PAGE NO.:
DATE: / /

Cloud Computing Architecture:

Cloud Computing architecture comprises of many cloud components, which are loosely coupled. We can broadly divide the cloud architecture into two parts.

→ Front End
→ Back End

Each of the ends is connected through a network, usually Internet.

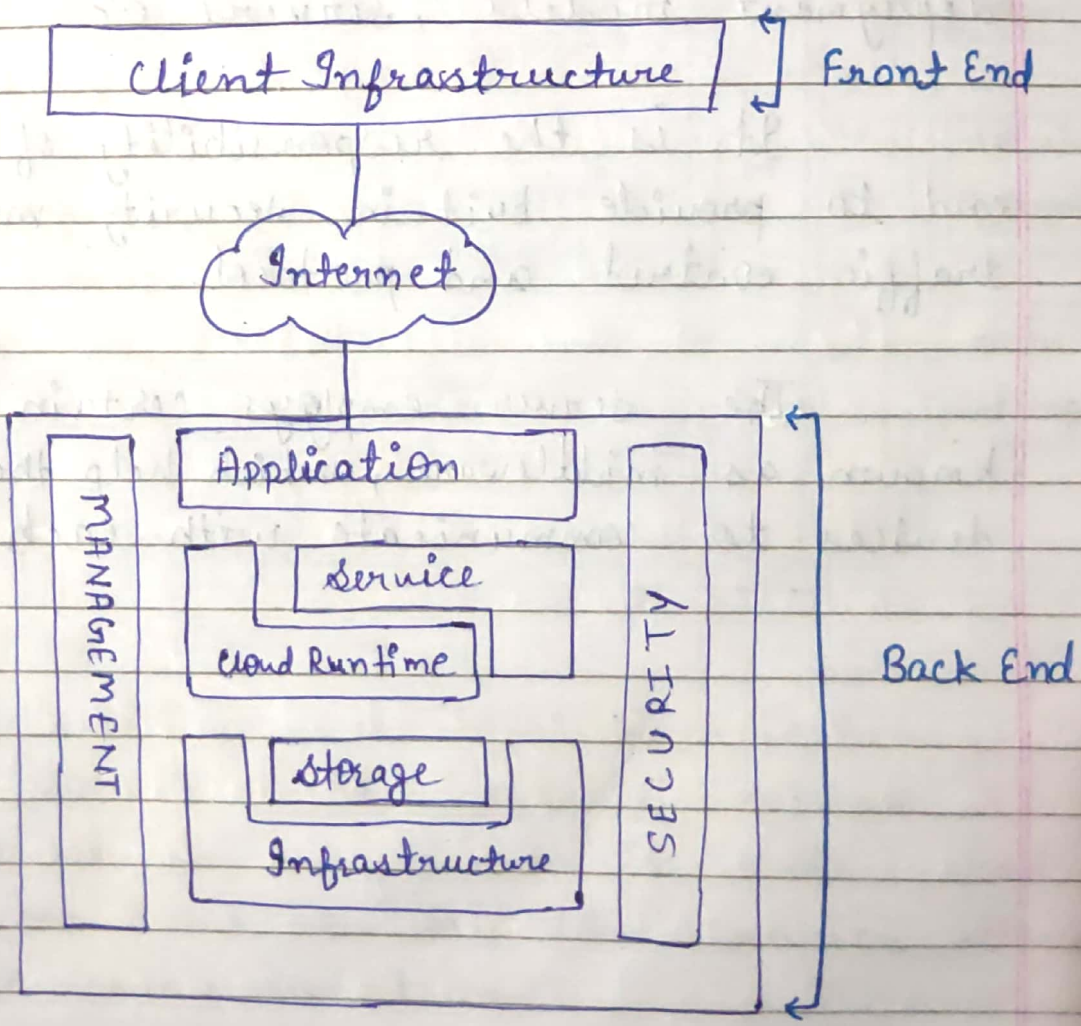


Fig: Architecture of Cloud Computing

PAGE NO.:
DATE: / /

Front End- The front end refers to the client part of cloud computing system. It consists of interfaces and applications that are required to access the cloud computing platforms.
Example - Web Browser

Back End- The back end refers to the cloud itself. It consists of all the resource required to provide cloud computing services. It comprises of huge data storage, virtual machines, security mechanism, services deployment models, servers etc.

It is the responsibility of the back end to provide buit-in security mechanism traffic control and protocol.

The server employs certain protocols known as middleware, which help the connected devices to communicate with each other.

PAGE NO.:
DATE: / /

Cloud References Model -

Cloud Computing conceptual reference model identifies the major actors, their activities and functions in cloud computing.

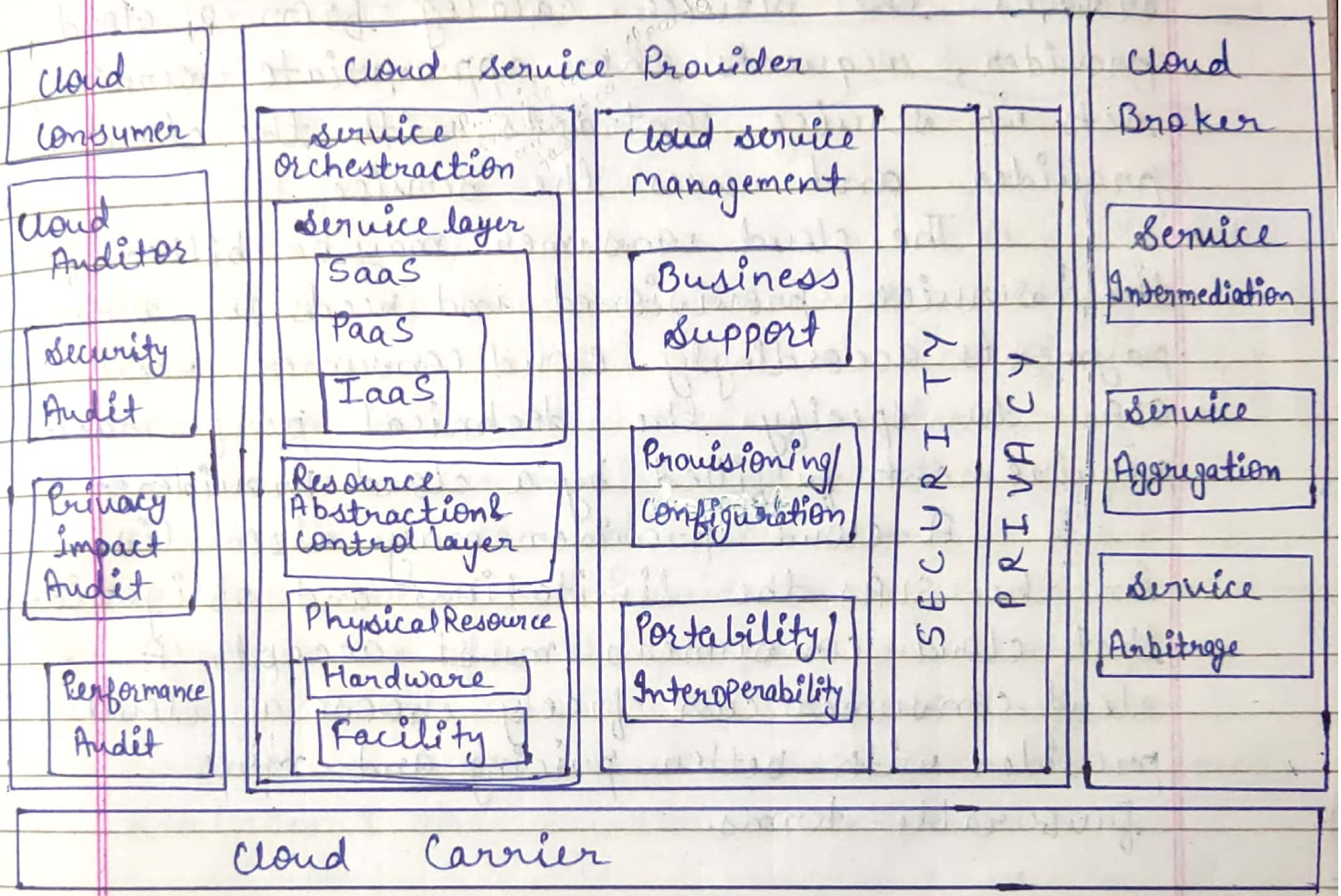


Fig: Cloud Reference Model Architecture

The NIST cloud computing reference architecture defines five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker.

Each actor is an entity (a person or organization) that participates in a transaction or process and/or performs tasks in cloud computing.

Teacher's Signature

PAGE NO.:
DATE: / /

① Cloud Consumer – The cloud consumer is the principal stakeholder for the cloud computing service. A cloud consumer represents a person or organization that maintains a business relationship with and uses the services from a cloud provider. A cloud consumer browses the service catalog from a cloud provider, requests the appropriate service, sets up service contracts with the cloud provider and uses the service.

The cloud consumer may be billed for the service provisioned and needs to arrange payments accordingly. Cloud consumers need SLAs to specify the technical performance requirement fulfilled by a cloud provider.

A cloud provider may also list in the SLAs the limitations and obligation that cloud consumers must accept. A cloud consumer can freely choose a cloud provider with better pricing and more favourable terms.

② Cloud Providers – A cloud provider is a person or an organization; or an entity responsible for making a service available to interested parties.

A cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services to the cloud consumers through network access.

Teacher's Signature

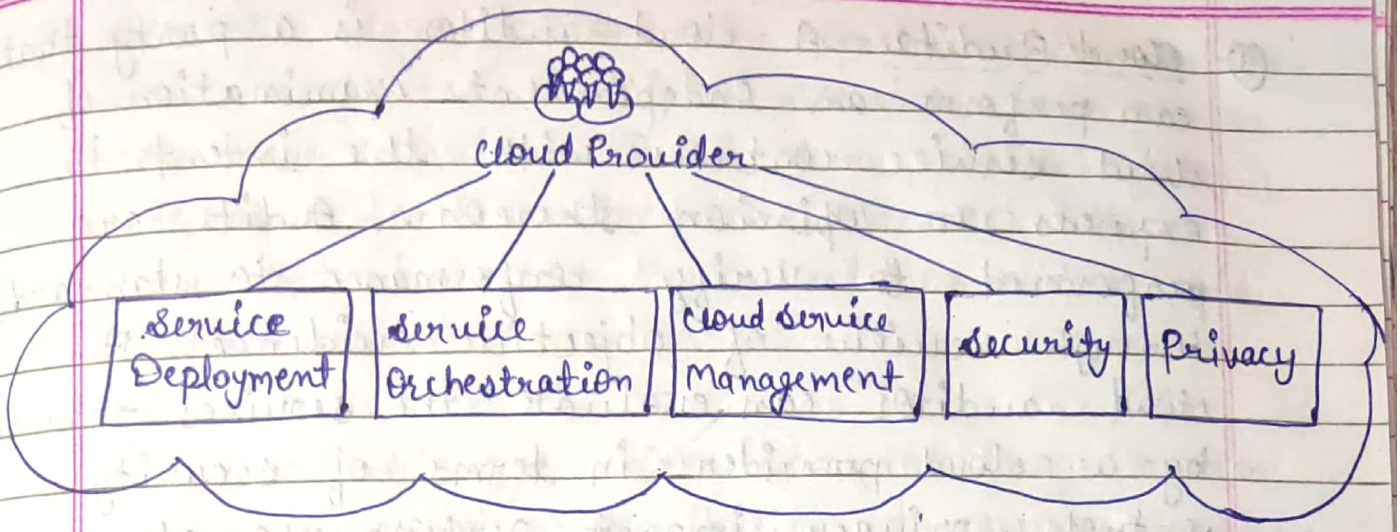


Fig: Cloud Provider

→ For SaaS, the cloud provider deploys, configure, maintains and updates the operation of the s/w application on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumer.

→ For Paas, the cloud Provider manages the computing infrastructure for the platform and runs the cloud s/w that provides the components of the platform, such as runtime s/w execution stack, databases & other middleware components.

For Iaas the cloud Provider acquires the physical computing resources underlying the service, including the servers, n/w, storage and hosting infrastructure.

PAGE NO.:
DATE: / /

③ Cloud Auditor- A cloud auditor is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon. Audits are performed to verify conformance to standards through review of objective evidence. A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy, impact, performance etc.

Auditing is especially important for federal agencies as "agencies should include a contractual clause enabling third parties to assess security controls of cloud providers".

④ Cloud Broker- A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationship between cloud provider and cloud consumers.

A cloud broker can provide services in three categories:-

ⓐ Service Intermediation- A cloud broker enhances a given services by improving some specific capability and providing value-added services to cloud consumers. The improvement can be managing access to cloud services, identity management, performance reporting, enhanced security etc.

(b) Service Aggregation - A cloud broker combines and integrates multiple services into one or more new services. The broker provides data integration and ensures the secure data movement between the cloud consumer and multiple cloud provider.

(c) Service Arbitrage - Services arbitrage means a broker has the flexibility to choose services from multiple agencies.

(5) Cloud Carrier - A cloud carrier acts as an intermediary that provides connectivity & transport of cloud services between cloud consumers and cloud providers.

Cloud carriers provide access to consumers through network, telecommunication and other access devices. For example - cloud consumers can obtain cloud services through network access devices, such as computers, laptop mobile phones, mobile internet devices etc.

PAGE NO.:
DATE: / /

Cloud Computing Layers-

① Physical layer-

→ Foundation layer of the cloud infrastructure
→ specifies entities that operate at this layer:
- Compute systems, network devices & storage devices
- Operating environment, protocol, tools & processes
→ Functions of physical layer
- Executes requests generated by virtualization and control layer
→ Examples of requests from the layers include storing data on the storage devices, performing communication among compute systems, executing programs on a compute systems, creating backup copy of data or executing security policy to block an unauthorized activity.

② Virtual layer-

→ Deployed on the physical layer
→ specifies entities that operate at this layer:
virtualization s/w
Ressource Pools
Virtual resources
→ Functions of virtual layer-
- Abstract physical resources and makes them appear as virtual resources
→ Enables multitenant environment, thereby improving utilization. Improved utilization of physical resources results in increased return of investment (ROI) on the infrastructure entities.

Teacher's Signature

- Executes the requests generated by control layer, ex of requests generated by control layer includes creating pools of resources & creating virtual resources.

③ Control layer -
→ Deployed either on virtual layer or on physical layer
→ specifies entities that operate at this layer -
→ control s/w
→ Functions of control layer:
→ Enables resource configuration & resource pool configuration.
→ Enables resource Provisioning.
→ Executes requests generated by service layer
→ Exposes resources to and supports the service layer
→ Collaborates with the virtualization s/w & enables
→ Resource pooling & creating virtual resources.
→ Dynamic allocation of resources.
→ Optimizing utilization of resources.

④ Service Orchestration layer -
→ Specifies the entity that operate at this layer:
→ Orchestration s/w
→ Functions of this layer -
→ Provides workflows for executing automated tasks to accomplish a desired outcome.
"Workflow refers to a series of inter-related tasks that perform a business operation."

Teacher's Signature

The orchestration s/w enables this automated arrangement, coordination and management of the tasks. This helps to group & sequence tasks with dependencies among them into a single, automated workflow.

→ Associated with each service listed in the service catalog, there is an orchestration workflow defined.

"When a consumer selects a service from the service catalog, an associated workflow in the orchestration layer is triggered."

Based on this workflow, the orchestration s/w interacts with various entities (from control layer, business continuity function, security function and service management function) to invoke the provisioning tasks to be executed by the entities.

⑤ Service layer -

→ Consumers interact & consume cloud resources via this layer.

→ Specifies the entities that operate at this layer:

- → Service catalog
- → Self service portal

→ Functions of service layer.

→ Stores information about cloud services in service catalog and presents them to the consumers.

A service catalog is a database of information about the cloud services offered by a service Provider. The service catalog includes a variety of information about the services, including description of the services, the types of services, cost, supported SLAs, security mechanisms & so on.

Enables consumers to access & manages cloud services via a self-service portal. The provisioning & management requests are passed on to the orchestration layer, where the orchestration - to fulfill the requests - are defined.

Types of Cloud-

① Public Cloud - Public Cloud allows the accessibility of system and services easily to general public. Ex- Amazon, IBM, Microsoft, Google, Rackspace etc.

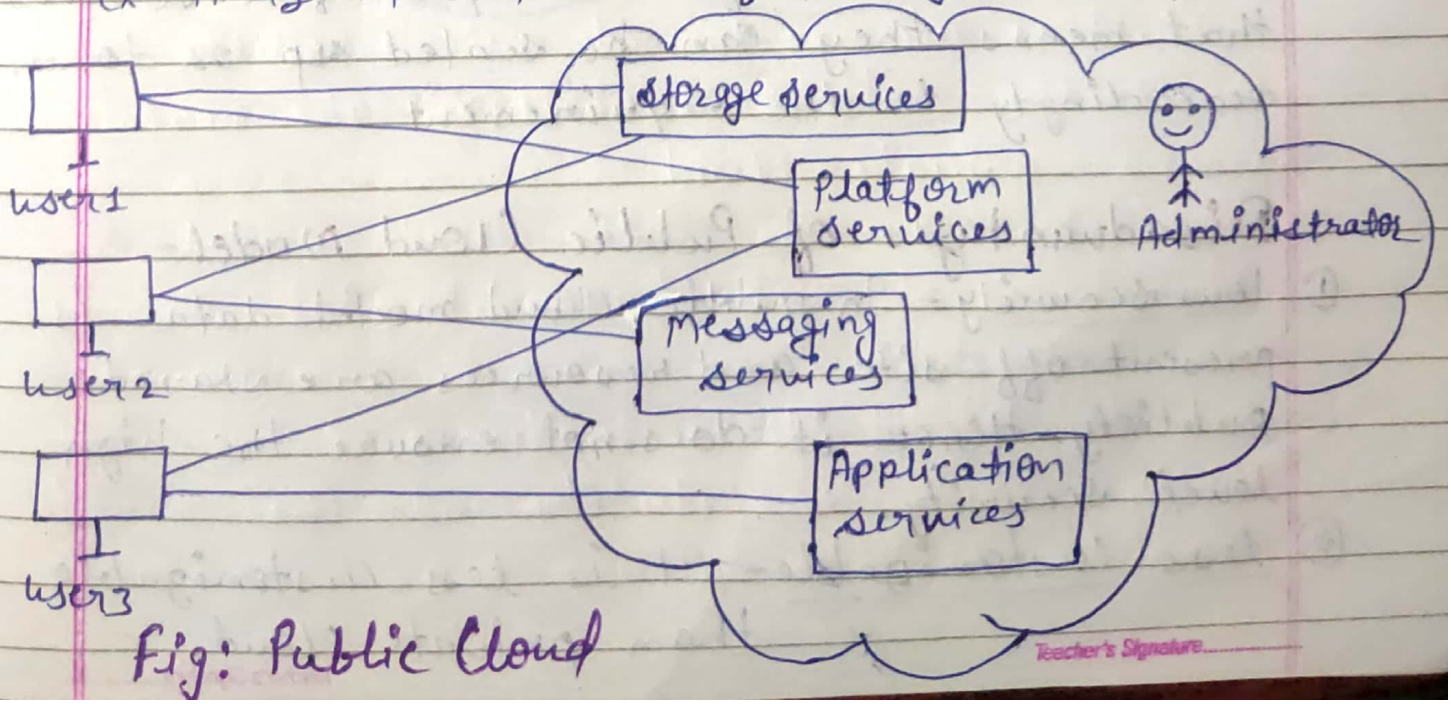


Fig: Public Cloud

Teacher's Signature.....

DATE: / /

Advantages of Public Cloud Model:

① Low Cost - Public cloud is having low cost as compared to private or hybrid cloud, because it shares same resources with large number of consumer.

② Reliable - Public cloud provides large number of resources from different locations, if any of the resource fail, public cloud can employ another one.

③ Flexible - It is very easy to integrate public cloud and hence it gives flexible approach to consumers.

④ Location Independent - It ensures the Independency of location, because public cloud services are delivered through internet.

⑤ High Scalability - Cloud resources are available as per the demand from the pool of resources that means they can be scaled up or down accordingly to the requirement.

Disadvantages of Public Cloud Model -

① Low Security - In public cloud model data is present off-site and resources are shared publicly. Hence it does not ensure the high level security.

② less Customizable - It is less customizable than private cloud.

Teacher's Signature.....

② Private cloud - The Private cloud allows the accessibility of systems and services within the organization. Private cloud is operated only within a particular organization. But it will be managed internally or by third party.

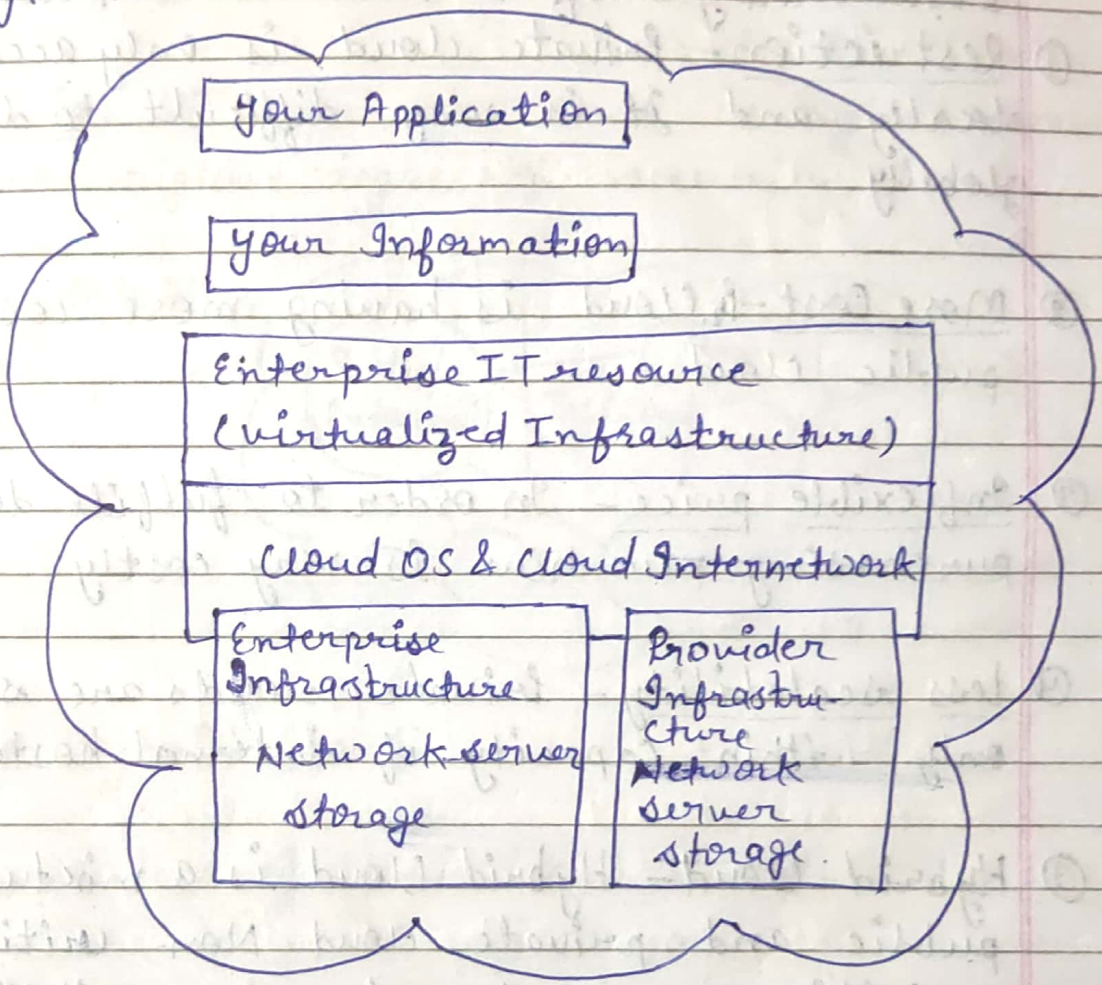


Fig: Private Cloud

Advantages of Private Cloud Model -

① High security and privacy - Private cloud resources are shared from distinct Pool of resources & hence highly secured.

Teacher's Signature

PAGE NO.:
DATE: / /

② More Control- Private clouds have more control on its resources and h/w than public cloud because it is accessed only within the boundary of an organization.

Disadvantages of Private cloud model-

① Restriction- Private cloud is only accessible locally and it is very difficult to deploy globally.

② More Cost- Cloud is having more cost than public cloud.

③ Inflexible price- In order to fulfill demands, purchasing new h/w is very costly.

④ less scalability- Private clouds are scaled only within capacity of internal hosted resources

③ Hybrid Cloud- Hybrid Cloud is a mixture of public and private cloud. Non-critical activities are performed using public cloud while the critical activities are performed using private cloud.

Advantages of Hybrid Cloud Model-

① Scalability- It offers features of both the public cloud scalability and the private cloud scalability

② Flexibility - It offers secure resources and scalable public resources.

③ Cost Efficiency - Public clouds are most cost effective than private ones. Therefore, hybrid clouds can be cost saving.

④ Security - The private cloud in hybrid cloud ensures higher degree of security.

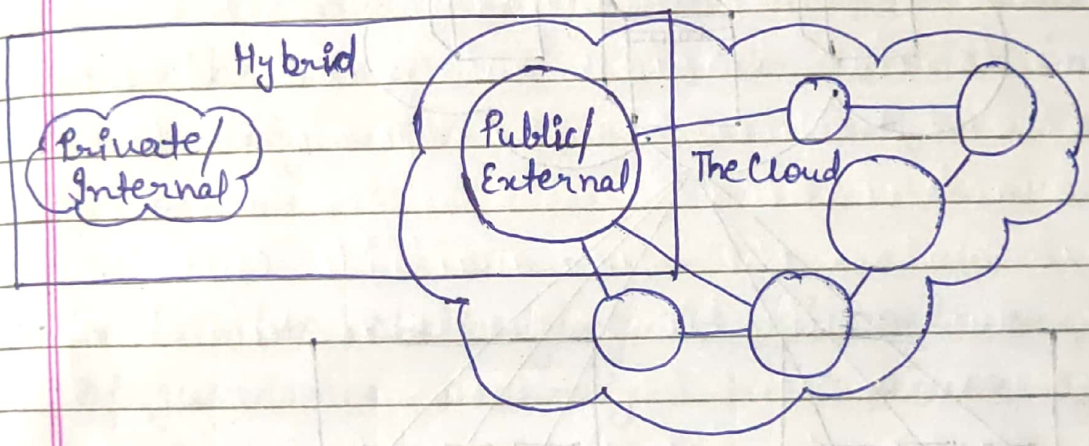


Fig: Hybrid Cloud

Disadvantage of Hybrid Cloud Model -

① Networking Issues - Networking becomes complex due to presence of private and public cloud.

② Security Compliance - It is necessary to ensure that cloud services are compliant with security policies of the organization.

③ Infrastructure Dependency - The hybrid cloud model is dependent on internal IT infrastructure, therefore it is necessary to ensure redundancy across data centers.

Teacher's Signature.....

④ Community Cloud- It allows system and services to be accessible by group of organization. It shares the infrastructure between several organizations from a specific community. It may be managed internally by organization or by the third-party.

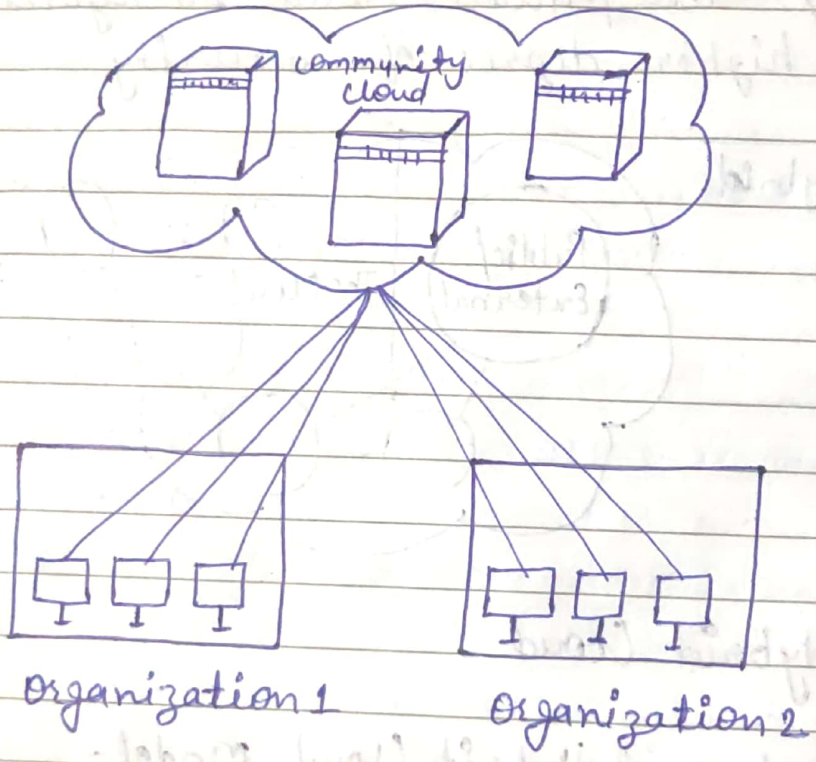


Fig: Community Cloud

Advantages of Community model-

① Cost Effective- Community cloud offers same advantages as that of private cloud at low cost.

② Sharing Among Organizations - Community cloud provides an infrastructure to share cloud

resources and capabilities among several organizations.

Security- The community cloud is comparatively more secure than the public cloud but less secured than the private cloud.

Issues in Community cloud Model:

→ Since all data is located at one place, one must be careful in storing data in community cloud because it might be accessible to others.

→ It is also challenging to allocate responsibilities of governance, security & cost among organizations.

Services Model of Cloud Computing:

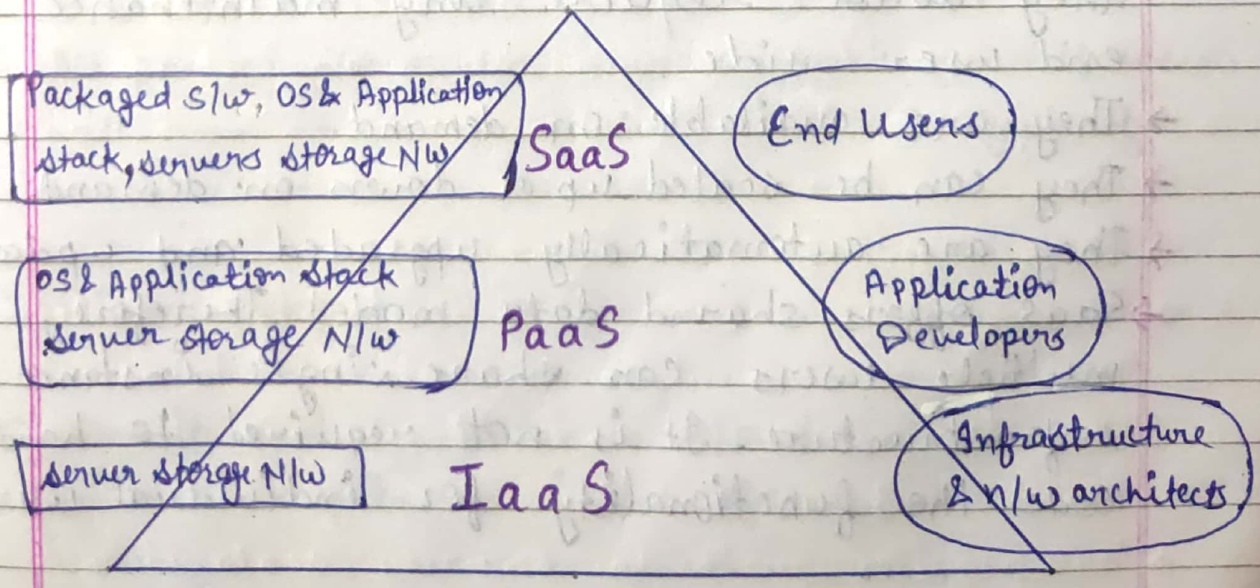


Fig: Cloud Service Model

Teacher's Signature

PAGE NO.:
DATE: / /

① Software-as-a Service (SaaS)-
SaaS model allows to provide s/w application as a service to the end user. It refers to a s/w that is deployed on a host service & is accessible via Internet. There are several SaaS applications as-
→ Billing and invoicing system
→ Customer Relationship Management (CRM) applications
→ Help desk applications.
→ Human Resource solutions
SaaS is also known as "On Demand Software"
Ex- Microsoft office365, G-mail, Google talk, oracle etc

Characteristics-
→ SaaS makes the s/w available over the internet
→ The s/w applications are maintained by the vendor
→ The license to the s/w may be subscription based or usage based. And it is billed on recurring basis.
→ SaaS applications are cost-effective since they donot require any maintenance at end user side.
→ They are available on demand.
→ They can be scaled up or down on demand.
→ They are automatically upgraded and updated
→ SaaS offers shared data model. Therefore, multiple users can share single instance of infrastructure. It is not required to hard code the functionality for individual users.

PAGE NO.:
DATE: / /

Advantages of SaaS -

① SaaS is easy to buy - SaaS allows organization to access business functionality at a low cost which is less than licensed applications. SaaS providers generally pricing the application using a subscription fee, most commonly a monthly or annually fee.

② Less h/w required for SaaS - The s/w is hosted remotely, so organizations don't need to invest in additional h/w.

③ Low maintenance required for SaaS - SaaS removes the necessity of installation, set-up & often daily upkeep & maintenance for org. Initial set up cost for SaaS is typically less than the enterprise s/w. SaaS vendors actually pricing their applications based on some usage parameters, such as no of users using the app.

④ No special s/w or h/w version required - All users will have the same version of s/w & typically access it through the web browser.

Disadvantages of SaaS -

① Security - Actually data is stored in cloud, so security may be an issue for some users.

Teacher's Signature.....

PAGE NO.:
DATE: / /

② Latency issue - Because the data & application are stored in cloud at a variable distance from the end user, so there is a possibility that there may be more latency while interacting with the application than a local deployment. So Saas model is not suitable for applications whose demand response times are in milliseconds.

③ Total dependency on Internet - without internet connection, most Saas applications are not usable.

② Platform-as-a Service -
Paas is a developer programming platform which is created for the programmer to develop, test, run and manage the application. A developer is able to write the application as well as deploy it directly into this layer easily. In Paas, back end scalability is handled by the cloud service provider & the end user doesn't have to worry about to manage the infrastructure.

Characteristics -
→ Paas offers browser based development environment. It allows the developer to create database & edit the application code either via API or point-and-click tools.
→ Paas provides built-in security, scalability & web service interfaces.

→ PaaS provides built-in tools for defining workflow approval processes, and business rules.
It is easy to integrate PaaS with other applications on the same platform.

Advantages of PaaS-

① Simplified Development- Developers can focus on development & innovation without worrying about infrastructure.

② Lower risk - NO requirements of up-front investment in h/w & s/w. Developers only need a PC & an internet connection to start building applications.

③ Prebuilt business functionality - Some PaaS vendors also provide already defined business functionality so that users can avoid building everything from very scratch & hence can directly start the projects only.

④ Instant community - PaaS vendors frequently provides online communities where developer can get the ideas, share experiences & seek advice from others.

⑤ Scalability - Applications deployed can scale from one to thousands of users without any changes to the applications.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Disadvantages of Paas -

① Vendor lock-in - One have to write the applications according to the platform provided by Paas vendor so migration of an application to another Paas vendor would be a problem.

② Data Privacy - Corporate data, whether it can be critical or not, will be private so if it is not located with in the walls of the company there can be a risk in terms of privacy of data.

③ Integration with the rest of the system app - It may happen that some applications are local and some are in cloud. So there will be chances of increased complexity when we want to use data which is in the cloud with the local data.

Top Vendors who are providing Paas cloud computing platform -

① Google App Engine (GAE)
② Salesforce.com
③ Windows Azure
④ AppFog
⑤ Openshift
⑥ Cloud Foundary from VMware

PAGE NO.:
DATE: / /

③ Infrastructure-as-a Service (IaaS)-

IaaS is one of the layers of cloud computing platform wherein the customer organization outsources its IT infrastructure such as servers, networking, processing, storage, virtual machines and other resources. Customers access these resources over internet i.e. cloud computing platform, on a pay-per-use model.

IaaS earlier called Hardware as a Service (HaaS)

Characteristics-

→ Virtual machines with pre-installed Sw.
→ Virtual machines with pre-installed O.S.
→ On demand availability of resources.
→ Allows to store copies of particular data at different location.
→ The computing resources can be easily scaled up and down.

Advantages-

① You can dynamically choose a CPU, memory & storage configuration as per your needs.
② You easily access the vast computing power available on IaaS cloud platform.
③ You can eliminate the need of investment in rarely used IT h/w.
④ IT infrastructure will be handled by the IaaS cloud computing platform vendors.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Disadvantages -

1. Data privacy can be a risk for an organization whose data is stored in Iaas cloud computing platform
2. Iaas cloud computing platform model is fully dependent on internet availability.
3. It is also dependent on the availability of virtualization services.

Top Vendors who are praviding Iaas cloud computing platform -

1. Amazon Web services - Elastic Compute Cloud (EC2), MapReduce, Route 53, Virtual Private cloud etc.
2. Netmagic solution - Netmagic Iaas cloud.
3. Rackspace - Cloud servers, cloud files, cloud sites etc.
4. Reliance Communications - Reliance Internet Data center
5. Sify Technology - Sify Iaas
6. Tata Communications - Insta Compute

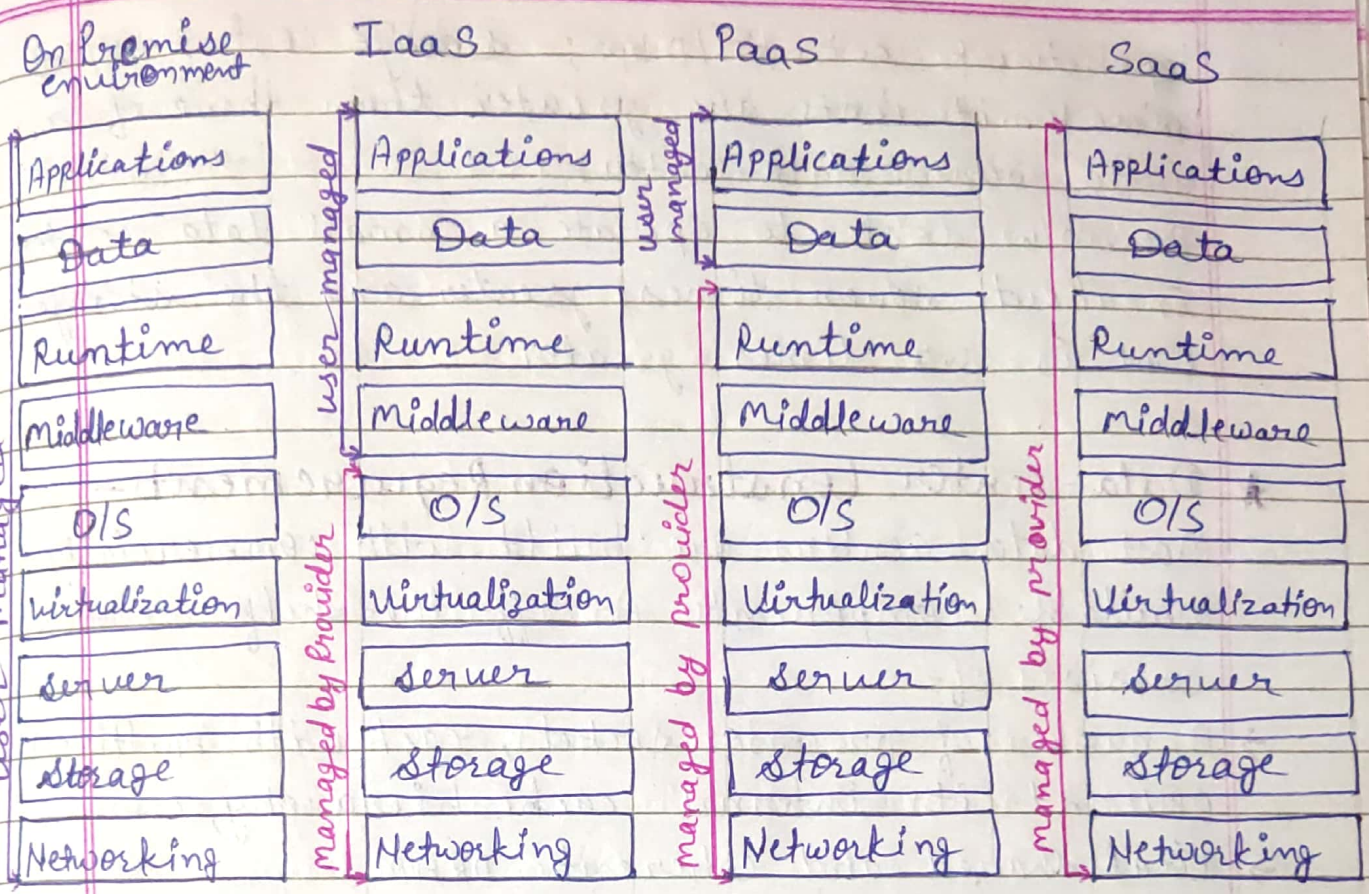


Fig: Service Model of Cloud Computing

✳ Data-Center Design and Interconnection Networks-

A data center is often built with a large number of servers through a huge interconnection network.

① Warehouse-scale Data-Center Design- The cloud is built on massive datacenters. Such a datacenter can house 400,000 to 1 million servers. A small datacenter could have 1000 servers. The data centers are built economics of scale- meaning lower unit cost for larger data centers.

The approximate monthly cost to operate a huge 400-server data center is estimated

Teacher's Signature

by network cost $13/Mbps; storage cost $0.4/GB. These unit costs are greater than those of a 1000-server data center.
The n/w cost to operate a small data center is about seven times greater and the storage cost is 5.7 times greater.

* **Data-Center Construction Requirements-**
Most data centers are built with commercially available components. An off-the-shelf server consists of-
→ A number of processor sockets, each with multicore CPU and its internal cache hierarchy,
→ local shared and coherent DRAMs
→ a number of directly attached disk drives.

The DRAM & disk resources with in the rack are accessible through first-level rack switches & all resources in all racks are accessible via a cluster level switch.

Consider a data center with 2000 servers each with 8 GB of DRAM and four 1 TB disk drives. Each group of 40 servers is connected through a 1 Gbps link to a rack-level switch that has an additional eight 1 GBps ports used for connecting the rack to the cluster-level switch.

The bandwidth available from local disk is 200 MB/s, whereas the bandwidth from off-rack disk is 25 MB/s via shared rack uplinks.

Teacher's Signature

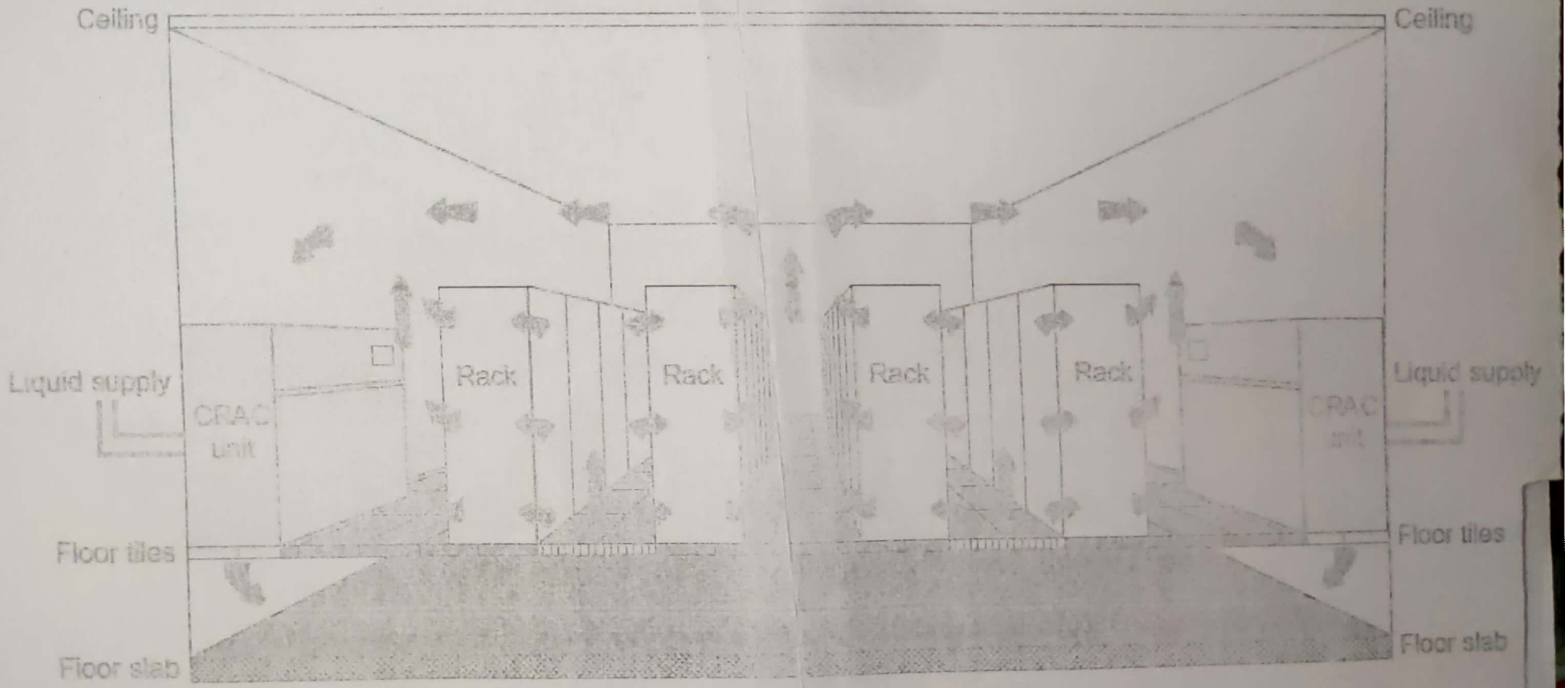


Fig- Cooling system in a raised floor data center with hot-cold air circulation supporting water heat exchange facilities

PAGE NO.:
DATE: / /

The total disk storage in the cluster is almost 10 million times larger than local DRAM. In a very large scale data center, components are relatively cheaper.

With a scale of thousands of server, it may be possible that any failure can occure in h/w; for ex- CPU failure, disk I/O failure and n/w failure. Also, some failure are brough on by s/w. The service & data should not be lost in a failure situation. Reliability can be achieved by redundant h/w. The s/w must keep multiple copies of data in different locations & keep the data accessible while facing h/w or s/w errors.

★ Cooling System of a Data Center Room –

The figure shows the layout & cooling facility of a warehouse in a data center. The data center room has raised floors for hiding cables, power lines, & cooling supplies. The raised floor has a steel grid resting on stanchions about 2-4 ft above the concrete floor.

The under-floor area is often used to route power cables to racks, but its primary use is to distribute cool air to the server rack.

The CRAC (Computer Room Air Conditioning) unit pressurizes the raised floor plenum by blowing cold air into the plenum.

The cold air escapes from the plenum through perforated tiles that are placed in front of server racks.

Racks are arranged in long aisles that alternate between cold aisles & hot aisles to avoid mixing hot & cold air.

The hot air produced by the servers circulates back to the intakes of the CRAC units that cool it & then exhaust the cool air into the raised floor plenum again.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Data-Center Interconnection Network-

A critical core design of a data center cluster. This n/w design must meet five special requirements: low latency, high bandwidth, low cost, message passing interface (MPI) communication support and fault tolerance. The design of an inter-server n/w must satisfy both point-to-point & collective communication patterns among all server nodes. Specific design consideration are as follows-

★ **Application Traffic Support-** The n/w topology should support all MPI communication pattern. Both point-to-point & collective MPI communication must be supported. The n/w should have high bisection bandwidth to meet this requirement.

For ex, one-to-many communications are used for supporting distributed file access. One can use one or a few servers as metadata master servers which need to communicate with slave server nodes in the cluster. The underlying n/w structure should support various n/w traffic patterns demanded by user application.

★ **Network Expandability-** The interconnection n/w should be expandable. With thousands or even hundreds of thousands of server nodes, the cluster n/w interconnection shoud be allowed to expand once more servers are added to the data center. The n/w topology shoud be

Teacher's Signature

restructured while facing such expected growth in the future. Also the n/w should be designed to support load balancing and data movement among the servers. None of the links should become a bottleneck that slows down application performance. The topology of the interconnection should avoid such bottleneck.

Fault Tolerance and Graceful Degradation -

The interconnection n/w should provide some mechanism to tolerate link or switch failure. In addition, multiple paths should be established between any two server nodes in a data center. Fault tolerance of servers is achieved by replicating data and computing among redundant servers. Similar redundancy technology should apply to the n/w structure. There should be no critical paths or critical points which may become a single point of failure that pulls down the entire system.

* **Switch-centric Data-Center Design -**

There are two approaches to building data-center scale networks:

→ switch centric
→ server centric

In a switch centric n/w the switches are used to connect the server nodes. The switch centric design does not affect the server side. NO modification to the server are needed.

Teacher's Signature

PAGE NO.:
DATE: / /

Example:- Fat-tree Interconnection Network

The fat-tree topology is applied to interconnect the server nodes. The topology is organized into two layers.
Server nodes are in the bottom layer, and edge switches are used to connect the nodes in the bottom layer.
The upper layer aggregates the lower layer edge switches.

A group of aggregation switches, edge switches and their leaf nodes form a pod. Core switches provide paths among different pods.
The fat tree structure provides multiple paths between any two server nodes. This provides fault-tolerant capability with an alternate path in case of some isolated link failure.

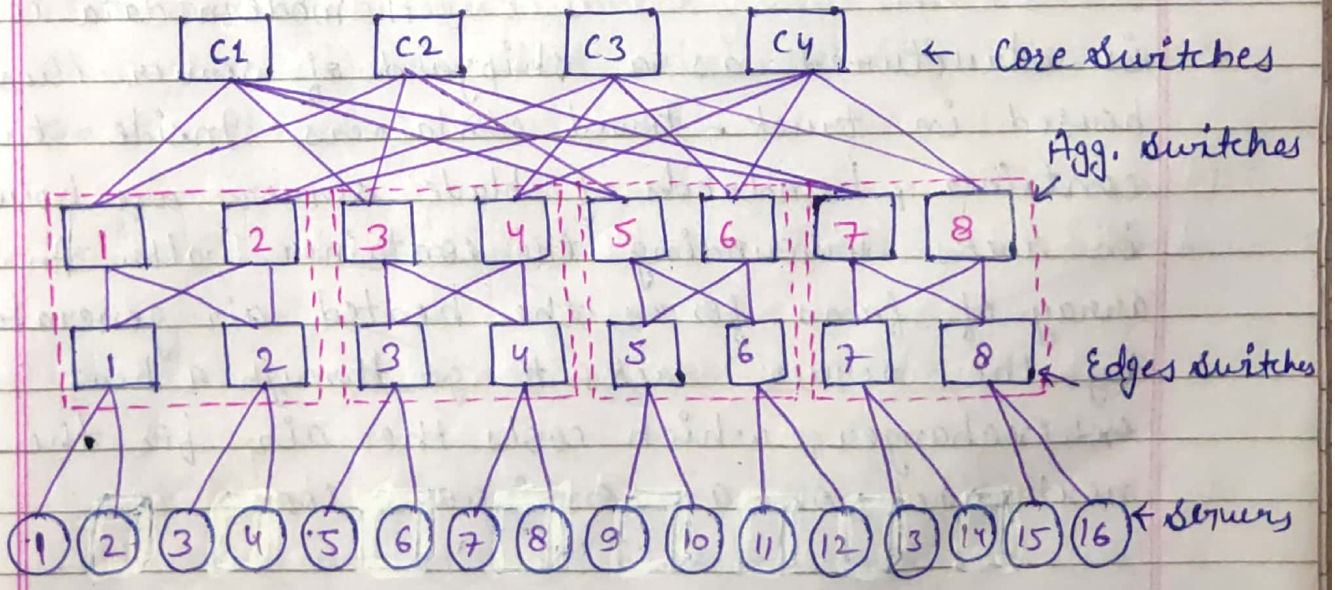


Fig: Fat tree Construction

Teacher's Signature.....

PAGE NO.:
DATE: / /

Steps to design a Fat-tree

Let k be the number of ports that each switch contains.
If the value of k = 4, each switch has 4 ports
Number of core switches is $(k/2)^2 = 4$
Number of pods = k = 4
Each pod consists of $(k/2) = (4/2) = 2$ aggregation switch and $(k/2) = (4/2) = 2$ edge switch.
Each edge switch within pod is connected to
$(k/2) = (4/2) = 2$ core switch and
$(k/2) = (4/2) = 2$ edge switch
Each edge switch within a pod is connected to
$(k/2) = (4/2) = 2$ servers and
$(k/2) = (4/2) = 2$ aggregation switch.
Each pod will be connected to $(k/2)^2 = (4/2)^2 = 4$ servers

★ Modular Data Center design -

A modern data center is structured as a shipyard of server clusters housed in truck-towed containers. Inside the container, hundreds of blade servers are housed in racks surrounding the container walls. An array of fans forces the heated air generated by the server racks to go through a heat exchanger, which cools the air for the next rack on a continuous loop.

large-scale data center built with modular containers appear as a big shipping

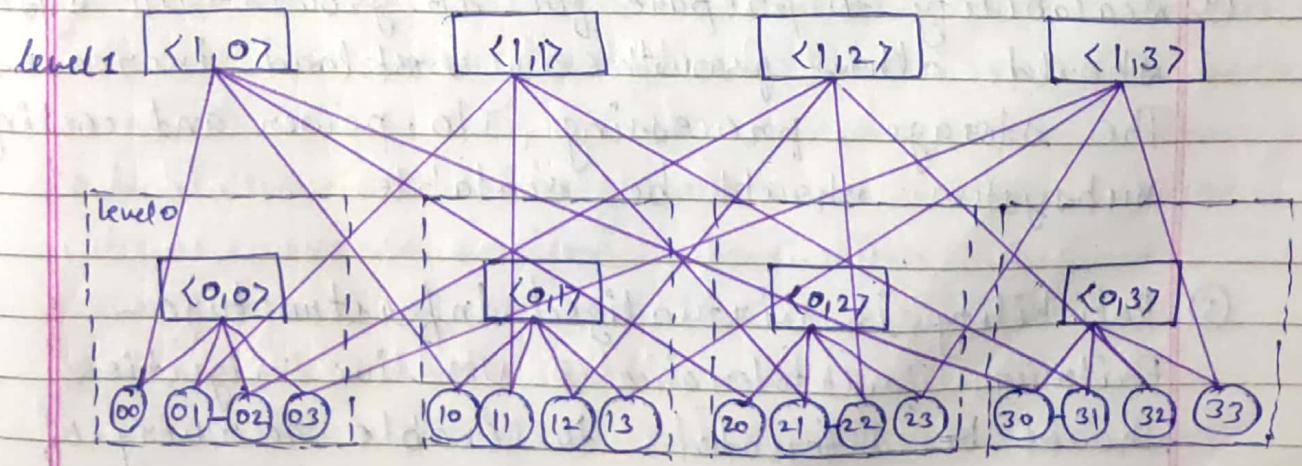
yard of container trucks. This container based data center was motivated by demand for lower power consumption, higher computer density and mobility to relocate data centers to better locations with lower electricity costs, better cooling water supplies and cheaper housing for maintenance engineers. The container must be designed to be weatherproof & easy to transport.

Interconnection of Modular data center -

Modular data center generally use server centric design. The server centric design does modify the O.S running on the servers.

Ex.

BCube is designed specifically for shipping-container based modular data center (MDC). MDC are ad-hoc data centers consisting of thousand of server which are interconnected through switches that are readily deployable on a shipping-container.



BCube - Server Centric Network for MDC

Teacher's Signature

The servers are represented by circles & switches by rectangles. The BCube provides a layered architecture. The bottom layer contains all the server nodes & they form level 0. layer 1 switches form the top layer of BCube$_0$.

Data-Center Management Issues -

(1) Making common user happy - The data center should be designed to provide quality of service to the majority for at least 30 years.

(2) Controlled information flow - Information flow should be streamlined. Sustained services and high availability (HA) are primary goals.

(3) Multiuser manageability - The system must be managed to support all functions of a data-center, including traffic flow, database updating & server maintenance.

(4) Scalability to prepare for db. growth - The system should allow growth as workload increases. The storage, processing, I/O, power and cooling subsystem should be scalable.

(5) Reliability in virtualized infrastructure - Failover, fault tolerance & VM live migration should be integrated to enable recovery of critical applications from failures or disasters.

Teacher's Signature

⑥ Low cost to both users and providers- The cost to users & providers of the cloud system built over the data centers should be reduced, including all operational costs.

⑦ Security and data protection- Data privacy & security defense mechanisms must be deployed to protect the data center against n/w attacks and system interrupts and to maintain data integrity from user abuses or n/w attacks.

⑧ Green information technology- Saving power consumption & upgrading energy efficiency are in high demand when designing & operating current & future data centers.

Architectural Design of Compute & storage Cloud-

Scalability, virtualization, efficiency & reliability are four major design goals of a cloud computing platform. Clouds support web 2.0 applications. Cloud management receives the user request, finds the correct resources and then calls the provisioning services which invokes the resources in the cloud. The cloud management s/w needs to support both physical & virtual machines.

The platform needs to establish a very large scale HPC infrastructure.

Teacher's Signature

The h/w and s/w systems are combined to make it easy and efficient to operate. System scalability can benefit from cluster architecture. If one service takes a lot of processing power, storage capacity or n/w traffic, it is simple to add more servers and bandwidth. System reliability can benifit from this architecture. Data can be put into multiple locations. In such situations even if one of the data centers crashes, the user data is still accessible.

Cloud-Enabling Technologies -

Fast platform deployment
Virtual cluster on demand
Multitenant techniques
massive data processing
Web-scale communication
Distributed storage
Licensing & billing services.

★ Generic Cloud Architecture -

The Internet cloud is envisioned as a massive cluster of servers. These servers are provisioned on demand to perform collective web services or distributed applications using data-centers resources. The cloud platform is formed dynamically by a

Teacher's Signature

provisioning or deprovisioning servers, s/w and database resources. Servers in the cloud can be physical machines or virtual machines (VMs). User interfaces are applied to request services. The provisioning tool carves out the cloud system to deliver the requested service.

The cloud computing resources are built into the data centers, which are typically owned and operated by a third party provider. Consumers do not need to know the underlying technologies. In a cloud, s/w becomes a service.

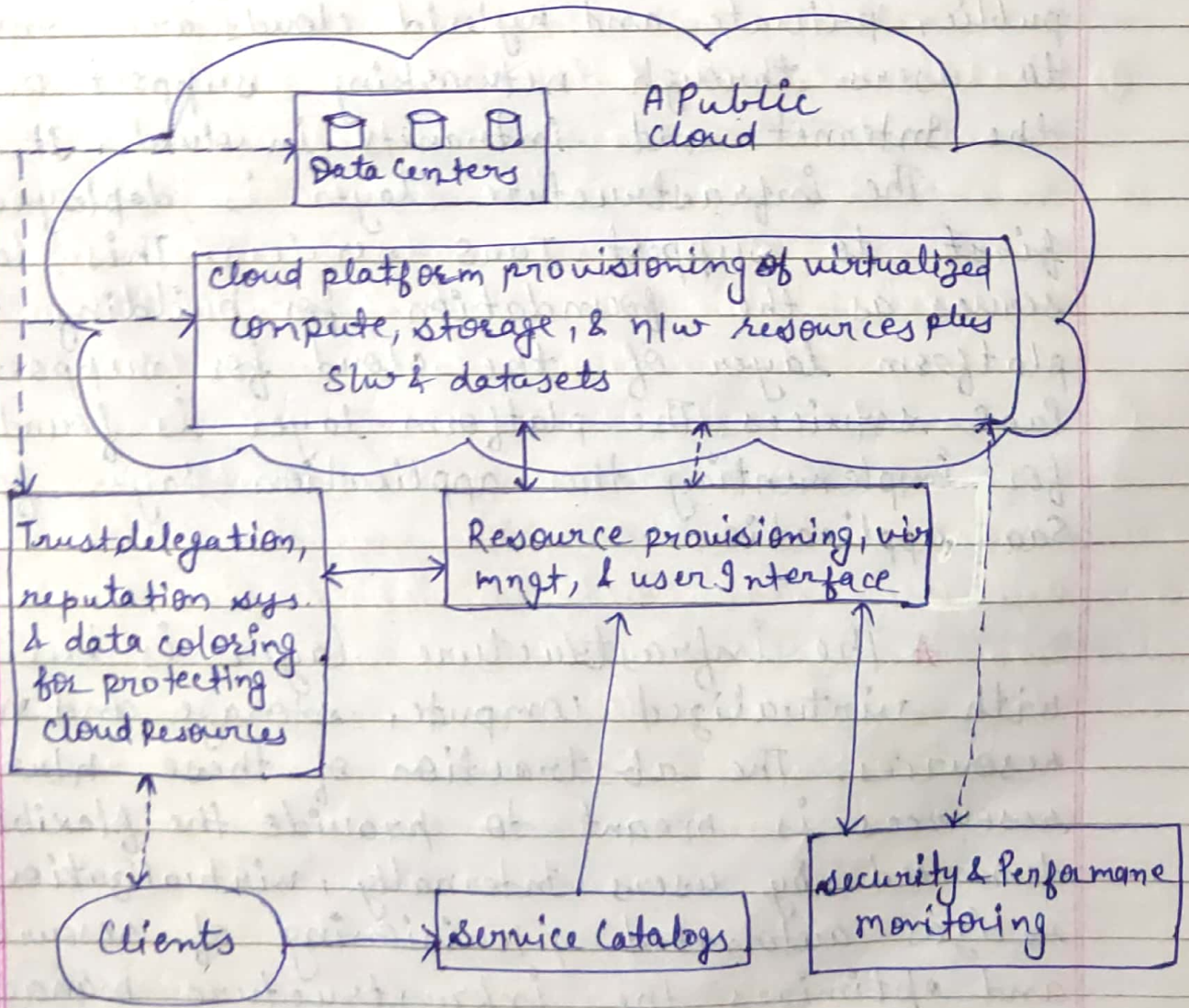


Fig: Security-aware cloud platform

Teacher's Signature

PAGE NO.:
DATE: / /

★ Layered Cloud Architecture development :-

The architecture of a cloud is developed at three layers:

→ Infrastructure
→ Platform
→ Application

These three development layers are implemented with virtualization and standardization of hlw and slw resources provisioned in the cloud. The services to public, private and hybrid clouds are conveyed to users through networking support over the Internet and intranets involved.

The infrastructure layer is deployed first to support Iaas services. This layer serves as the foundation for building the platform layer of the cloud for supporting Paas services. The platform layer is foundation for implementing the application layer for Saas applications.

★ The infrastructure layer is built with virtualized compute, storage and n/w resources. The abstraction of these hlw resources is meant to provide the flexibility demanded by users. Internally, virtualization relizes automated provisioning of resources and optimizes the infrastructure management process.

Teacher's Signature

PAGE NO.:
DATE: / /

★ The platform layer is for general-purpose & repeated usage of the collection of s/w resources. This layer provides users with an environment to develop their app. to test operation flows and to monitor execution results and performance.

★ The application layer is formed with a collection of all needed s/w modules for saas applications. Service applications in this layer include daily office mngt. work, such as information retrieval, document processing and calendar and authentication services.

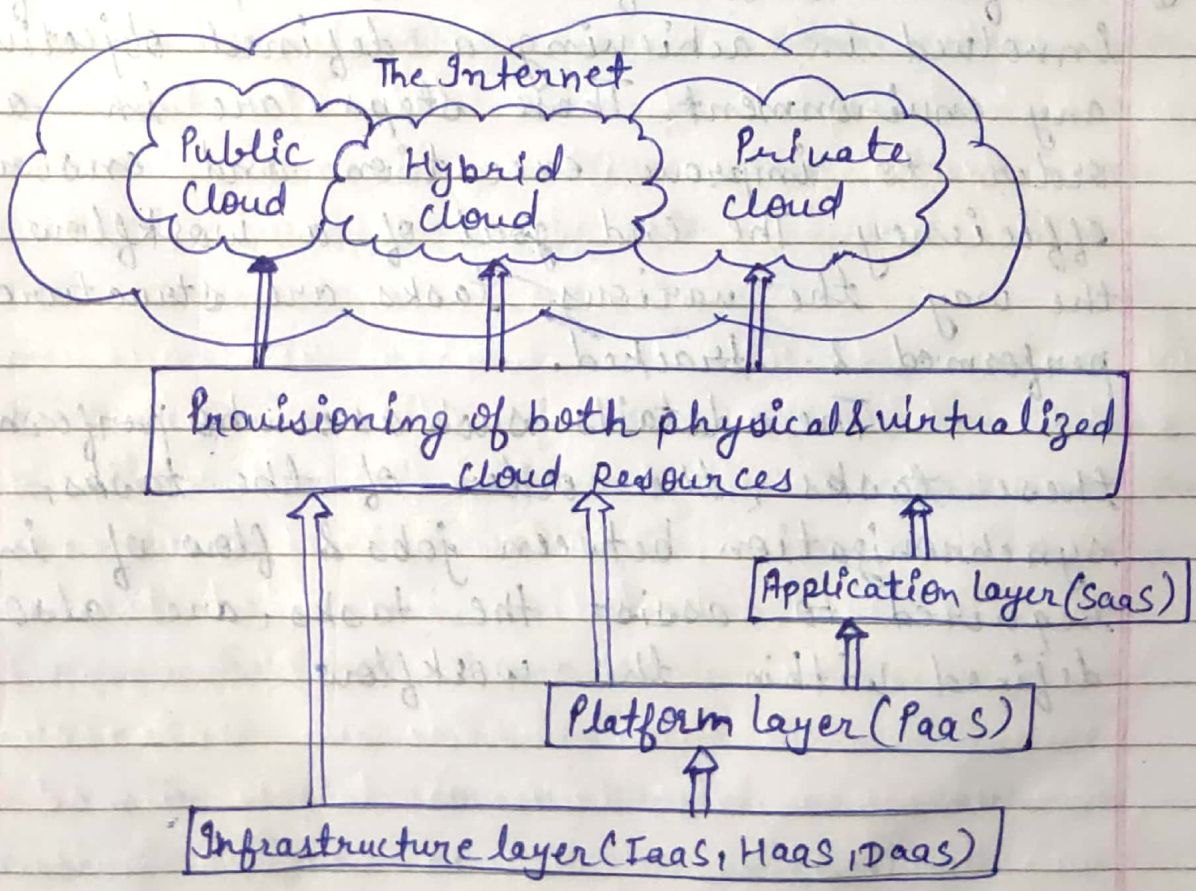


Fig: layered Cloud Architecture development

Teacher's Signature

PAGE NO.:
DATE: / /

Architectural Design Challenges -

→ Service Availability & Data lock-in Problem
→ Data Privacy & security concerns
→ Unpredictable performance & Bottlenecks
→ Distributed storage & widespread s/w Bugs
→ Cloud scalability, Interoperability & Standardization
→ s/w licensing & Reputation sharing

✳ Features of Cloud Programming -

★ Traditional Features common to Grid & Cloud -

① **Workflow** - A workflow is a sequence of steps involved in achieving a defined objective in any environment. These steps are in a certain order to improve execution and ensure efficiency. The end goal of a workflow defines the way the various tasks are structured, performed & tracked.

The details such as: who performs these tasks, the order of the tasks, synchronization between jobs & flow of info. required to assist the tasks, are also defined within the workflow.

Teacher's Signature.....

PAGE NO.:
DATE: / /

Infrastructure Cloud Features -

① **Accounting** - Includes economics; clearly an active area for commercial clouds.

② **Appliances** :- Preconfigured virtual machine (VM) image supporting multifaceted tasks such as message passing interface (MPI) clusters.

③ **Authentication & authorization** - Cloud need single sign-on to multiple systems

④ **Operating System** - Apple, Android, Linux, Windows

⑤ **Program Library** - stores images & other program material.

⑥ **Security** - Security features other than basic authentication & authorization; includes higher level concepts such as trust

⑦ **Virtualization** - This feature allows to share a single physical instance of a resource or an application among multiple customers & organizations

Platform Features Supported by Clouds

① **Blob** - The basic storage concept in clouds is blobs for Azure and S3 for Amazon.

② **DPFS** - Support of file system such as Google (MapReduce), HDFS (Hadoop) and Cosmos (Dryad) with compute-data affinity optimized for data processing.

Teacher's Signature.....

PAGE NO.:
DATE: / /

③ MapReduce- Support MapReduce programming model including Hadoop on Linux, Dryad on windows HPCS & Twister on windows and Linux. Include new associated languages such as Sawzall, Pregel, Pig latin & Linq.

④ Programming model- Cloud programming models are built with other platform features and are related to familiar web & grid models.

⑤ Queues- Queuing system possibly based on publish- subscribe.

⑥ SQL- Relational Database

⑦ Table- Support of table data structure modeled on Apache Hbase or Amazon SimpleDB/Azure Table.

⑧ Worker and Web Roles- Worker roles are basic schedulable processes and are automatically launched. Explicit scheduling is unnecessary in clouds for individual worker roles and for the "gang scheduling" supported transparently in MapReduce.

Web roles provide an interesting approach to portals. GAE is largely aimed at web applications, whereas science gateways are successful in TeraGrid.

PAGE NO.:
DATE: / /

Parallel and Distributed Programming Paradigms -

We define a parallel and distributed program as a parallel program running on a set of computing engines or a distributed computing system

A distributed computing system is a set of computational engines connected by a n/w to achieve a common goal of running a job or an application. A computer cluster or n/w of workstation is an example of a distributed computing system.

Parallel computing is the simultaneous use of more than one computational engine (not necessarily connected via a n/w) to run a job or an application. For instance, parallel computing may use either a distributed or a nondistributed computing system such as multiprocessor platform.

Parallel Computing & Programming Paradigms -

Consider a distributed computing system consisting of a set of networked nodes or workers The system issues for running a typical parallel program in either a parallel or a distributed manner would include the following -

① **Partitioning** - This is applicable to both computation and data as follows -

Teacher's Signature.....

Computation Partitioning – This splits a given job or a program into smaller tasks. Partitioning greatly depends on correctly identifying portions of the job or program that can be performed concurrently. In other words, upon identifying parallelism in the structure of the program, it can be divided into parts to be run on different workers. Different part may process different data or a copy of same data.

Data Partitioning – This splits the input or intermediate data into smaller pieces. Similarly, upon identification of parallelism in the input data, it can also be divided into pieces to be processed on different workers. Data pieces may be processed by different parts of a program or a copy of the same program.

Mapping – This assigns either smaller parts of a program or smaller pieces of data to underlying resources. This process aims to appropriately assign such parts or pieces to be run simultaneously on different workers and is usually handled by resource allocators in the system.

Synchronization – Because different workers may perform different tasks, synchronization

and coordination among workers is necessary so that race conditions are prevented & data dependency among different workers is properly managed. Multiple accesses to a shared resource by different workers may raise race conditions, whereas data dependency happens when a worker needs the processed data of other workers.

Communication – Because data dependency is one of the main reasons for communication among workers, communication is always triggered when the intermediate data is sent to workers.

Scheduling – For a job or program, when the number of computation parts (tasks) or data pieces is more than the no. of available workers, a scheduler selects a sequence of tasks or data pieces to be assigned to the workers.

Scheduling is necessary when system resources are not sufficient to simultaneously run multiple jobs or programs.

PAGE NO.:
DATE: / /

MapReduce

MapReduce is a s/w framework which supports parallel and distributed computing on large data sets. This s/w framework abstracts the data flow of running a parallel program on a distributed computing system by providing users with two interfaces in the form of two functions:
- Map and Reduce

Users can override these two functions to interact with and manipulate the data flow of running their programs.

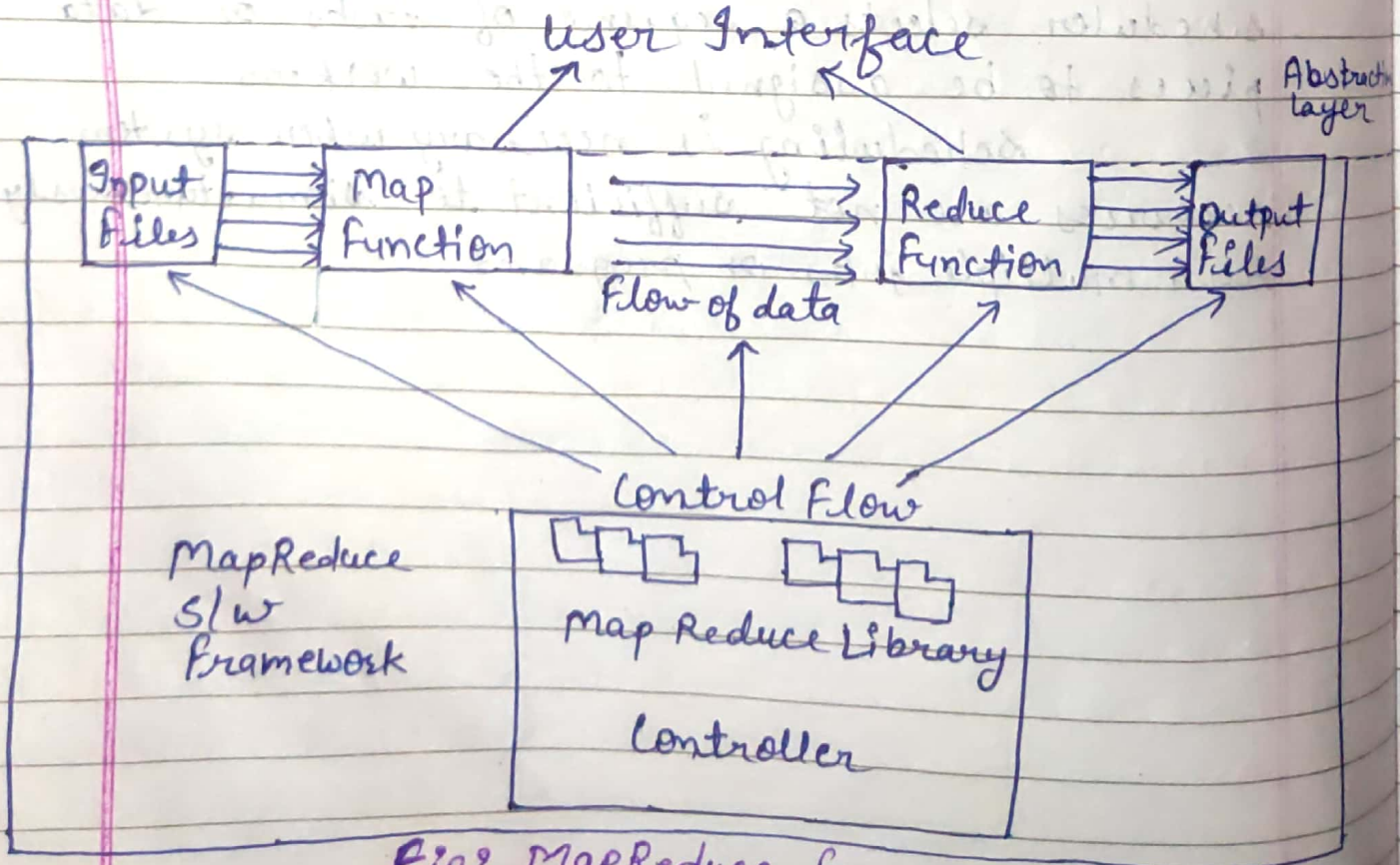


Fig: MapReduce Framework

In this framework, the "value" part of the data, (key, value) is the actual data & the "key" part is only used by the MapReduce controller to control the data flow.

The overall structure of user's program containing the Map, Reduce and the main functions is given below. The Map & Reduce are two major subroutines. They will be called to implement the desired function performed in the main program.

```
Map function (.....)
{
 - - - - - -
}
Reduce function (.....)
{
 - - - - - -
}
Main function (.....)
{
   Initialize Spec object
   ---------
   MapReduce (Spec, & Results)
}
```

PAGE NO.:
DATE: / /

MapReduce logical Data Flow - The input data to the Map function is in the form of a (key, value) pair. For ex., the key is the line offset within the input file and value is the content of the line. The output data from the Map function is structured as (key, value) pairs called intermediate (key, value) pairs.

The user-defined Map function processes each input (key, value) pair and produces a no. of (zero, one or more) intermediate (key, value) pairs.

The Reduce function receives the intermediate (key, value) pairs in the form of a group of intermediate values associated with one intermediate key, (key, [set of values]).

The MapReduce framework forms these groups by first sorting the intermediate (key, value) pairs and then grouping values with the same key.

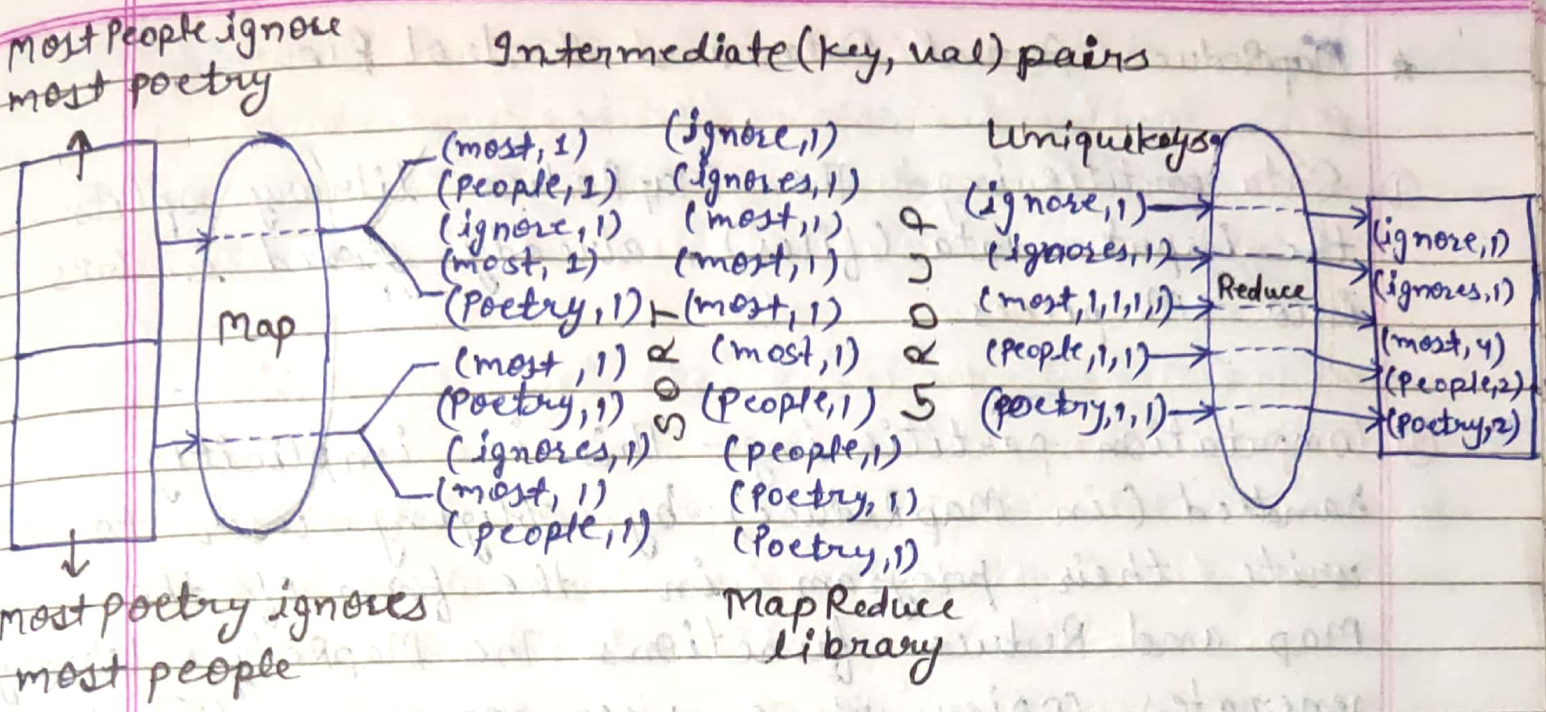


Fig: Data flow of a word-count problem using MapReduce functions (Map, Sort, Group & Reduce)

Formal Notation of MapReduce Data Flow - The Map function is applied in parallel to every input (key, value) pair & produces new set -

$$(key_1, val_1) \xrightarrow{\text{Map function}} \text{List}(key_2, val_2)$$

MapReduce library collects all the produced pair from all input and sorts them based on the "key" part. It then groups the value of all occurrences of same key. Finally, Reduce fun is applied to produce output -

$$(key_2, \text{list}(val_2)) \xrightarrow{\text{Reduce fun}} \text{list}(val_2)$$

PAGE NO.:
DATE: / /

★ MapReduce Actual Data and control Flow -

① Data partitioning - The MapReduce library splits the input data (files), already stored in GFS into M pieces.

② Computation partitioning - This is implicitly handled (in MapReduce) by obliging user to write their program in the form of the Map and Reduce functions. The MapReduce library generates copies of a user program, distributes them up on a no of available computation engine.

③ Determining the Master & workers - The MapReduce architecture is based on a master-worker model. Therefore, one of the copies of the user prg becomes the master & the rest become workers. The master picks idle workers, & assigns the map & reduce tasks to them.

④ Reading the input data (data distribution) - Each map worker reads its corresponding portion of the input data & sends it to its map function.
(Input data split)

⑤ Map function - Each map function receives the input data split as a set of (key, value) pairs to process & produce the intermediate (key, value) pairs.

Teacher's Signature

⑥ Combiner Function - This fun. merges the local data of each map worker before sending it over the n/w to effectively reduce its communication costs.

⑦ Partitioning Function - The intermediate (key, value) pairs produced by each map worker are partitioned into R regions, equal to the number of reduce tasks, by the Partitioning function to guarantee that all (key, value) pairs with identical keys are stored in the same region. Partitioning function forwards the data into particular regions.

⑧ Synchronization - MapReduce applies a simple synchronization policy to coordinate map workers with reduce workers, in which the communication between them starts when all map tasks finish

⑨ Communication - Reduce worker i, already notified of the location of region i of all map workers, uses a remote procedure call to read the data from the respective region of all map workers. Since all reduce workers read the data from all map workers, all-to-all communication among all map and reduce workers, which incurs n/w congestion occurs in the n/w.

⑩ Sorting and Grouping - When the process of reading input data is finalized by a reduce

Teacher's Signature.....

worker, the data is initially buffered in the local disk of the reduce worker. Then the reduce worker groups intermediate (key, value) pairs by sorting the data based on their keys, followed by grouping all occurrences of identical keys.

(11) Reduce Function - The reduce worker iterates over the grouped (key, value) pairs and for each unique key, it sends the key & corresponding values to the Reduce function. Then this function processes its input data & stores the output results in predetermine files in the user's program.

PAGE NO.:
DATE: / /

Hadoop -

Hadoop is an open source framework from Apache & is used to store, process and analyze data which are very huge in volume.
Hadoop is written in Java & is not OLAP (Online Analytical Processing).
It is used for batch/offline processing
It is being used by Facebook, Yahoo, Google, Twitter LinkedIn and many more.

Hadoop Architecture - Hadoop has two major layers -
→ MapReduce - Processing/Computation layer
→ Hadoop Distributed File System - Storage layer.

MapReduce - MapReduce is a parallel programming model for writing distributed application devised at Google for efficient processing of large amounts of data (multi-terabyte data-sets), on large clusters (thousands of nodes) of commodity h/w in a reliable, fault-tolerant manner. The MapReduce program runs on Hadoop which is an Apache open source framework.

HDFS - HDFS is based on the Google File system (GFS) & provides a distributed file sytem that is designed to run on commodity h/w.
HDFS holds very large amount of data & provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant

Teacher's Signature.....

fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

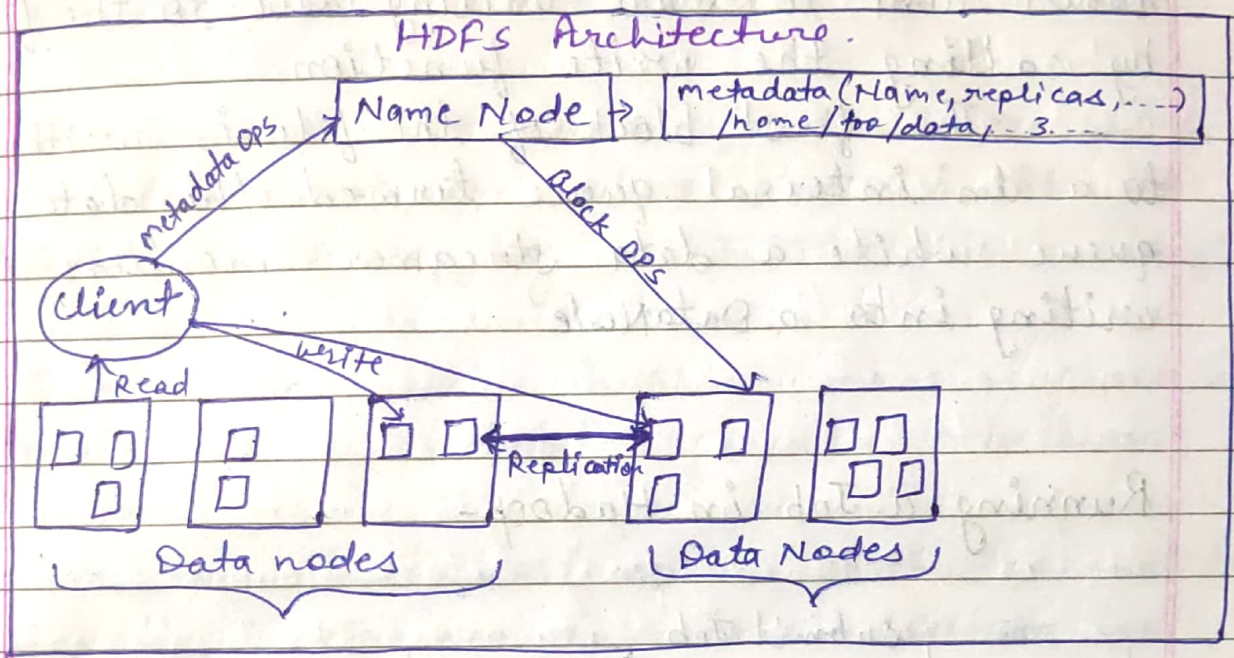
HDFS Architecture- HDFS follows master-slave architecture & it has following elements

① Blocks- A Block is the minimum amount of data that it can read or write. HDFS blocks are 128 MB by default & this is configurable The user data is stored in the files of HDFS. Files in HDFS are broken into block-sized chunks, which are stored as independent units. If the file is smaller than block size, then it doesnot occupy full block.

② Name Node- HDFS works in master-worker pattern where the name node acts as master. Name-Node is controller & manager of HDFS as it knows the status & the metadata of all files in HDFS; the metadata information being file permission, names & location of each block. The metadata are small, so it is stored in the memory of name node, allowing faster access to data. HDFS cluster is accessed by multiple clients concurrently, so all this information is handled by a single machine.

PAGE NO.:
DATE: / /

③ Data Node- Datanodes perform read-write operations on the file systems, as per client request. They also perform operations such as block creation, deletion & replication according to the instructions of namenode



HDFS Operation-

Reading a file- To read a file in HDFS, a user sends an "Open" request to the NameNode to get the location of file blocks. For each file block, the NameNode returns the address of a set of DataNodes containing replica information for the requested file.

Upon receiving such information, the user calls the read function to connect to the closest DataNode containing the first block of the file.

Teacher's Signature.....

Writing to a file- To write a file in HDFS a user sends a "create" request to the NameNode to create a new file in the file system namespace. If the file doesn't exist, the NameNode notifies the user and allows him to start writing data to the file by calling the write function.

The first block of the file is written to a an internal queue termed the data queue while a data streamer monitors its writing into a DataNode.

Running a Job in Hadoop-

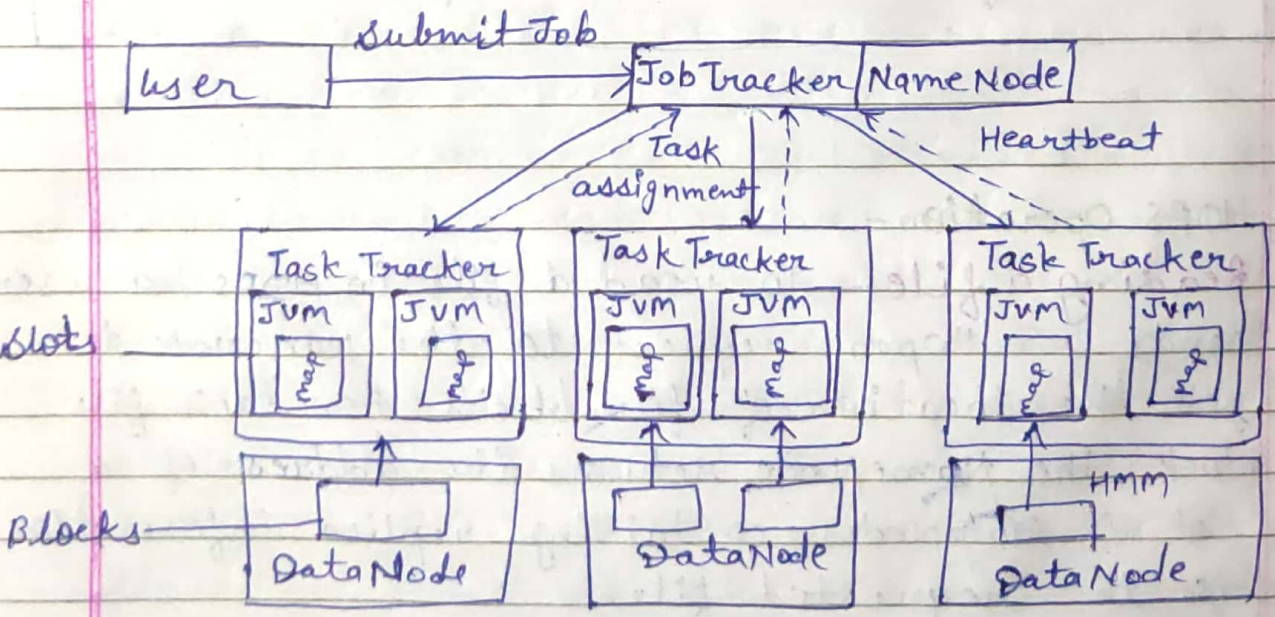


Fig: Data flow in running a MapReduce Job at various task tracker using Hadoop.

Teacher's Signature

PAGE NO.:
DATE: / /

Job Submission - Each job is submitted from a user node to the Job Tracker node that might be situated in different node within the cluster

Task assignment - The JobTracker assigns the map task to the Task Trackers.

Task execution - The control flow to execute a task (either map or reduce) starts inside the TaskTracker by copying the JobJAR file to its file sys.

Task running Check - A task running check is performed by receiving periodic heartbeat messages to the JobTracker from the TaskTracker. Each heartbeat notifies the JobTracker that the sending TaskTracker is alive & whether the sending TaskTracker is ready to run a new task.

PAGE NO.:
DATE: / /

High level language for Cloud

① SQL - Structured Query Language

SQL is a database computer language, designed for storing, manipulating and retrieving data in a relational database. Most application programs for computers use SQL to connect server and the database. The cloud is heavily dominated by SQL making it the most popular programming language for cloud computing.

SQL is categorized as a distinctive-purpose programming language because it is designed for data management in a relational database. SQL has simple syntax & is easy to learn.

② Python - Python is a high-level programming language that was designed to simplify the whole big data concept. It is quite readable & anyone can learn it within a short time.

Python is an ideal language for cloud computing because it allows programmers to create, analyze & organize large chunks of data with ease. It is also an ideal option for streaming analytics applications that are based on the cloud.

Teacher's Signature

PAGE NO.:
DATE: / /

③ Clojure Programming language -

"Clojure" is a general-purpose programming language that utilizes the strengths of scripting language & rides on the efficient resource-rich multithreaded programming. Despite being a compiled language, Clojure still manages to remain dynamic hence can be supported at runtime environment. The programming language gives programmers easy access to different Java frameworks. Clojure supports immutable data structure which are mainly used in cloud computing.

④ Go-Programming Language -

Go (Golang) programming language was invented at Google & was intended to be an alternative to C++. The language can be used to develop standalone PC S/w applications.

Go has an excellent portfolio in the cloud computing arena. There is quite a number of tools that are developed using Golang including Docker & Consul. These two tools are heavily used for big data and cloud computing. Go is known for scalability & memory efficiency.

⑤ XML with Java Programming -

XML (Extensible Markup language) is basically used to describe data. This is not a programming language that

Teacher's Signature.....

you can use to develop a complex application from the scratch. Instead, the language encodes data & documents into a format that can be understood by both the machine and human readers. xml tags identify the data & are used to store and organize the data.

⑥ Erlang- Erlang is an agile programming language used for building highly scalable solutions. It is also used to build systems that depend on the real-time transmission of data. This makes it an ideal language for cloud computing. Some of the major solutions that are built on Erlang include telecommunications system and real-time messaging platform.

Erlang support distributed system. It also has a "hot swapping" feature whereby a code can be changed without interfering with the system's functionality.

⑦ Haskell- Haskell is a general-purpose & functional programming language that is ideal for cloud computing projects. It is a statically typed language that is run during the compile time.

Unlike most programming languages, Haskell uses semantics instead of syntax. It does not have statements or give instructions. It only uses expressions.

Teacher's Signature

PAGE NO.:
DATE: / /

Programming of Google App Engine (GAE)-

→ GAE is an example of Platform-as-a-service (PaaS)
→ GAE provides web app. developers & enterprise with access to Google's scalable hosting & tier 1 Internet service.
→ GAE supports applications which are written in Java or Python.
→ Applications in Google App Engine stores data in Google Big Table.
→ Application in GAE uses Google query language. If applications are non-compatible to GAE, then application needed to be make compatible with GAE. All applications are not supported by GAE.
→ GAE also removed some system administration & developmental task to make it easier to write scalable application.

→ App Engine is a platform to create, store & run applications on Google's servers using development languages as Java & python.
→ GAE includes tools for managing the data store, monitoring the site & its resource consumption, and debugging & logging.
→ A user can share his application with the world, or limit access to members of organization.
→ All applications can use upto 1 GB of storage and enough CPU & bandwidth to support an efficient app serving around 5 million page views a month, absolutely free.

Teacher's Signature.....

GAE enables users to build a basic web application very quickly.
The GAE architecture provides a new approach without dealing with web servers & load balancers but instead deploying the applications on the GAE cloud by providing instance access & scalability which is shown in fig. The architecture defines the structure of application that run on the GAE.

Python- The GAE allows implementation of app. using Python prg. language & running them on its interpreter. The GAE provides rich API & tools for designing web applications, data modeling, managing apps data, support for mature libraries & framework like Django.

Bigtable- The Datastore is built on top of Bigtable which is built on top of GFS(Google File System). The data store is the only database GAE supports for logging & storing data, including session data.

The Bigtable is a tabular NoSQL database that is designed to reliably scale to petabyte of data & thousand of machines. It is a sparse, distributed, persistent, multi dimensional storage map. It is generally referred to as a "map" indexed with row key, column key & a time stamp.

PAGE NO.:
DATE: / /

The Bigtable has a master server that coordinates the large segment of a logical table called "tablets". The tablets are split across a row with an optimal size of 200MB per each tablet for organization purposes. The table contains rows & columns & each cell has time stamp. The GAE allows usage of the Bigtable in applications through the Datastore API.

Datastore API - The Datastore is responsible for the scalability of the GAE applications. The structure of the applications enables them to distribute the requests across the server which shoud be compromised with relational databases.

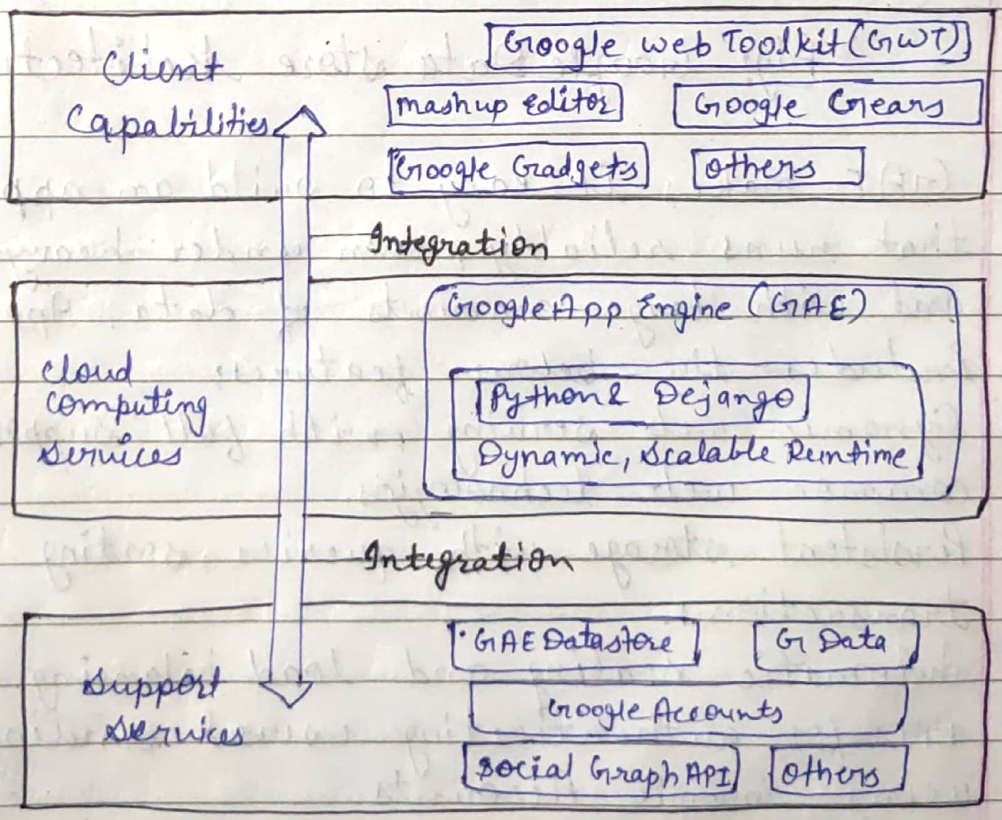


Fig: Architecture of Google App Engine

Teacher's Signature

DATE: / /

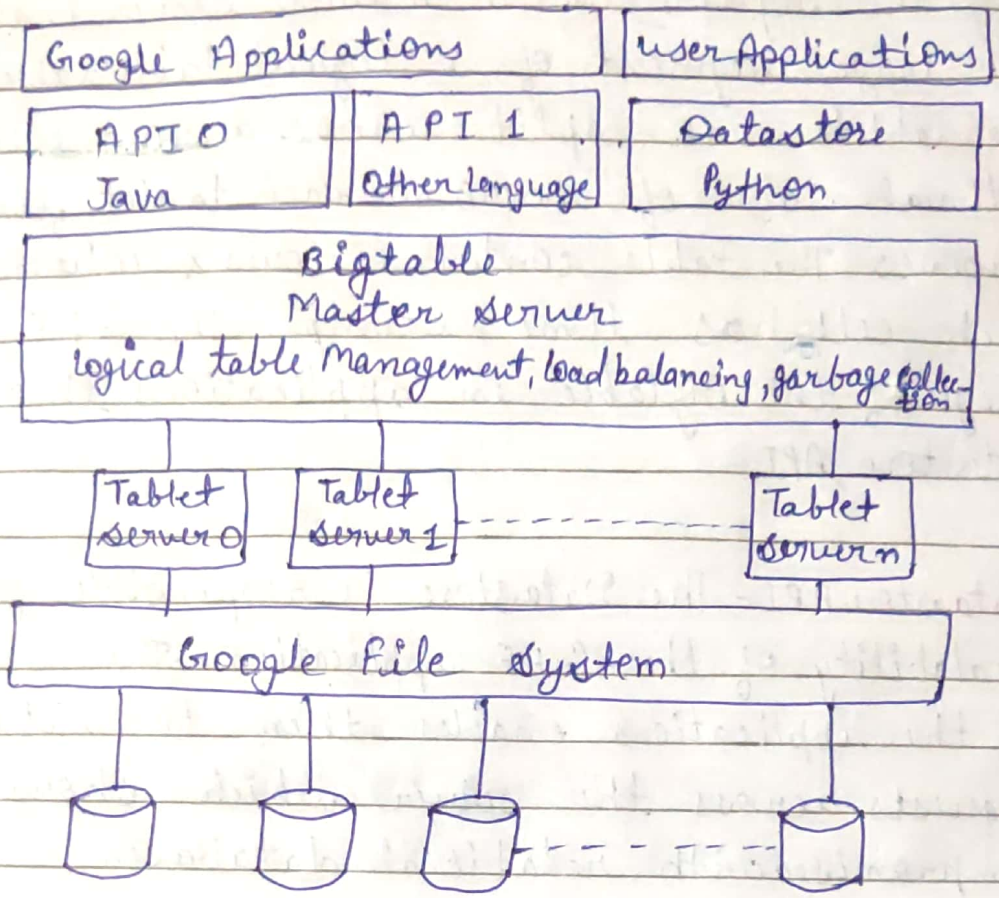


Fig: Google Data store Architecture

GAE makes it easy to build an application that runs reliably, even under heavy load and with large amounts of data. App Engine includes the below features:

→ Dynamic web serving, with full support for common web technologies.
→ Persistent storage with queries, sorting and transactions.
→ Automatic scaling and load balancing.
→ APIs for authenticating users & sending email using Google Accounts.
→ Scheduled tasks for triggering events at specified times and regular intervals.