

Big Data (CS-3032)

**Kalinga Institute of Industrial Technology
Deemed to be University
Bhubaneswar-751024**

School of Computer Engineering



Strictly for internal circulation (within KIIT) and reference only. Not for outside circulation without permission

3 Credit

Lecture Note

Course Contents



Sr #	Major and Detailed Coverage Area	Hr s
2	Big Data Technology Foundations Exploring the Big Data Stack, Data Sources Layer, Ingestion Layer, Storage Layer, Physical Infrastructure Layer, Platform Management Layer, Security Layer, Monitoring Layer, Analytics Engine, Visualization Layer, Big Data Applications, Virtualization.	8

Exploring the Big Data Stack

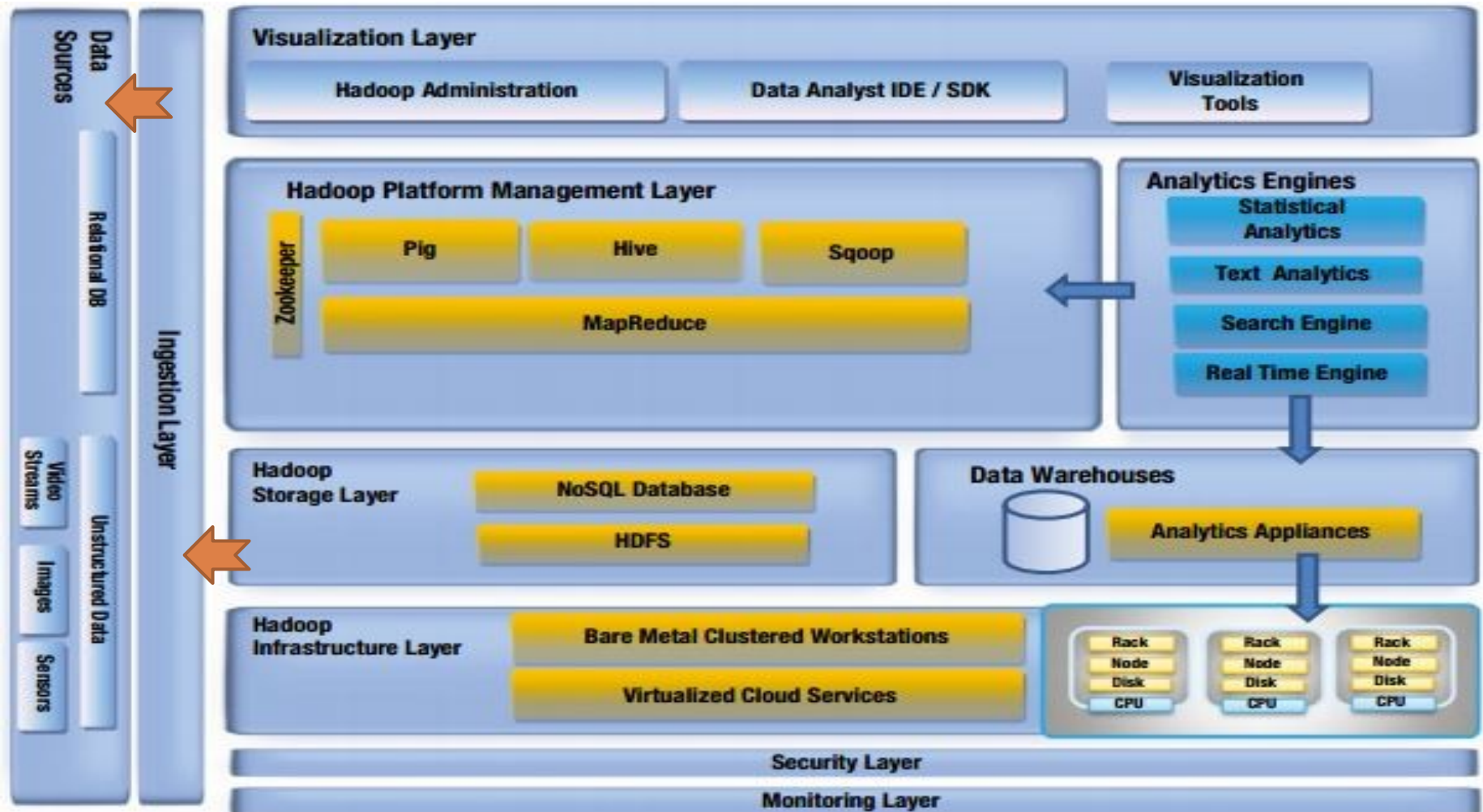


The first step in the process of designing any data architecture is to create a model that should give a complete view of all required elements. Although, initially, creating a model may seem to be a time-consuming task, however, it can save a significant amount of time, effort, and rework during the subsequent implementations.

Big Data analysis also needs the creation of a model or architecture, commonly known as the Big Data architecture. Such architecture of the big data environment must fulfill all the foundational requirements and must be able to perform the following key functions:

- ☐ Capturing data from different data sources
- ☐ Cleaning and integrating data of different types of format
- ☐ Storing and organizing data
- ☐ Analyzing data
- ☐ Identifying relationship and patterns
- ☐ Deriving conclusions based on the data analysis results

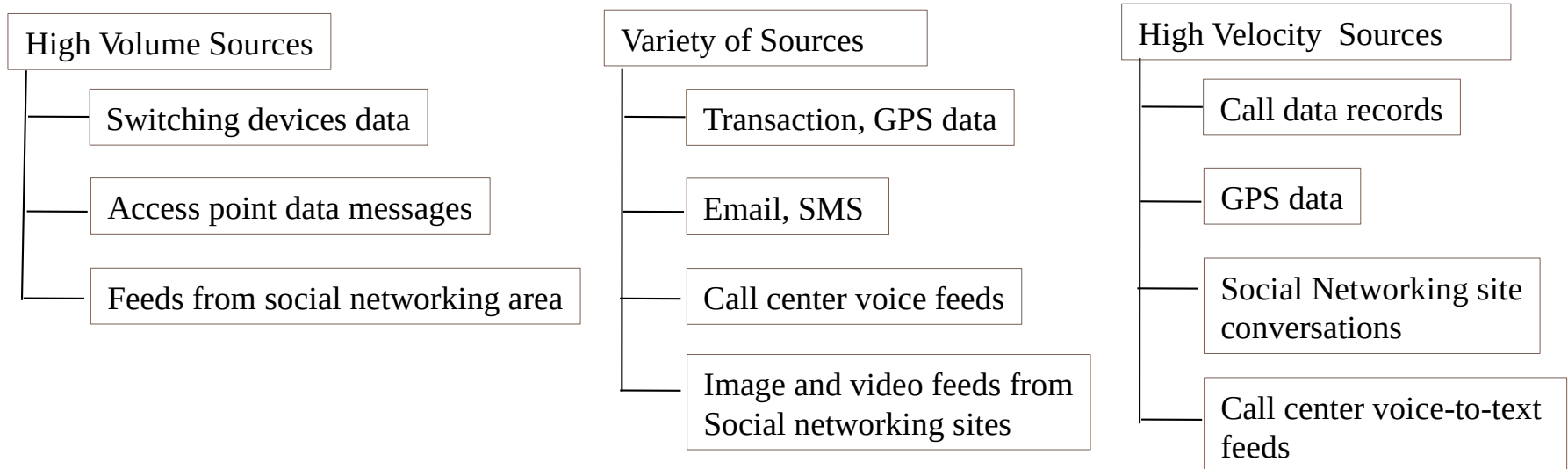
Big Data Architecture Layers



Data Sources Layer



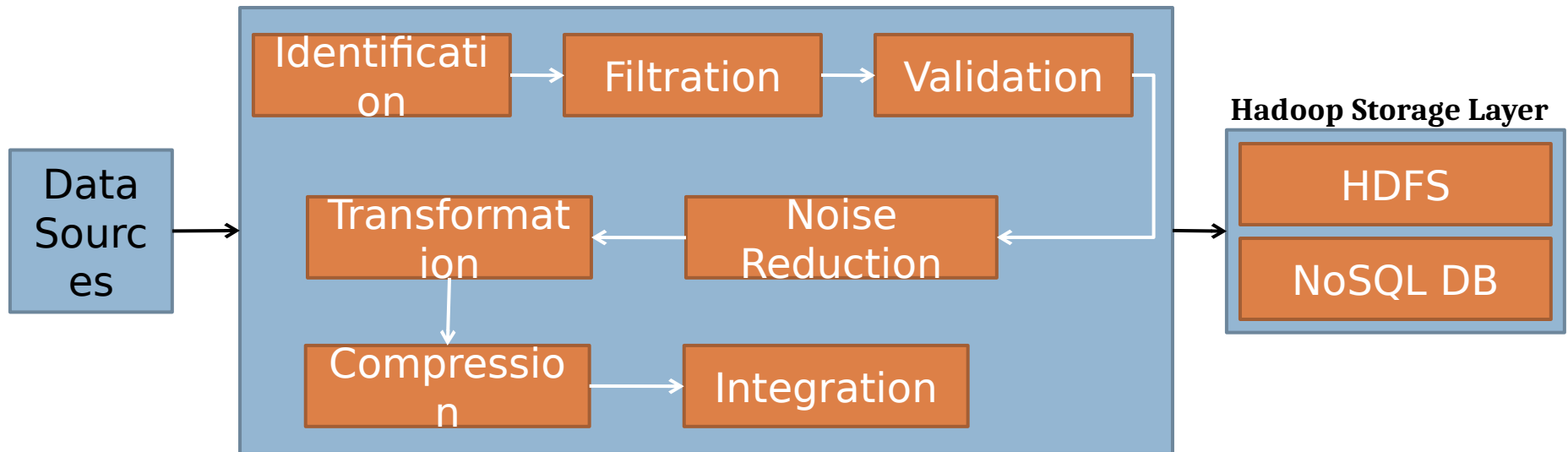
The basic function of the data sources layer is to absorb and integrate the data from various sources, at varying velocity and in different formats. It includes everything from sales records, customer database, feedback, social media channels, marketing list, email archives and any data gleaned from monitoring or measuring aspects of operations. Before this data is considered for big data stack, differentiation is required between the noise and relevant information. Example of different data sources the telecom industry obtains its data:



Ingestion Layer



It absorb the huge inflow of data and separates noise from relevant information. It handle huge volume, high velocity, and a variety of data. It validates, cleanses, transforms, reduces, and integrates the unstructured data into the Big Data stack for further processing. In this layer, the data passes through the following stages.



Ingestion Layer cont'd



In the ingestion layer, the data passes through the following stages:

1. **Identification** – At this stage, **data is categorized into various known data formats or even unstructured data** is assigned with default formats.
2. **Filtration** – At this stage, the information relevant for the enterprise is filtered on the basis of the **Enterprise Master Data Management (MDM)** repository. **MDM is a comprehensive method of enabling an enterprise to link all of its critical non-transactional data to a common point of reference.** Example – One probably have an address book on both phone and email system and would like to keep the contact details the same on both systems. You also probably use Facebook and would like to add Facebook friends as contacts in your address book. Usually this is viewed in terms of synchronization between the different systems and various software methods are available to enable this. MDM solution for contact management is a single version of the address book is maintained as the master. Different views of the master address book are able to be synchronized to any of the linked address book applications. Every time a contact is added to any of the linked address books it is matched against the master to check whether the contact already exists.

Ingestion Layer cont'd



3. **Validation** – At this stage, **the filtered data is analyzed against MDM metadata.**
4. **Noise Reduction**– At this stage, **data is cleansed by removing the noise and minimizing the related disturbances.**
5. **Transformation** - At this stage, **data is split or combined on the basis of its type, contents, and the requirements of the organizations.**
6. **Compression** - At this stage, **the size of the data is reduced without affecting its relevance for the required process** and it guarantees of no adverse affect to the analysis result.
7. **Integration** - At this stage, the refined dataset **is integrated with the Hadoop storage layer, which primarily consists of HDFS (Hadoop Distributed File System) and NoSQL databases.**

Storage Layer



It is the place where **Big Data** lives. **Hadoop** is an open source framework used to store large volumes of data in a distributed manner across multiple machines. The Hadoop storage layer supports fault-tolerance and parallelization, which enable **high-speed distributed processing algorithms to execute over large-scale data**. Two major component of **Hadoop**: a scalable **Hadoop Distributed File System** that can support petabytes of data and a **MapReduce** Engine that can compute results in batches. **HDFS stores data in the form of block of files and follows the write-once-read-many model to access data from these blocks of files**. However, apart from files, different types of database are also used. However, **storage requirement can be addressed by a single concept known as Not Only SQL (NoSQL) databases**. Hadoop has its own database, known as Hbase, but others including Amazon's DynamoDB, MongoDB, AllegroGraph, Cassandra (used by Facebook) and IndefiniteGraph are popular too.

SQL vs. NoSQL



SQL	NoSQL
SQL databases are primarily called as Relational Databases (RDBMS)	NoSQL database are primarily called as non-relational or distributed database.
SQL databases are table based databases i.e. represent data in form of tables which consists of n number of rows of data	NoSQL databases are document based, key-value pairs, graph databases or wide-column stores. It do not have standard schema definitions which it needs to adhered to.
Predefined schema for structured data	Dynamic schema for unstructured data
Vertically scalable wherein databases are scaled by increasing the horse-power of the hardware.	Horizontally scalable wherein databases are scaled by increasing the databases servers in the pool of resources to reduce the load.
Emphasizes on ACID properties (Atomicity Consistency Isolation	Follows Brewers CAP theorem (Consistency Availability and















SQL vs. NoSQL cont'd



SQL	NoSQL
Not best fit for hierarchical data storage	Fits better for the hierarchical data storage as it follows the key-value pair way of storing data similar to JSON data. highly preferred for large data set
Good fit for the complex query intensive environment	Not good fit for complex queries. On a high-level, NoSQL don't have standard interfaces to perform complex queries, and the queries themselves in NoSQL are not as powerful as SQL query language.
Examples - MySql, Oracle, Sqlite, Postgres and MS-SQL	Examples - MongoDB, BigTable, Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb

Different NoSQL Databases



Document Database	Graph Databases
   Couchbase MarkLogic mongoDB	  Neo4j InfiniteGraph The Distributed Graph Database
Wide Column Stores	Key-Value Databases
    redis amazon DynamoDB AEROSPIKE riak	     accumulo™ HYPERTABLE INC Cassandra APACHE HBASE Amazon SimpleDB

Source: <http://www.aryannava.com>

Physical Infrastructure Layer



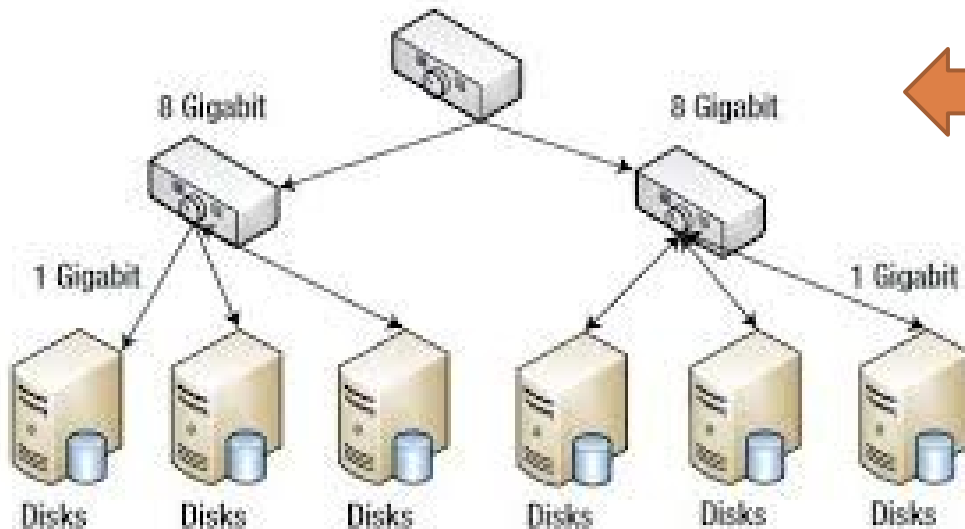
Principles on which **Big Data implementation** is based:

1. **Performance** – High-end infrastructure is required to deliver high performance with low latency (the total time taken by a packet to travel from one node to another node). It is measured end to end, on the basis of a single transaction or query request. It would be rated high if the total time taken in traversing a query request is low.
2. **Availability** – The infrastructure setup must be available at all times to ensure nearly a 100 percent uptime guarantee of service.
3. **Scalability** - The infrastructure must be scalable enough to accommodate varying storage and computing requirements. It must be capable enough to deal with any unexpected challenges.
4. **Flexibility** – Flexible infrastructures facilitate adding more resources to the setup and promote failure recovery.
5. **Cost** – Affordable infrastructure must be adopted including hardware, networking and storage requirements. Such parameters must be considered from the overall budget and trade-offs can be made, where necessary.

Physical Infrastructure Layer cont'd



So it can be concluded that a robust and inexpensive physical infrastructures can be implemented using Big Data. **This requirement is handled by the Hadoop physical infrastructure layer.** **This layer is based on distributed computing model, which allows the physical storage of data in many different locations by linking item through networks and the distributed file systems.** It also has to support redundancy of data. **This layer takes care of the hardware and network requirements and can provide a virtualized cloud environment** or a distributes grid of commodity servers over a fast gigabit network. Hardware topology used for Big Data Implementation is as follows.



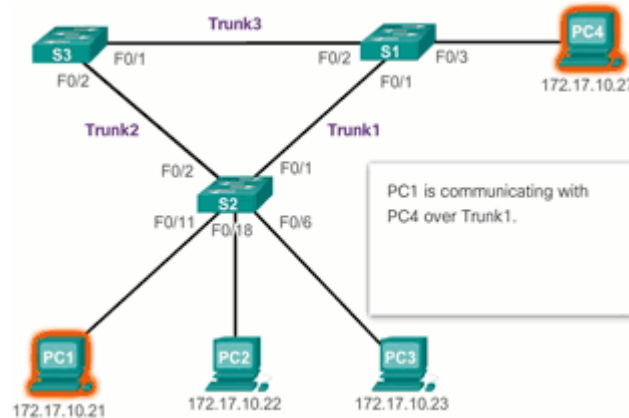
The main components of a Hadoop infrastructure:

- 1.n commodity servers (8-core, 24GBs RAM, 4 to 12 TBs)*
- 2.2-level network (20 to 40 nodes per rack)*

Physical Infrastructure Design Consideration



- 1. Physical Redundant Networks** – In the Big data environment, networks should be redundant and capable of accommodating the anticipated volume and velocity of the inbound and outbound data in case of heavy network traffic. The strategy must be prepared for improving their network performance to handle the increase in the volume, velocity, and variety of data.



Network redundancy is a process through which additional or alternate instances of network devices, equipment and communication mediums are installed within network infrastructure. It is a method for ensuring network availability in case of a network device or path failure and unavailability.

Physical Infrastructure Design Consideration cont'd

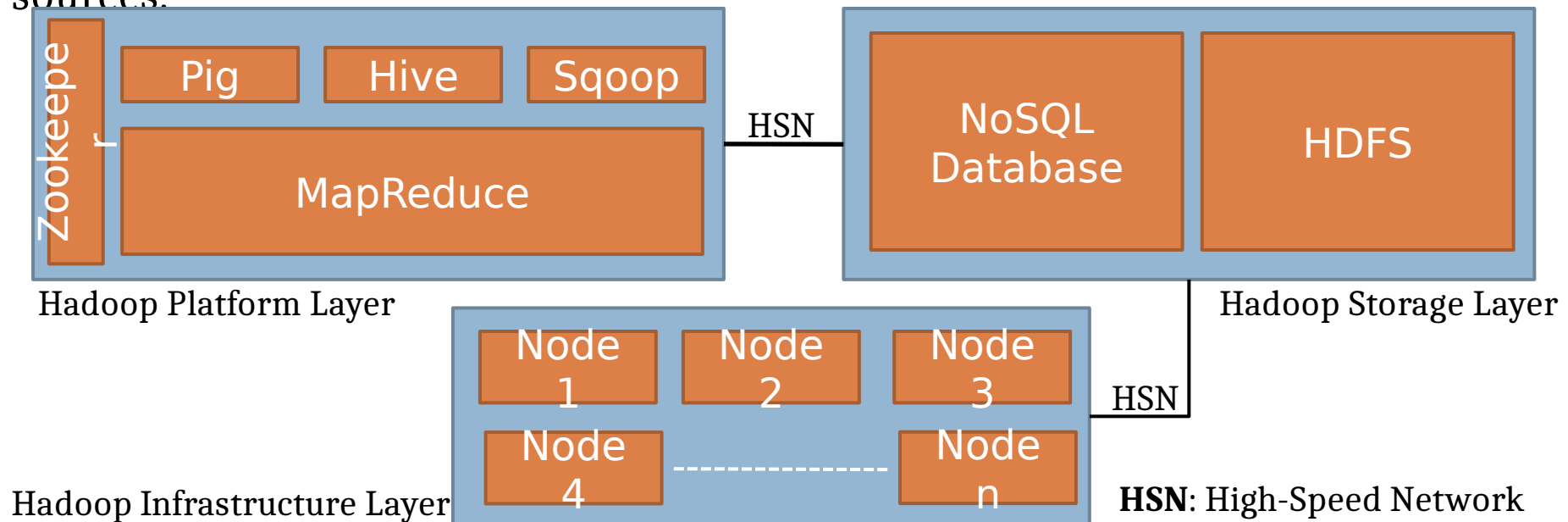


2. **Managing Hardware: Storage and Servers** – **Hardware resources for storage and servers must have sufficient speed and capacity to handle all expected types of Big Data.** If slow servers are connected to high-speed networks, the slow performance of the servers will be of little use and can at a times also become a bottleneck.
3. **Infrastructure Operations** - Proper management of data handling operations provides a well-managed environment, which in turn gives the greatest levels of performance and flexibility.

Platform Management Layer



The role of the **platform management layer is to provide tools and query languages for accessing NoSQL databases**. This layer uses the **HDFS storage file system that lies on the top of the Hadoop physical infrastructure layer**. The data is no longer stored in a monolithic server where the SQL functions are applied to crunch it. **Redundancy is built into this infrastructure** for the very simple reason that we are dealing with large volume of data from different sources.



Security Layer



The **security layer** handles the basic security principles that **Big Data architecture** should follow. Following security checks must be considered while designing a Big Data stack:

- ❑ It must **authenticate nodes by using protocols** such as Kerberos*
- ❑ It must enable **file-layer encryption**
- ❑ It must subscribe a **key management service for trusted keys and certificates**
- ❑ It must **maintain logs of the communication that occurs between nodes and trace any anomalies across layers** by using distributed logging mechanisms.
- ❑ It must ensure **a secure communication between nodes by using the Secure Sockets Layer (SSL)**
- ❑ It must validate data during the deployments of datasets or while applying service patches on virtual nodes

*Kerberos is a protocol for authenticating service requests between trusted hosts across an untrusted network, such as the internet. Kerberos support is built in to all major computer operating systems, including Microsoft Windows, Apple macOS, FreeBSD and Linux

Monitoring Layer



The **monitoring layer consists of a number of monitoring systems**. The system remains **automatically aware of all the configurations and functions of different operating systems and hardware**. It also provide the facility of machine communication with the help of a monitoring tool through high-level protocols, such as Extensible Markup Language (XML). **Monitoring systems also provide tools for data storage and visualization**.

Tools for Monitoring :- Ganglia and Nagios

Analytics Engine

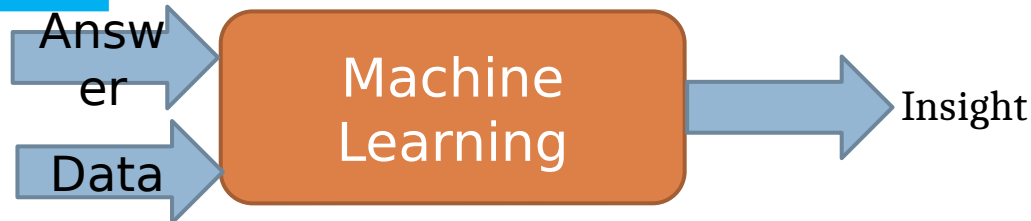


The role of an **analytics engine is to analyze huge amounts of unstructured data**. This type of analysis is related to **text analytics and statistical analytics**. Analytics is broadly the bridge between data and insight. At the outset, the need is to understand (or describe) the data, which is usually done using rule based approach. After which to predict what can happen in order to get some insight and make good decisions usually done with learning based approach.

Rules based approach



Learning based approach



Note: Contributed by Manoj Kumar Lenka (Roll No: 1605456, 3rd year KIIT CSE student)

Analytics Engine cont'd



Rules based approach

Rules are defined based on which output is generated from the data. It is useful to extract the structural/syntactic meaning of the data. Examples are:

- ☐ Finding frequency of words. ☐ Find particular words in sentences
- ☐ Check if a set of words follows a particular pattern, etc.. ☐ Length of sentences.

Learning based approach

The rules are not explicitly programmed and the data is self-learnt. It has two phases namely training phase and the inference (testing) phase. In the training phase several features (what we have) and their corresponding labels (what to predict) are given to the system, based on which the system learns the relation between them by training its parameters. In the inference phase the trained system is given just the features and using the learned parameters it predicts the output. Learning based systems mainly consists of Machine Learning (ML) models. Examples are:

- ☐ Recommendation Engine ☐ Medical diagnosis
- ☐ Image/Speech Reorganization ☐ Prediction, Extraction

Note: Contributed by Manoj Kumar Lenka (Roll No: 1605456, 3rd year KIIT CSE student)

Analytics Engine cont'd



Some statistical and numerical methods used for analyzing various unstructured data sources are:

- ☐ **Natural Language Processing**
- ☐ **Text Mining**
- ☐ **Machine Learning**
- ☐ **Linguistic Computation**
- ☐ **Search and Sort Algorithms**
- ☐ **Syntax and Lexical Analysis**

Some examples of different types of unstructured data that are available as large dataset include the following:

- ☐ Machine generated data such as RFID feeds and weather data
- ☐ Documents containing textual patterns
- ☐ Data generated from application logs about upcoming or down time details or about maintenance and upgrade details

Analytics Engine cont'd



The following types of engines are used for analyzing Big Data:

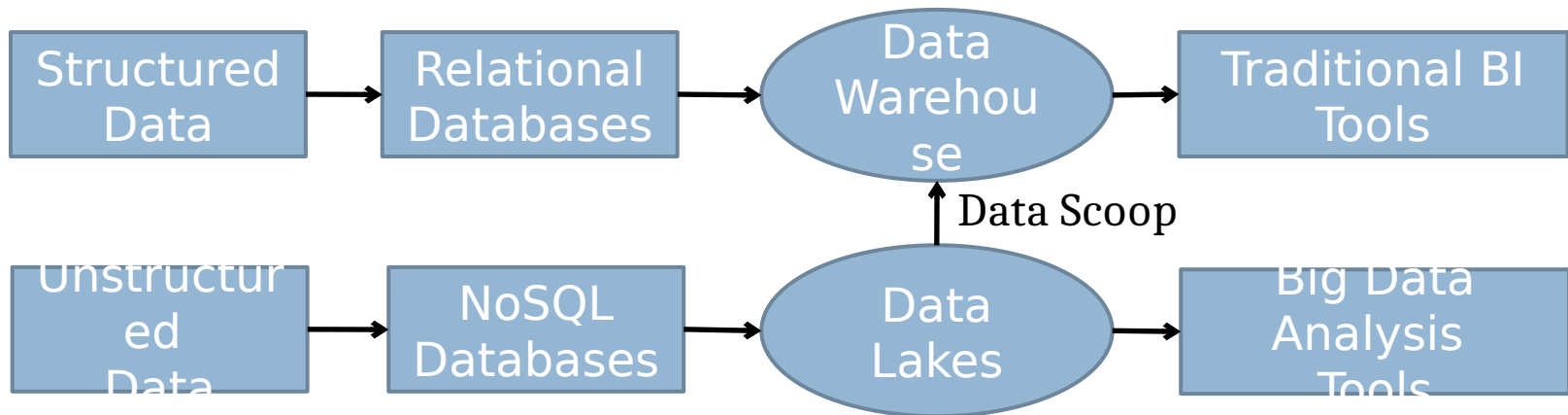
- ❑ **Search engines:** Big data analysis requires **extremely fast search engines with iterative and cognitive data discovery mechanisms for analyzing huge volumes of data.** This is required because the data loaded from various sources has to be indexed and searched for Big Data analytics processing.
- ❑ **Real-time engines:** **Real-time application in modern era generating data at a very high speed and even a few-hour old data becomes obsolete and useless as new data continues to flow in.** Real-time analysis is required in the Big Data environment to analyze this type of data. For this purpose, real-time engines and NoSQL data stores are used.

Visualization Layer



It handles the task of interpreting and visualizing Big Data. Visualization of data is done by data analysts and scientists to have a look at the different aspects of the data in various visual modes. **It can be described as viewing a piece of information from different perspectives, interpreting it in different manners,** trying to fit in different types of situations and deriving different types of conclusions from it.

Some examples of visualization tools are **Tableau, Clickview, Spotfire, MapR, and revolution R.** These tools work on the top of the traditional components such as reports, dashboards, scorecards, and queries.



Visualization Layer cont'd



A data lake is a centralized repository that allows to store all your structured and unstructured data at any scale. It can store data as-is, without having to first structure the data, and run different types of analytics—from dashboards and visualizations to big data processing, real-time analytics, and machine learning to guide better decisions.

Characteristics	Data Warehouse	Data Lake
Data	Relational from transactional systems, operational databases, and line of business applications	Non-relational and relational from IoT devices, web sites, mobile apps, social media, and corporate applications
Schema	schema-on-write	schema-on-read
Users	Business analysts	Data scientists, and Business analysts
Analytics	Batch reporting, BI and visualizations	Machine Learning, Predictive analytics, and data discovery

Virtualization



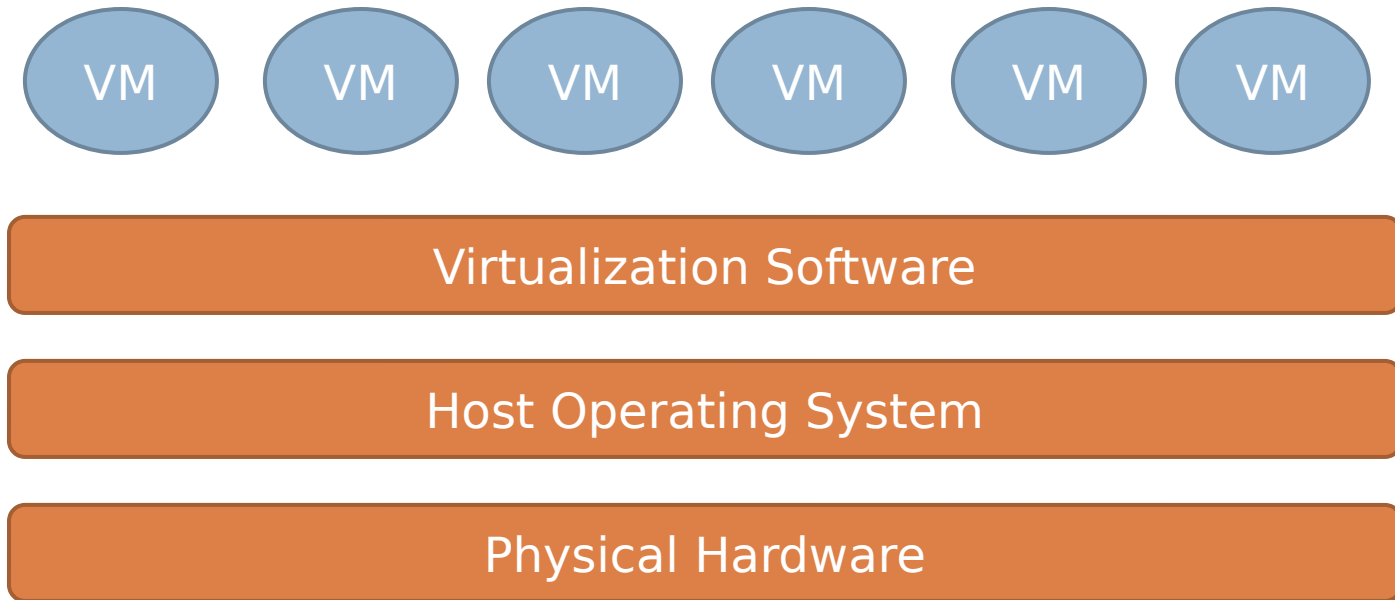
In computing, **virtualization** refers to the act of creating a virtual (rather than actual) version of something, such as virtual computer hardware platforms, storage devices, operating systems and computer network resources.

In other words, **it is a process to run the images of multiple operating systems on a physical computer**. The images of the **operating systems** are called **virtual machines (VMs)**. **A virtual machine (VM) is basically a software representation of a physical machine that can execute or perform the same functions as the physical machines.**

Although virtualization is not a requirement for Big Data analysis, but works efficiently in a virtualized environment.

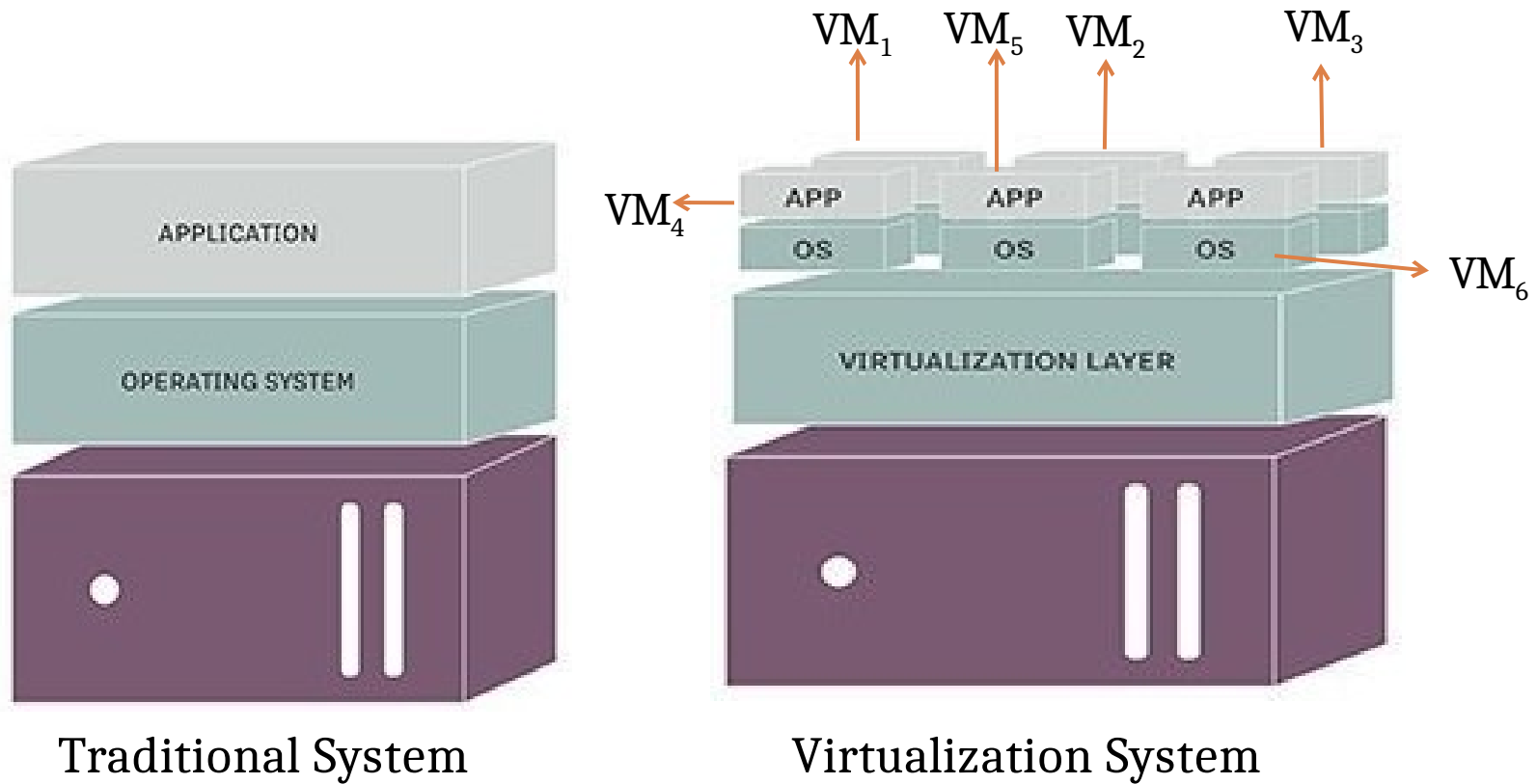
Server virtualization is the process in which multiple operating systems (OS) and applications run on the same server at the same time, as opposed to one server running one operating system. If that still seems confusing, think of it as one large server being cut into pieces. **The server then imitates or pretends to be multiple servers on the network when in reality it's only one.** This offers companies the capability to utilize their resources efficiently and lowers the overall costs that come with maintaining servers.

Virtualization Environment



The **operating system that runs as a virtual machine is known as the guest, while the operating system that runs the virtual machine is known as the host.** A guest operating system runs on a hardware virtualization layer, which is at the top of the hardware of a physical machine.

Traditional vs. Virtualization System



Source: researchgate.net

Basic Features of Virtualization

- ❑ **Partitioning:** Multiple applications and operating systems are supported by a single physical system by partitioning the available resources.
- ❑ **Isolation:** Each VM runs in an isolated manner from its host physical system and other VMs. If one VM crashes, the other VMs and the host system are not affected.
- ❑ **Encapsulation:** Each VM encapsulates its state as a file system i.e. it can be copied or moved like a simple file.
- ❑ **Interposition:** Generally in a VM, all the new guest actions are performed through the monitor (Hypervisor). A monitor can inspect, modify or deny operations such as compression, encryption, profiling, and translation. Such types of actions are done without the knowledge of the operating system.

Types of Virtualization



Virtualization can be utilized in many different ways and can take many forms aside from just server virtualization. The main types include application, desktop, user, storage and hardware.

- ❑ **Server virtualization:** a single physical server is partitioned into multiple virtual server. Each virtual server has its own hardware and related resources, such as RAM, CPU, hard drive and network controllers. A thin layer of software is also inserted with the hardware which consists of a VM monitor, also called hypervisor and it's to manage the traffic between VMs and the physical machine.
- ❑ **Application virtualization:** allows the user to access the application, not from their workstation, but from a remotely located server. The server stores all personal information and other characteristics of the application, but can still run on a local workstation. Technically, **the application is not installed, but acts like it is.**
- ❑ **Data and Storage virtualization:** is the process of grouping the physical storage from multiple network storage devices so that it acts as if it's on one storage device.

Types of Virtualization cont'd



- ❑ **Network virtualization:** It means using virtual networking as a pool of connection resources. During implementing such virtualization, physical network is to be relied for managing traffic between connections. As many as number of virtual networks can be created from a single physical implementation.
- ❑ **Processor and Memory virtualization:** It decouples memory from the server and optimize the power of the processor and maximizes its performance. Big data analysis needs systems to have high processing power (CPU) and memory (RAM) for performing complex computations. These computations can take a lot of time in case CPU and memory resources are not sufficient. Processor and Memory virtualization thus can increase the speed of processing and the analysis results sooner.

Benefits of Virtualization



Virtualization provides several benefits for companies, including:

- ☐ Greater efficiency and company agility
- ☐ Ability to more-effectively manage resources
- ☐ Increased productivity, as employees access the company network from any location
- ☐ Data stored on one centralized server results in a decrease in risk of lost or stolen data

Not only is it beneficial for companies, but virtualization provides several benefits for data centers as well, including:

- ☐ Cutting waste and costs associated with maintaining and cooling its servers by maximizing the capabilities of one server
- ☐ Allows data centers to be smaller in size, resulting in overall savings due to a reduction in:
 - ☐ Energy needed
 - ☐ Hardware used
 - ☐ Time and money needed for maintenance

**THANK
YOU!**