Beneath-the-sea portion is too complex, while upper-depth-sea portion just plain.
- The children **combine all four parts and finally complete the puzzle**.
- Each one has to **look into the other three parts** to find a match and complete the task.
- Time taken to solve the puzzle is **[$T/4+T_I(4)+T_C(4)$].**
  - *where $T_I(4)$ is the time taken in **seeking from others** the pieces not available to a child during intermediate phases,*
  - *and $T_C(4)$ in **combining the results** of the four children.*
  - ***Scaling factor is slightly less than 4. The proposed distributed model works well.***

Solution for 100 children

- Assume a second situation in which 100 children assemble their parts of 50 pieces each, and finally combine and complete the puzzle.
- Each child must seek a piece, not available with her/him daring the intermediate phase.
- Combining also becomes difficult and a time-consuming exercise compared to the four children case because each child now matches the results with the remaining 99 counterparts to arrive at the final solution.
- The time taken to solve the puzzle is $[T/100 + T_I (100) + T_C (100)]$
- Scaling is by factor less than 100.
- The distributed model has issues like sharing pieces, seeking pieces not available and combining issues. Issues are at the intermediate aw well as at the end stages.

- **Solution for 200 Children**
- If 200 children attempt to solve the puzzle simultaneously at the same time then finally combining all 200 portions of the Beneath the Sea, the integration of 200 portions will be tedious and will be a far more time-consuming exercise than with 4 or 100,
- The time taken to solve the puzzle is [**T/200+T, (200) + Tc (200)],**
- Scaling up is by factor much less than 200 and may even be less than even 100.
- The distributed model with pieces sharing between the children is unsatisfactory because T, (200)+Tc (200)<T/200.
- Problem of inter-children interactions exponentially grows with the number of children in the proposed distributed model with seeking pieces in intermediate phases.
- Time TI becomes significantly high

■ Final Solution

- Alternatively, the picture parts and corresponding pieces of each part distribute to each participating child distinctly
- **Distributed computing model with no data sharing.**
- Time TI taken in seeking a piece from available with him/her is zero.
- The time taken in joining the assembled picture portions is only at the end.
- Problem of inter-children interactions during solving the puzzle does not exist.

- Traditionally, a program when executes calls the data inputs.

1.Centralized computing model requires few communication overheads.

2.Distributed computing model requires communication overheads for seeking data from a remote source when not available locally, and arrive at the final result.

  - *The completion of computations will* ***take more and more time*** *when the number of distributed computing nodes increase.*

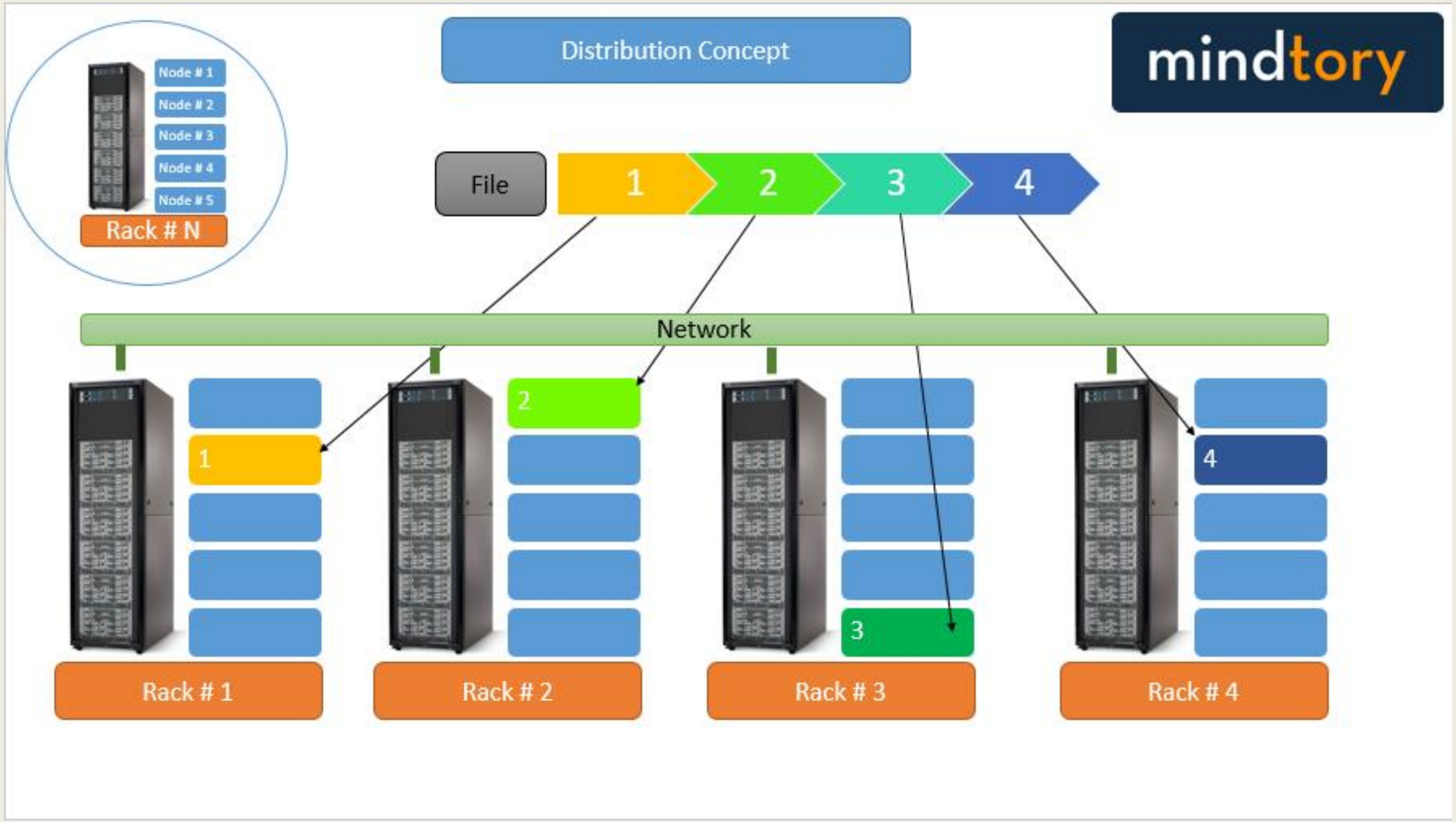
- The application tasks and datasets needed for computations distribute at a number of geographic locations and remote servers.

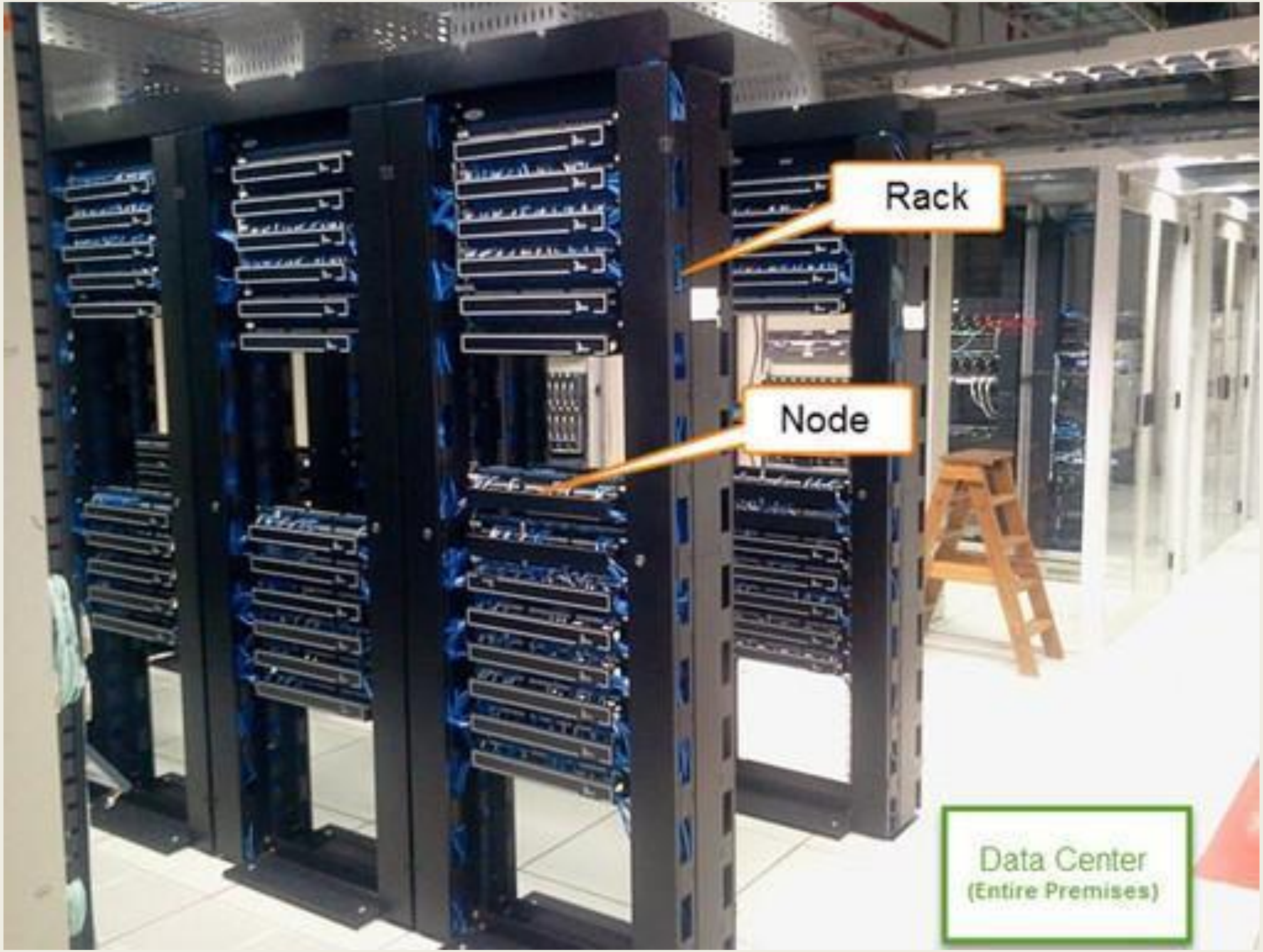
3. **Massive Parallel Processing MPP**s or computing clusters when datasets are too large.

  - *Application is* ***divided in number of tasks and sub-tasks****. The sub-tasks get inputs from* ***data nodes same cluster****. Results of sub-tasks aggregate and communicate to the application. The* ***aggregate*** *results from each cluster collect using APIs at the application.*

(i) Big Data Store Model

- **A model for Big Data store is as follows:**
- Data store in file system consisting of data blocks (physical division of data).
- The data blocks are distributed across multiple nodes.
- Data nodes are at the racks of a cluster. Racks are scalable.
- A Rack has multiple data nodes (data servers), and each cluster is arranged in a number of racks.




Rack
Node
Data Center
(Entire Premises)

(ii) Big Data Programming Model

- Big Data programming model is that application in which application jobs and tasks (or sub-tasks) is scheduled on the same servers which store the data for processing.
- **Job means** running an assignment of a **set of instructions** for processing.
  - *For example, processing the queries in an application and sending the result back to the application is a job.*
- Other example is instructions for sorting the examination performance data is a job.
- **Job scheduling means** assigning a job for processing following a schedule.
  - *For example, scheduling after a processing unit finishes the previously assigned job, scheduling as per specific sequence or after a specific period.*

- Hadoop system uses programming model, where jobs or tasks are assigned scheduled on the same server which hold the data.
- Hadoop is one of widely technologies. Google and Yahoo use Hadoop.
- Hadoop creators created cost-effective method to search indexes. Facebook, Twitter, and LinkedIn use Hadoop.
- IBM implemented BigInsights and licensed Apache Hadoop.
- Oracle implement Hadoop system with Big Appliance, IBM Infosphere Microsoft with Big solutions

- **Following important terms and their meaning**
- **Cluster Computing** refers computing, storing and analyzing huge amounts unstructured or structured data in distributed computing environment.
  - *Each cluster forms by a set loosely or tightly connected computing nodes that work*
  - *Clusters improve the performance, provide cost-effective and improved node accessibility compared to single computing node.*
  - *Each node of computational cluster is set to perform the same task sub-tacks*
- **Data Flow (DF)** refers to data flow of from one node another.
- **Data Consistency** means copies of blocks have the values.
- **Data Availability** means at least one copy is available in case a partition becomes inactive or fails.

- **Resources** means computing system resources, ie, physical, virtual components devices, files, network connections and memory blocks made available for specified scheduled periods within system.
- **Resource management** refers managing resources such their creation, deletion and controlled usages. The manager functions also include managing

  (i) Availability for specified or scheduled periods,

  (ii)Prevention of resource unavailability after task finishes and

  (iii)Resources allocation when multiple tasks attempt to use same set resources,

- **Horizontal scalability** means increasing the number systems working in coherence. For example, using MPP's number servers as per the size of dataset.
- **Vertical scalability** means scaling up using giving system resources and increasing number of tasks in system.

- **Ecosystem** refers to system made up of multiple computing components, which work together.
- **Distributed File System means** system storing files. **Files can be for the set of data records**, key value pairs, hash key-value pairs, relational database or NoSQL database at distributed computing nodes, accessible after referring to their resource-pointer using **master directory service, look-up tables name node servers**.
- **Hadoop Distributed File System** means system of storing files (set data records, key-value pairs, hash key-value pairs or applications data)at distributed computing nodes **according to Hadoop architecture and accessibility of data blocks after finding reference to their racks and cluster.** NameNode servers enable referencing to data blocks.

- **Scalability of storage and processing** means the **execution using varying number of servers according to the requirements,** i.e., bigger data store on greater number of servers when required and on smaller data when smaller data used on limited number of servers.
- **Utility Cloud-based Services** means **infrastructure, software and computing platform services** similar to utility services, such as electricity, gas, water etc. Infrastructure refers to units for data-store, processing and network. **The IaaS, SaaS and PaaS are the services at the cloud**

2.2 HADOOP AND ITS ECOSYSTEM

- **Apache** initiated the project for developing storage and processing framework for Big Data storage and processing.
- **Doug Cutting and Machael J. Cafarelle** the creators named that framework as Hadoop.
- Cutting's son was fascinated by a **stuffed toy elephant**, named Hadoop, and this is how the name Hadoop was derived.
- **High Availability Distributed Object Oriented Platform**
- The project consisted of **two components**,
  - ***Data store*** *in blocks in the clusters*
  - ***Computations*** *at each individual cluster in parallel with another.*
- Hadoop components are written in **Java** with part of **native code in C**.
- The **command line** utilities are written in **shell scripts**.

- Hadoop is a **computing environment** in which input data stores, processes and stores the results.
- The environment consists of **clusters** which distribute at the cloud or set of servers.
- Each cluster consists of a string of **data files constituting data blocks.**
- The toy named Hadoop consisted of a **stuffed elephant**. The **Hadoop system cluster stuffs files in data blocks**.
- The complete system consists of a scalable distributed set of clusters

- The system characteristics are **scalable, self-manageable, self-healing** and distributed file system.
- **Scalable** means can be scaled up (enhanced) by **adding storage** and processing units as per the requirements.
- **Self-manageable** means **creation** of storage and **processing** resources which are used, **scheduled** and reduced or increased with the help of the **system itself**.
- **Self-healing** means that in case of faults, they are taken care of by the system itself. Self-healing **enables functioning and resources availability**. **Software detect and handle failures at the task level**. Software enable the service or task execution even in case of communication or node failure.

- The hardware scales up from a single server to thousands of machines that store the clusters.
- Each cluster stores a large number of **data blocks in racks**.
- Default data block size **is 64 MB**.
- **IBM BigInsights**, built on Hadoop deploys default **128 MB block size**.
- Hadoop framework provides the computing features of a system of distributed, flexible, scalable, fault tolerant computing with high computing power.
- Hadoop system is an efficient platform for the distributed storage and processing of a large amount of data.

- The Hadoop system manages **both, large sized structured and unstructured data** in different formats, such as XML, JSON and text with efficiency and effectiveness.
- The Hadoop system performs better with clusters of many servers when the focus is on **horizontal scalability**.
- The system **provides faster results** from Big Data and from unstructured data as well.

- Yahoo has more than **1,00,000 CPUs** in over **40,000 servers** running Hadoop, with its biggest Hadoop cluster running **4500 nodes** as of March 2017, according to the Apache Hadoop website.
- Facebook **has 2 major clusters**: a cluster has **1100-machines** with **8800 cores and about 12 PB raw storage.**
- A **300-machine** cluster with 2400 cores and about 3 PB (1 PB = 10^{15} B. nearly 2^{50} B) raw-storage.
- Each (commodity) node has 8 cores and 12 TB.

2.2.1 Hadoop Core Components

- The Hadoop core components of th

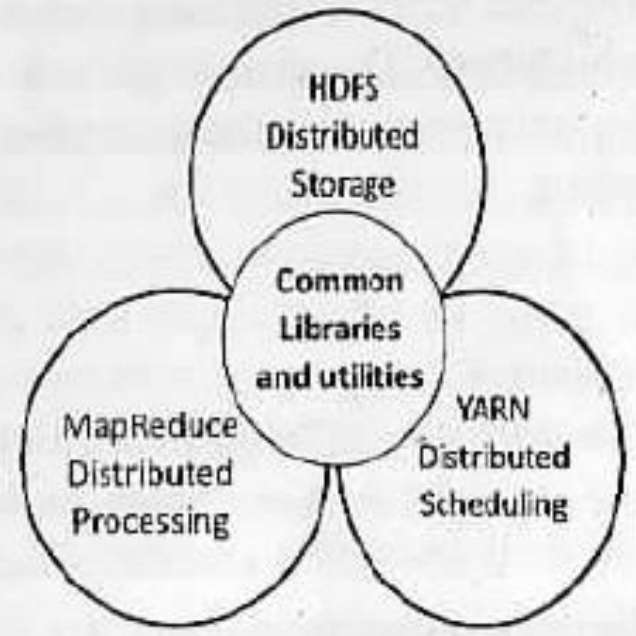


Figure 2.1 Core components of Hadoop

1. **Hadoop Common** –

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

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

3. MapReduce v1 - Software programming model in Hadoop 1 using Mapper and Reducer. The v1 processes large sets of data in parallel and in batches.

4. **YARN-Software** for managing resources for computing. The user application **tasks or sub-tasks run in parallel at the Hadoop**, uses scheduling and handles the requests for the resources in distributed way of running the tasks.

5 **MapReduce v2**-Hadoop 2 YARN-based system for parallel processing of large dataset and distributed processing of the application tasks

2.2.1.1 Spark

- Spark is an open-source cluster-computing framework of Apache Software Foundation.
- Hadoop deploys data at the disks. Spark provisions for **in-memory analytics**. Therefore, it also **enables OLAP and real-time processing**.
- Spark does **faster** processing of Big Data.
- Spark has been adopted by **large organizations**, such as Amazon, eBay and Yahoo. Several organization run Spark on clusters with thousands of nodes.
- Spark is now increasingly becoming popular.

Hadoop features are as follows

1 Fault-efficient scalable, flexible and modular design

- *Uses simple and modular programming model.*
- *The system provides **servers at high scalability**.*
- *The system is scalable by **adding new nodes** to handle larger data.*
- *Hadoop proves very helpful in storing, managing, processing and analysing Big Data.*
- *Modular functions make the system **flexible**. One can add or **replace components at ease.***
- *Modularity allows **replacing its components** for a different software tools.*

2. ***Robust design of HDFS:***

- *Execution of Big Data applications continue even when an **individual server or cluster fails.***
- *Hadoop provisions for **backup** (due to replications atleast three times for each data block) and a **data recovery mechanism**.*
- *HDPS thus has high **reliability**.*

3 Store and process Big Data: Processes Big Data of **3V characteristics**

4. Distributed clusters computing model with data locality Processes Big Data at high speed as the application **tasks and sub-tasks** submit to the DataNodes. One can achieve **more computing power by increasing the number of computing nodes**. The **processing splits** across multiple Datanodes(servers), and thus fast processing and **aggregated results**.

5 Hardware fault-tolerant A fault does not affect data and application processing. If a node goes down the other nodes take care of the residue. This is due to multiple copies of all **data blocks which replicate automatically**. Default is **three copies of data blocks.**

6 Open-source framework Open source access and **cloud services** enable large data store. Hadoop uses a cluster of multiple **inexpensive servers or the cloud**

7. Java and Linux based Hadoop uses **Java interfaces**. Hadoop base is Linux but has its own set of shell commands support.

- Hadoop provides various components and interfaces for distributed file system and general input/output.
- This includes serialization, Java RPC (Remote Procedure Call) and file-based data structures in Java.
- HDFS is basically designed more for batch processing. Streaming uses standard input and output to communicate with the Mapper and Reduce codes. Stream analytics and real-time processing poses difficulties when streams have high throughput of data. The data access is required faster than the latency at HDFS.
- YARN provides a platform for many **different modes of data processing**, from **traditional batch processing** to processing of the applications such as **interactive queries, text analytics and streaming analysis**.
- These different types of data can be moved in HDFS for analysis using interactive query processing tools of Hadoop ecosystem components, such as **Hive** or can be provided to **online business processes with the help of Apache HBase.**

2.2.3 Hadoop Ecosystem Components

- Hadoop ecosystem refers to a combination of **technologies**. Hadoop ecosystem consists of own **family of applications** which tie up together with the Hadoop.
- The system components support the storage, processing, access, analysis, governance, security and operations for Big Data.
- The data store system consists of **clusters, racks, DataNodes and blocks**.
- Hadoop deploys application **programming model, such as MapReduce and HBase.**
- **YARN manages resources** and schedules sub-tasks of the application.
- HBase uses columnar databases and does OLAP

- The four layers are

(i) Distributed storage layer

(ii) Resource-manager layer for job or application sub-tasks scheduling and execution

(iii) Processing-framework layer, consisting of Mapper and Reducer for the MapReduce process-flow

(iv) APIs at application support layer (applications such as Hive and Pig). The codes communicate and run using MapReduce or YARN at processing framework layer. Reducer output communicate to APIs.

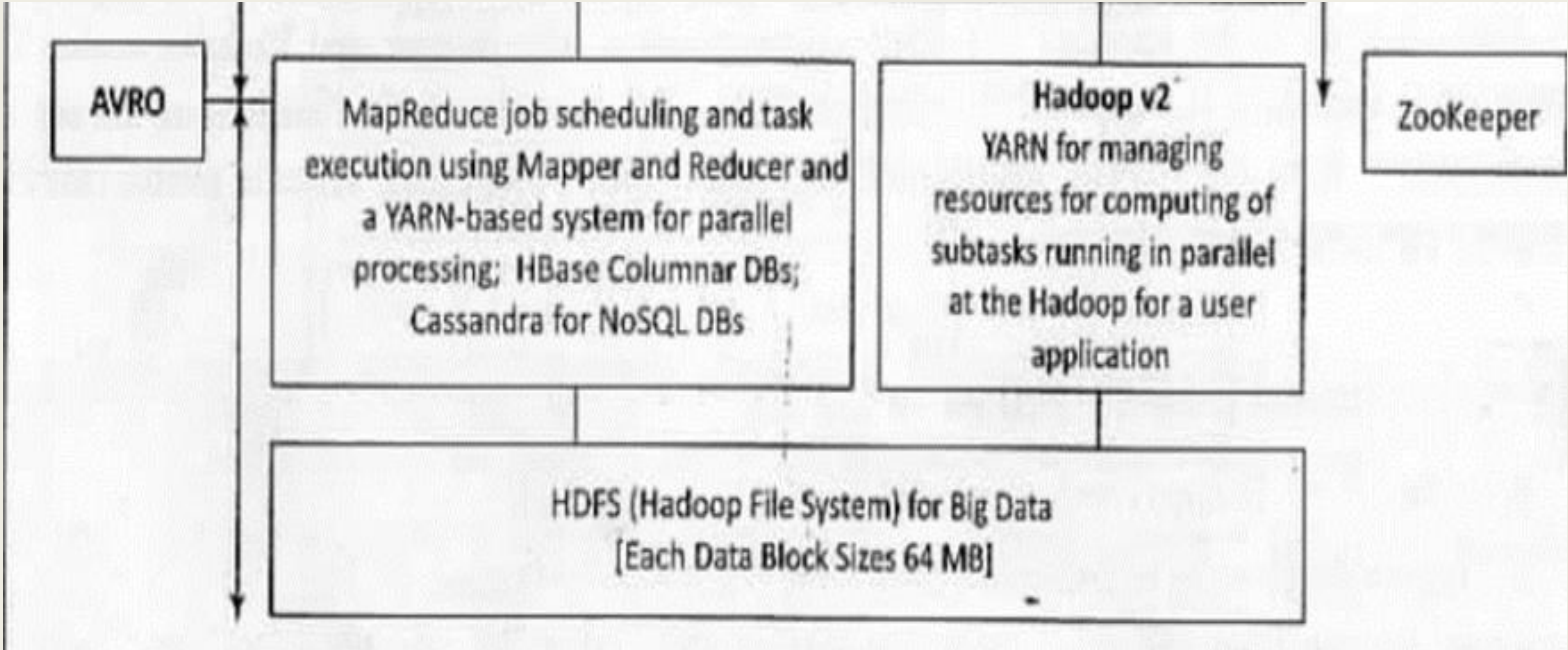
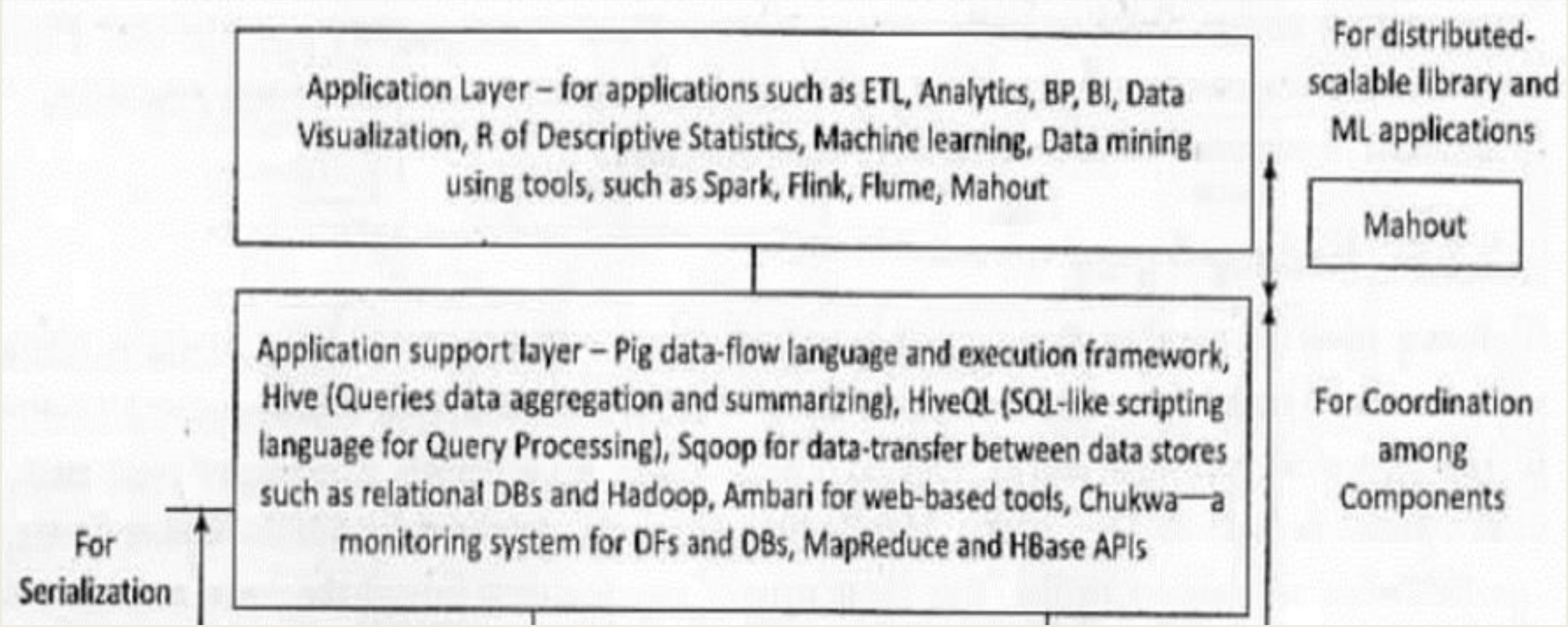


Figure 2.2 Hadoop main components and ecosystem components

AVRO enables data serialization between the layers. Zookeeper enables coordination among layer


Application Layer – for applications such as ETL, Analytics, BP, BI, Data Visualization, R of Descriptive Statistics, Machine learning, Data mining using tools, such as Spark, Flink, Flume, Mahout
For distributed-scalable library and ML applications
Mahout
Application support layer – Pig data-flow language and execution framework, Hive (Queries data aggregation and summarizing), HiveQL (SQL-like scripting language for Query Processing), Sqoop for data-transfer between data stores such as relational DBs and Hadoop, Ambari for web-based tools, Chukwa—a monitoring system for DFs and DBs, MapReduce and HBase APIs
For Coordination among Components
For Serialization

- The holistic view of Hadoop architecture provides an idea of implementation of Hadoop components of the ecosystem.
- Client hosts run applications using Hadoop ecosystem projects, such as **Pig, Hive and Mahout.**
- Hadoop programs run on any platform with the **Java virtual-machine** deployment model. HDFS is a Java-based distributed file system that can store various kinds of data on the computers.

2.2.4 Hadoop Streaming

- HDFS with **MapReduce and YARN-based** system enables parallel processing of large datasets.
- **Spark provides in-memory** processing of data, thus improving the processing speed.
- Spark and **Flink technologies** enable **in-stream processing**.
- Spark includes **security features**.
- Flink is emerging as a powerful tool. Flink improves the overall performance as it provides single run-time for streaming as well a batch processing. Simple and flexible architecture of Flink is suitable for streaming data.

2.2.5 Hadoop Pipes

- Hadoop Pipes are the **C++ Pipes** which interface with MapReduce.
- Apache Hadoop provides an **adapter layer**, which processes in pipes.
- **A pipe means data streaming into the system at Mapper input and aggregated results flowing out at outputs.**
- The adapter layer **enables running** of application tasks in C++ coded MapReduce programs.
- Applications which require faster numerical computations can **achieve higher throughput** using C++ when used through the pipes, as compared to Java.
- Pipes do not use the **standard I/O** when communicating with Mapper and Reducer codes.
- **Cloudera distribution** including Hadoop (CDH) version CDH 5.0.2 runs the pipes. Distribution means software downloadable from the website distributing codes.
- IBM Power Linux systems enable working with Hadoop pipes and system libraries.

2.3 HADOOP DISTRIBUTED FILE SYSTEM

- Big Data analytics applications are software applications that **leverage large scale data.**
- The applications **analyze** Big Data using **massive parallel processing** frameworks.
- HDFS is a **core component** of Hadoop. HDFS is designed to run on Big Data which may range from GBs (1 GB = 2^{30} B) to PBs (1 PB = 10^{15} B. nearly the 2^{50} B).
- HDFS stores the data in a **distributed manner** in order to compute fast.
- The distributed data store in HDFS stores data in any **format regardless of schema.**
- HDFS provides high **throughput access to data-centric** applications that require large-scale data processing workloads.

2.3.1 HDFS Data Storage

- Hadoop data store concept implies storing the data at a number of **clusters**. Each cluster has a number of data stores, called **racks**. Each rack stores a number of **DataNodes**. Each **DataNode** has a large number of **data blocks**. The racks distribute across a cluster.
- The **nodes have processing and storage capabilities**. The nodes have the data in data blocks to run the application tasks. The data blocks **replicate by default at least on three DataNodes in same or remote nodes.**
- Data at the stores enable running **distributed applications** including analytics, data mining, OLAP using the clusters.
- **A file, containing the data divides into data blocks**. A data block default size **is 64 MBs** (HDFS division of files concept is similar to **Linux or virtual memory page** in Intel x86 and Pentium processors where the block size is fixed and is of **4 KB**).

Hadoop HDFS features are as follows:

(1) Create, append, delete, rename and attribute modification functions

(2) Content of individual file **cannot be modified or replaced but appended with new data at the end of the file.**

(3) Write once but read many times during usages and processing.

(4) Average file size can be more than 500 MB.

The following is an example how the files store at a Hadoop cluster.

EXAMPLE 2.2

Consider a data storage for University students. Each student data, stuData which is in a file of size less than 64 MB (1 MB = 2^{20}B). A data block stores the full file data for a student of stuData_id N, where N=1 to 500.

(i) How the files of each student will be distributed at a Hadoop cluster? How many student data can be stored at one cluster? Assume that each rack has two DataNodes for processing each of 64 GB (1 GB= 2^{30}B) memory. Assume that cluster consists of 120 racks, and thus 240 DataNodes.

(ii) What is the total memory capacity of the cluster in TB ((1 TB=2^{40}B) and DataNodes in each rack?

(iii) Show the distributed blocks for students with ID=96 and 1025. Assume default replication in the DataNodes = 3.

(iv) What shall be the changes when a stuData file size <=128 MB?

SOLUTION (i)

- Data block default size is 64 MB. Each student's file size is less than 64MB.
- Therefore, for each student file one data block suffices. A data block is in a DataNode.
- Assume, for simplicity, each rack has two nodes each of memory capacity = 64 GB. Each node can thus store 64 GB/64MB = 1024 data blocks = 1024 student files. Each rack can thus store 2 x 64 GB/64MB= 2048 data blocks = 2048 student files.
- Each data block default replicates three times in the DataNodes.
- Therefore, the number of students whose data can be stored in the cluster = number of racks multiplied by number of files divided by 3= 120 x 2048/3=81920.
- Therefore, the maximum number of 81920 stuData_IDN files can be distributed per cluster, with N=1 to 81920.

(ii) What is the total memory capacity of the cluster in TB ((1 TB=2^{40}B) and DataNodes in each rack?

Solution

Total memory capacity of the cluster = 120 x 128 MB = 15360 GB = 15 TB. Total memory capacity of each DataNode in each rack = 1024 x 64 MB = 64 GB.

(iii) Show the distributed blocks for students with ID=96 and 1025. Assume default replication in the DataNodes = 3.

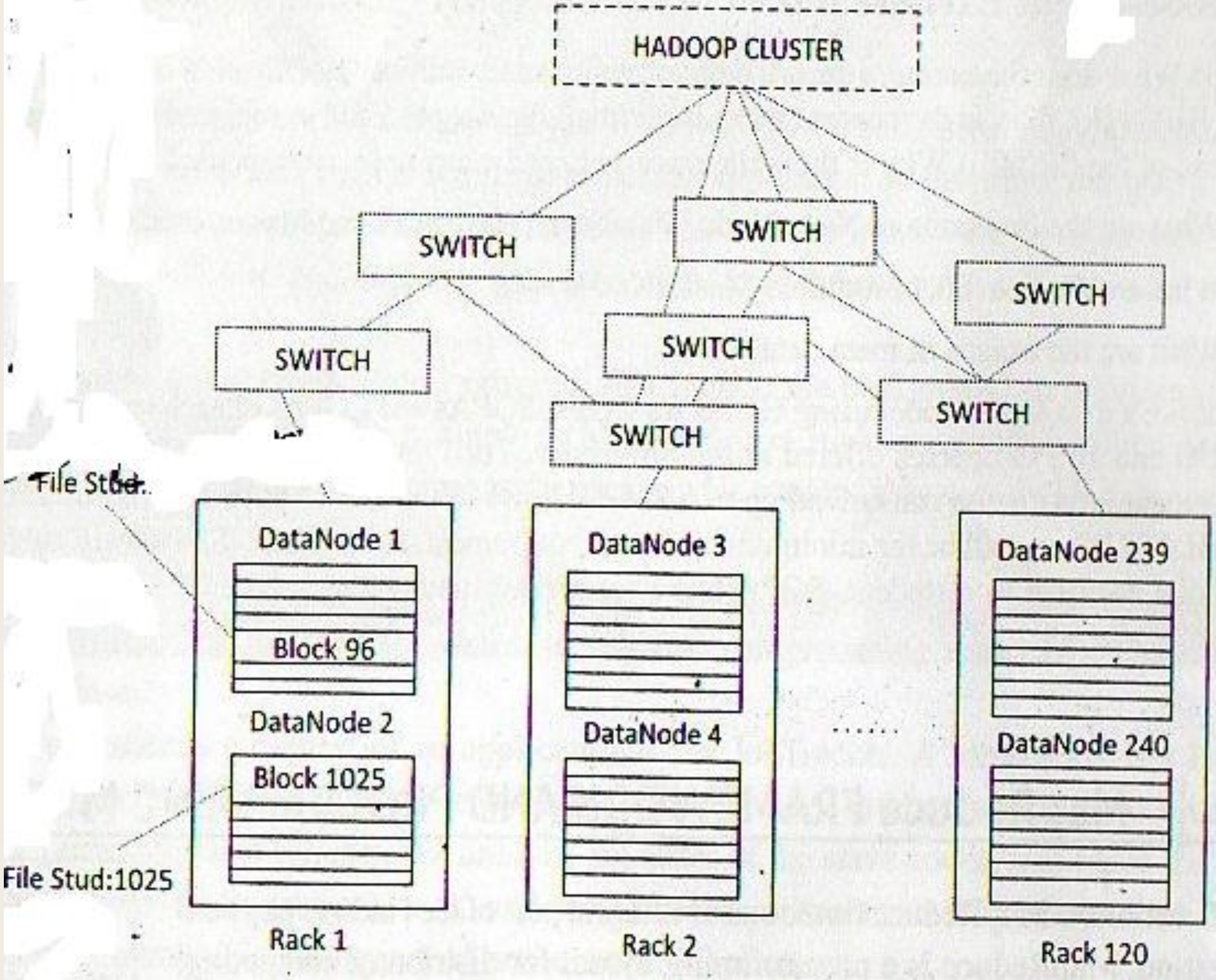


Figure 2.3 A Hadoop cluster example, and the replication of data blocks in racks for two students of IDs 96 and 1025

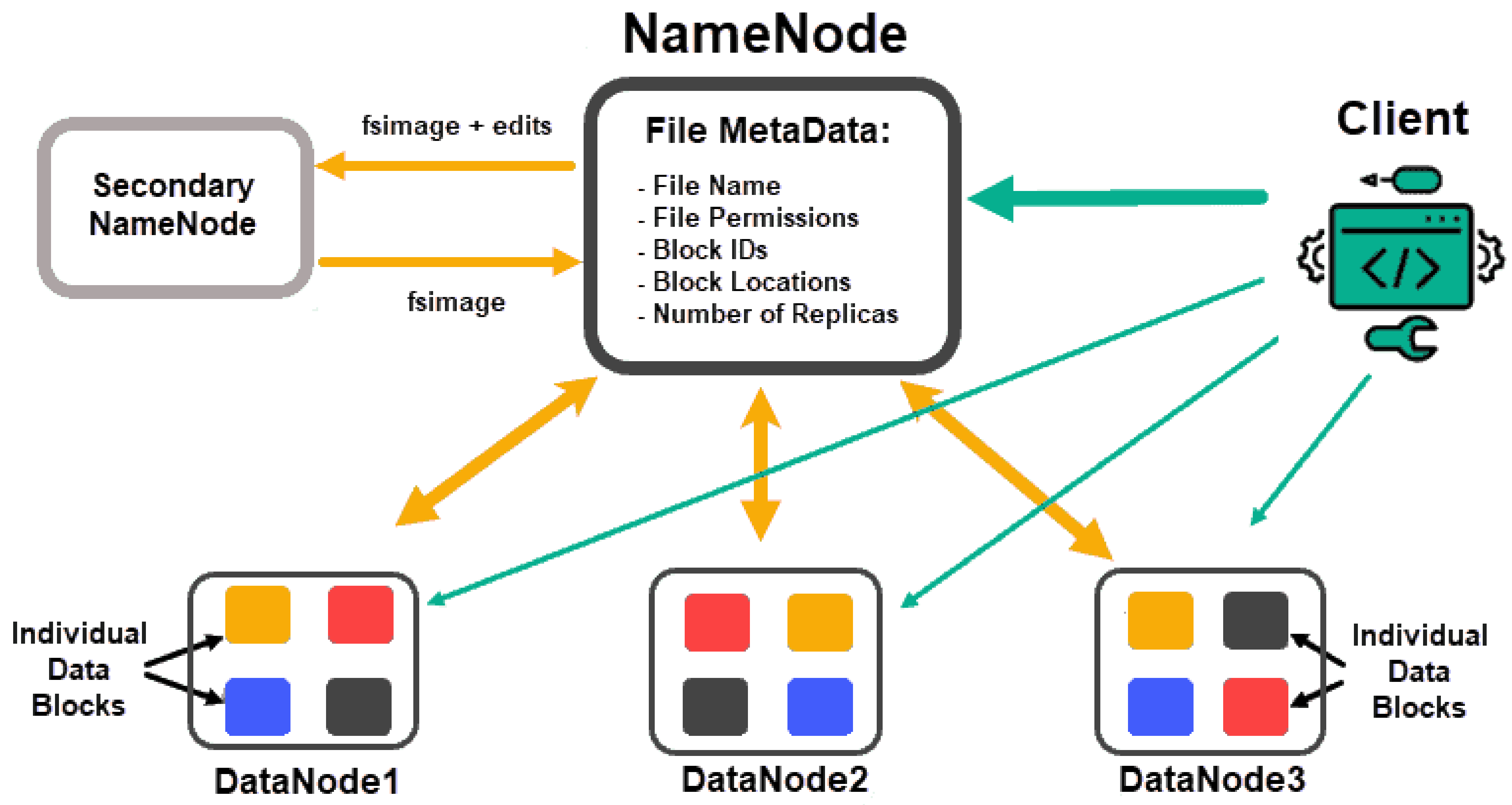
iv) What shall be the changes when a stuData file size <=128 MB?

Solution

Changes will be that each block will have half the number of data blocks.

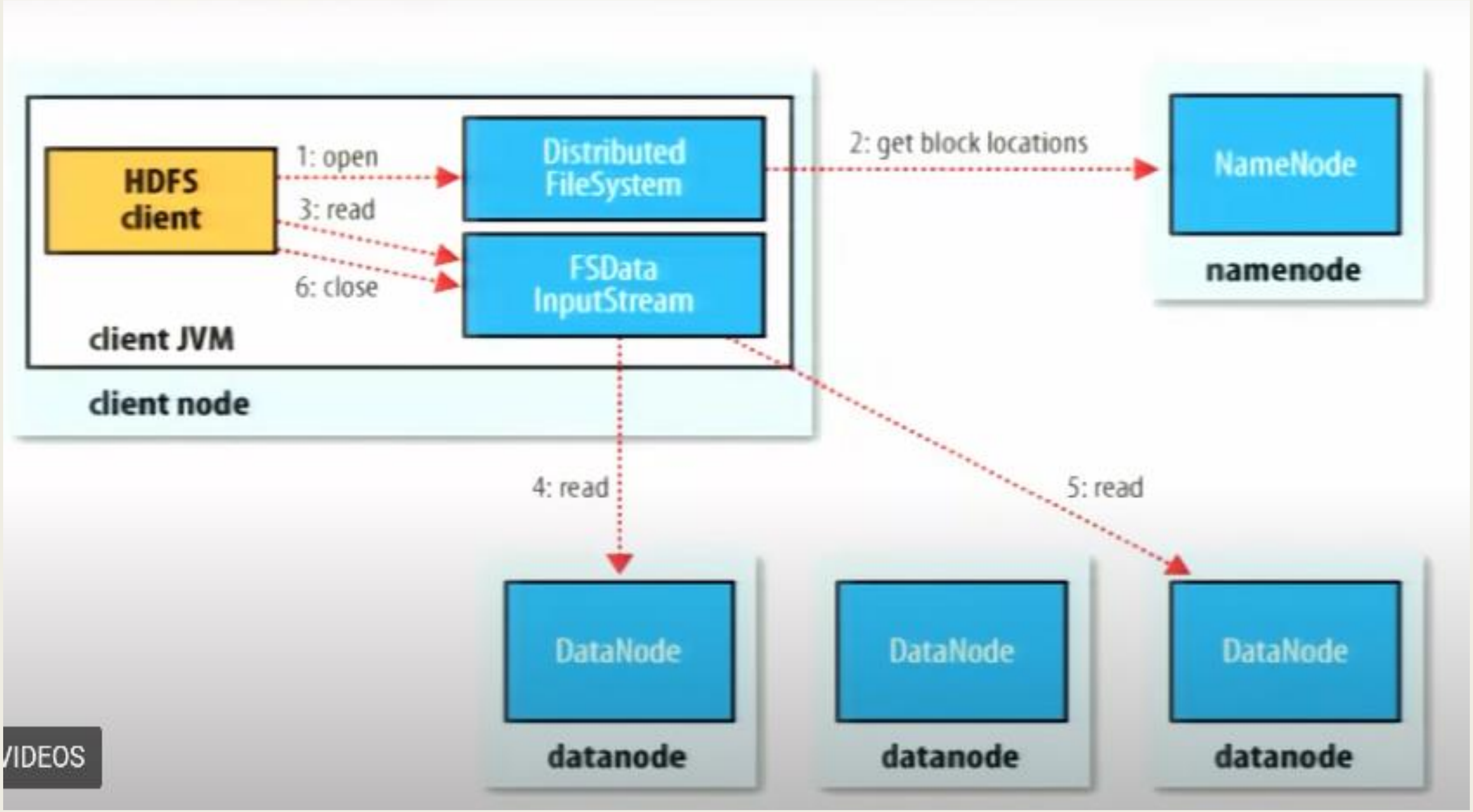
2.3.1.1 Hadoop Physical Organization

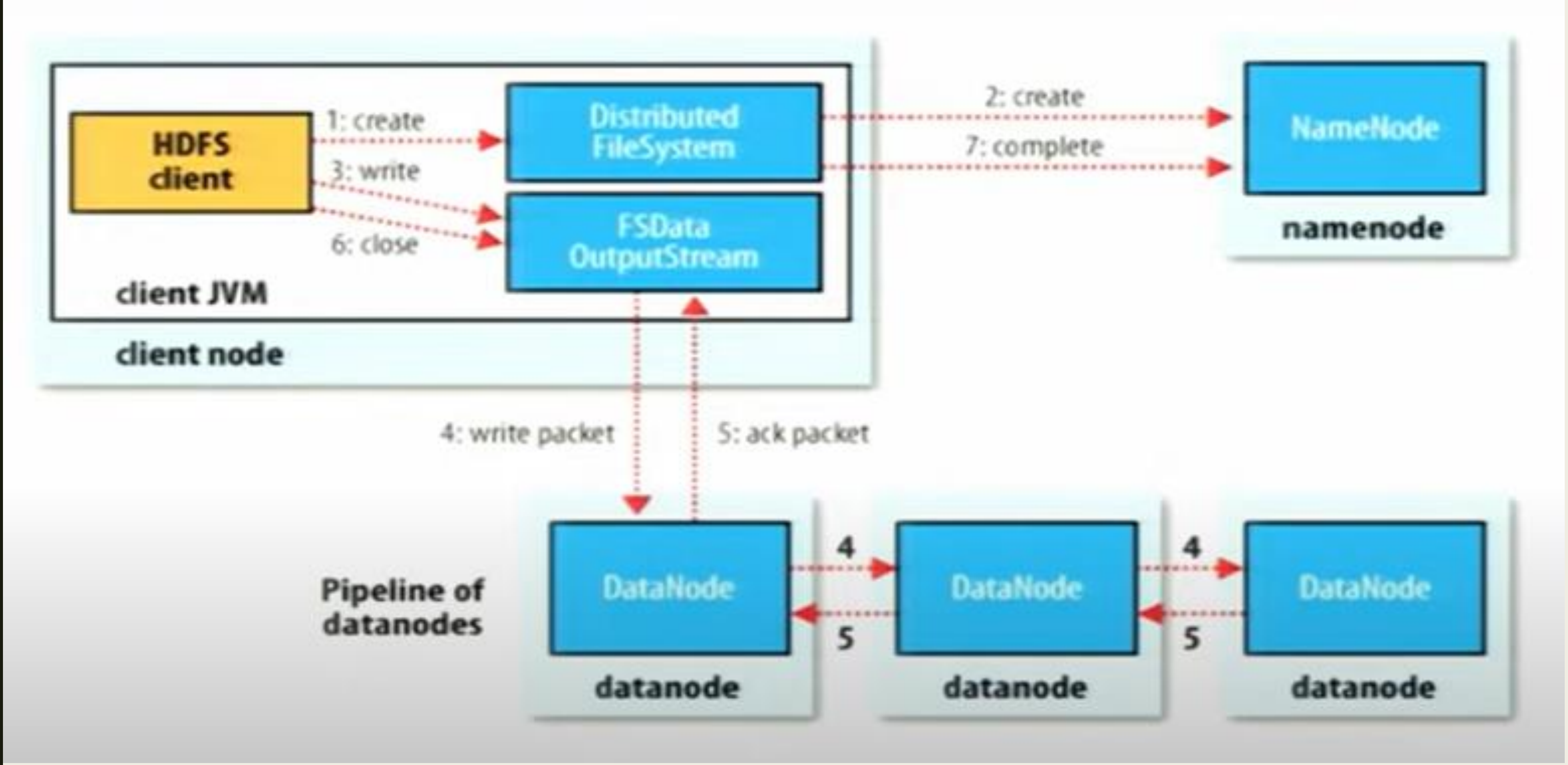
- The conventional file system uses **directories**. A directory consists of **folders**. A folder consists of **files**.
- When data processes, the data sources identify by **pointers** for the resources. **A data-dictionary stores the resource pointers.**
- **Master tables at the dictionary store at a central location**..
- Similarly, the files, **DataNodes and blocks** need the identification during processing at HDFS.
- HDFS use the **NameNodes and DataNodes**.
- A **NameNode stores the file's meta data**. Meta data gives **information** about the file of user application, **but does not participate in the computations**.
- The **DataNode stores the actual data files in the data blocks**.


NameNode
File MetaData:
- File Name
- File Permissions
- Block IDs
- Block Locations
- Number of Replicas
Secondary NameNode
fsimage + edits
fsimage
Client
</>
Individual Data Blocks
DataNode1
DataNode2
DataNode3
Individual Data Blocks

- Few nodes in a Hadoop cluster act as **NameNodes**. These nodes are termed as **MasterNodes** or simply **masters**.
- The masters have a different configuration supporting high **DRAM and processing power**. The masters have **much less local storage**.
- **Majority** of the nodes in Hadoop cluster act as **DataNodes and TaskTrackers.**
- These nodes are referred to as **slave nodes or slaves**. The slaves have **lots of disk storage and moderate amounts of processing capabilities and DRAM.**
- Slaves are **responsible** to store the data and process the computation tasks submitted by the clients.
- Clients as the users run the application with the help of Hadoop ecosystem projects.
- For example, Hive, Mahout and Pig are the ecosystem's projects.

- They are not required to be present at the Hadoop cluster.
- A **single MasterNode** provides HDFS, MapReduce and Hbase using threads in small to medium sized clusters.
- When the **cluster size is large**, multiple servers are used, such as to balance the load.
- The **secondary NameNode** provides NameNode management services and Zookeeper is used by HBase for metadata storage.


HDFS client
1: open
3: read
6: close
Distributed FileSystem
FSData InputStream
client JVM
client node
2: get block locations
NameNode
namenode
4: read
5: read
DataNode
datanode
DataNode
datanode
DataNode
datanode
VIDEOS



HDFS client
1: create
3: write
6: close
Distributed FileSystem
FSData OutputStream
client JVM
client node
2: create
7: complete
NameNode
namenode
4: write packet
5: ack packet
Pipeline of datanodes
DataNode
4
5
DataNode
4
5
DataNode
datanode
datanode
datanode

- The MasterNode fundamentally plays the role of a **coordinator**. The MasterNode receives **client connections**, maintains the description of the **global file system namespace**, and the **allocation of file blocks**.
- It also **monitors** the state of the system in order to detect any failure.
- The Masters consists of three components **NameNode, Secondary NameNode and JobTracker.**

The **NameNode stores all the file system related information such** as:

- ***The file section is stored in which part of the cluster***
- ***Last access time for the files***
- ***User permissions like which user has access to the file.***

- **Secondary NameNode** is an **alternate** for NameNode. Secondary node keeps a copy of NameNode meta data. Thus, stored meta data can be rebuilt easily, in case of NameNode failure.
- The **JobTracker** coordinates the parallel processing of data.
- Masters and slaves, and Hadoop client (node) load the data into cluster, submit the processing job and then retrieve the data to see the response after the job completion.

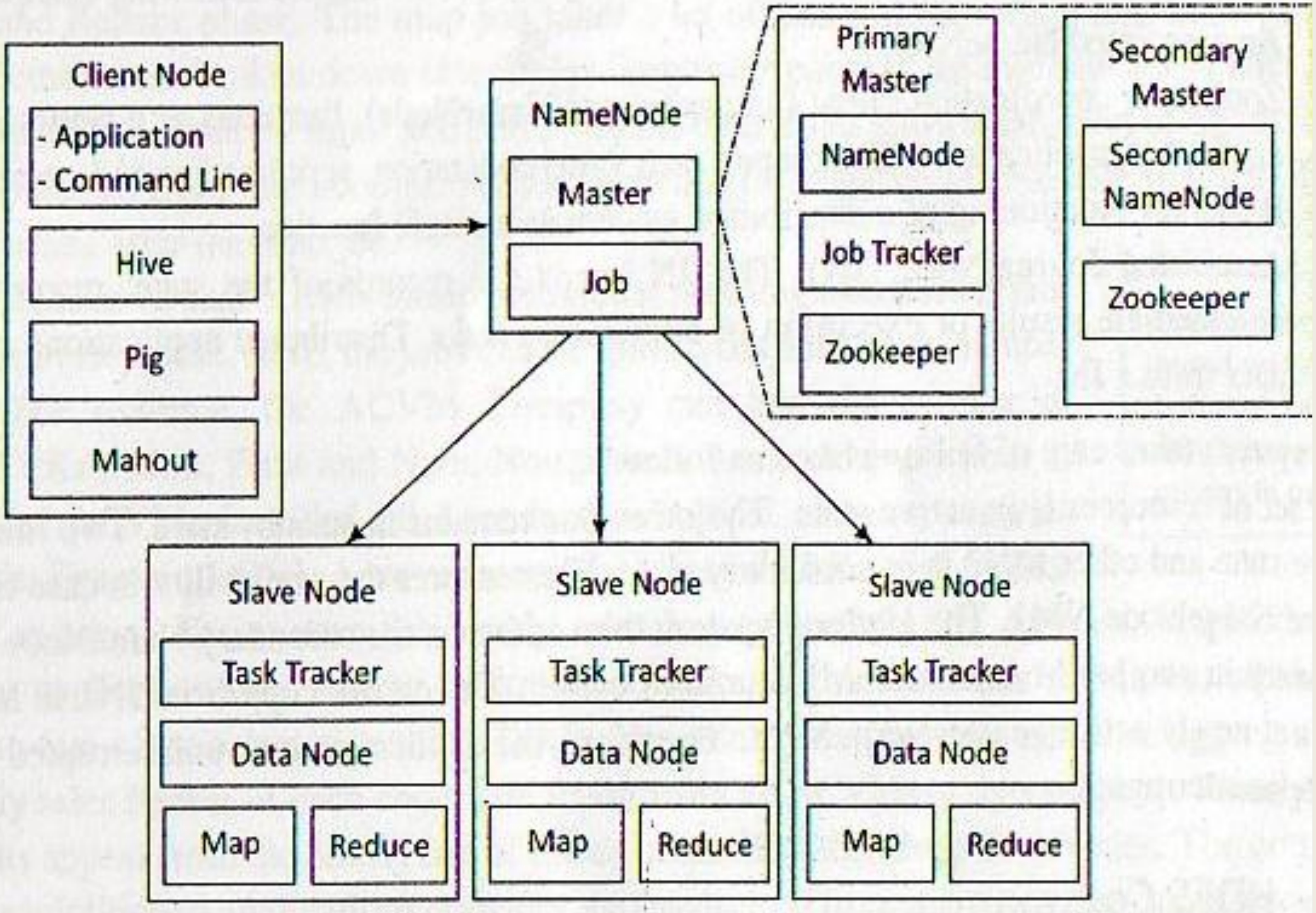


Figure 2.4 The client, master NameNode, MasterNodes and slave nodes