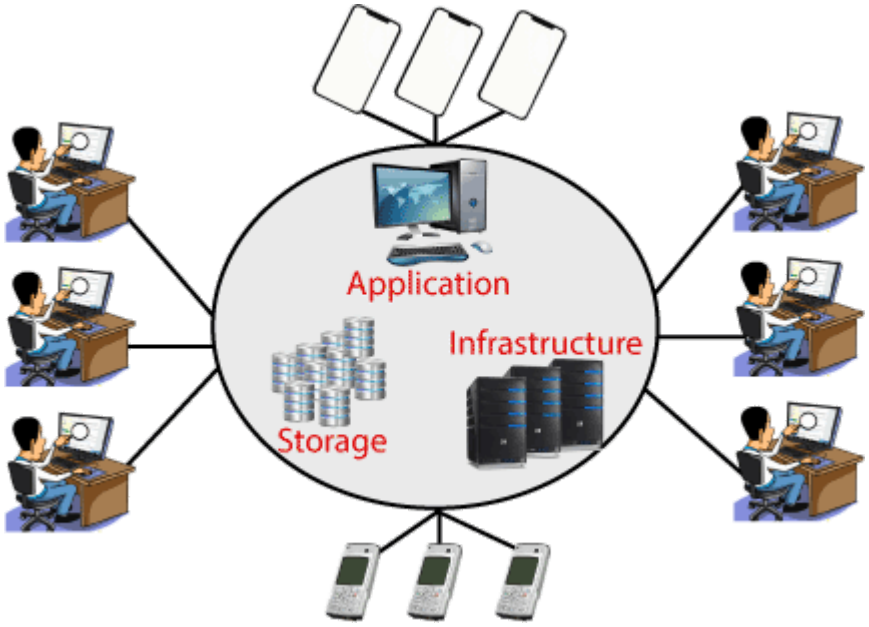
CHAPTER-1
INTRODUCTION TO CLOUD COMPUTING

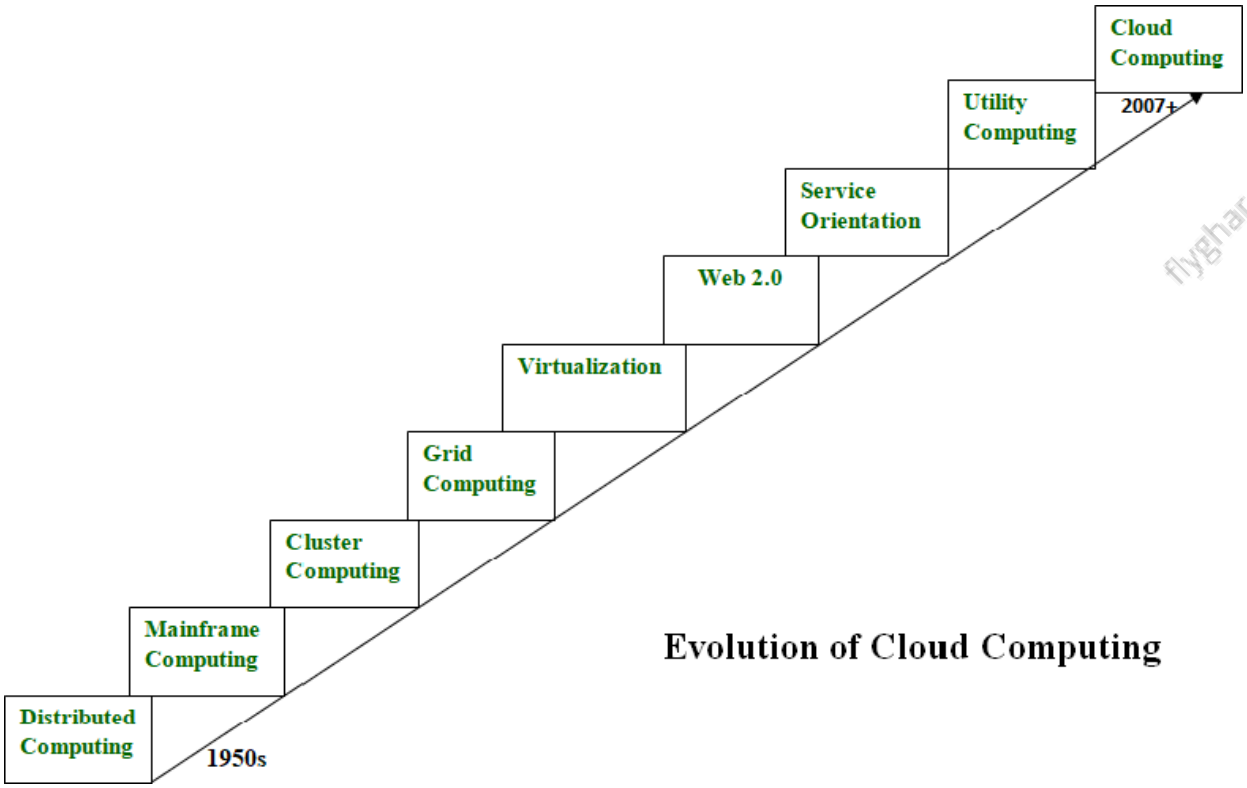
Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud (Internet).



Cloud Computing provides an alternative to the on-premises datacentre. With an on-premises datacentre, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data. After doing all the set-up, we become responsible for maintaining it through its entire lifecycle.

1.1 Evolution of Cloud Computing

Cloud computing is all about renting computing services. This idea first came in the 1950s. In making cloud computing what it is today, five technologies played a vital role. These are distributed systems and its peripherals, virtualization, web 2.0, service orientation, and utility computing.



- **Distributed Systems:**
  It is a composition of multiple independent systems but all of them are depicted as a single entity to the users. The purpose of distributed systems is to share resources and also use them effectively and efficiently. Distributed systems possess characteristics such as scalability, concurrency, continuous availability, heterogeneity, and independence in failures. But the main problem with this system was that all the systems were required to be present at the same geographical location. Thus to solve this problem, distributed computing led to three more types of computing and they were-Mainframe computing, cluster computing, and grid computing.
- **Mainframe computing:**
  Mainframes which first came into existence in 1951 are highly powerful and reliable computing machines. These are responsible for handling large data such as massive input-output operations. Even today these are used for bulk processing tasks such as online transactions etc. These systems have almost no downtime with high fault tolerance. After distributed computing, these increased the processing capabilities of the system. But these were very expensive. To reduce this cost, cluster computing came as an alternative to mainframe technology.
- **Cluster computing:**
  In 1980s, cluster computing came as an alternative to mainframe computing. Each machine in the cluster was connected to each other by a network with high bandwidth. These were way cheaper than those mainframe systems. These were equally capable of high computations. Also, new nodes could easily be added to the cluster if it was required. Thus, the problem of the cost was solved to some extent but the problem related

to geographical restrictions still pertained. To solve this, the concept of grid computing was introduced.

- **Grid computing:**
  In 1990s, the concept of grid computing was introduced. It means that different systems were placed at entirely different geographical locations and these all were connected via the internet. These systems belonged to different organizations and thus the grid consisted of heterogeneous nodes. Although it solved some problems but new problems emerged as the distance between the nodes increased. The main problem which was encountered was the low availability of high bandwidth connectivity and with it other network associated issues. Thus. cloud computing is often referred to as "Successor of grid computing".
- **Virtualization:**
  It was introduced nearly 40 years back. It refers to the process of creating a virtual layer over the hardware which allows the user to run multiple instances simultaneously on the hardware. It is a key technology used in cloud computing. It is the base on which major cloud computing services such as Amazon EC2, VMware vCloud, etc work on. Hardware virtualization is still one of the most common types of virtualization.
- **Web 2.0:**
  It is the interface through which the cloud computing services interact with the clients. It is because of Web 2.0 that we have interactive and dynamic web pages. It also increases flexibility among web pages. Popular examples of web 2.0 include Google Maps, Facebook, Twitter, etc. Needless to say, social media is possible because of this technology only. In gained major popularity in 2004.
- **Service orientation:**
  It acts as a reference model for cloud computing. It supports low-cost, flexible, and evolvable applications. Two important concepts were introduced in this computing model. These were Quality of Service (QoS) which also includes the SLA (Service Level Agreement) and Software as a Service (SaaS).
- **Utility computing:**
  It is a computing model that defines service provisioning techniques for services such as compute services along with other major services such as storage, infrastructure, etc which are provisioned on a pay-per-use basis.

Thus, the above technologies contributed to the making of cloud computing.

1.2 Characteristics of Cloud Computing:

The top 15 major characteristics of Cloud Computing:

1. Resources Pooling

Resource pooling is one of the essential characteristics of Cloud Computing. Resource pooling means that a cloud service provider can share resources among several clients, providing everyone with a different set of services as per their requirements. It is a multi-client strategy that can be applied to data storage services, processing services, and bandwidth provided services. The administration process of allocating resources in real-time doesn't conflict with the client's experience.

2. On-Demand Self-Service

It is one of the significant and essential features of Cloud Computing. It enables the client to constantly monitor the server uptime, abilities, and allotted network storage. This is a

fundamental characteristic of Cloud Computing, and a client can likewise control the computing abilities as per his needs.

3. Easy Maintenance

This is one of the best cloud characteristics. The servers are effortlessly maintained, and the downtime remains low or absolutely zero sometimes. Cloud Computing powered resources undergo several updates frequently to optimize their capabilities and potential. The updates are more viable with the devices and perform quicker than the previous versions.

4. Scalability And Rapid Elasticity

A key characteristic and benefit of cloud computing is its rapid scalability. This cloud characteristic enables cost-effective running of workloads that require a vast number of servers but only for a short period. Many clients have such workloads, which can be run very cost-effectively because of the rapid scalability of Cloud Computing.

5. Economical

This cloud characteristic helps in reducing the IT expenditure of the organizations. In Cloud Computing, the client needs to pay the administration for the space they have used. There is no covered up or additional charge which needs to be paid. The administration is economical, and more often than not, some space is allotted for free.

6. Measured And Reporting Service

Reporting services are one of the many cloud characteristics that make it the best choice for organizations. Measuring & reporting service is helpful for both cloud providers and their clients. It enables both the provider and the client to monitor and report what services have been used and for what purpose. This helps in monitoring billing and ensuring the optimum usage of resources.

7. Security

Data security is one of the best characteristics of Cloud Computing. Cloud services create a copy of the data that is stored to prevent any form of data loss. If one server loses the data by any chance, the copy version is restored from the other server. This feature comes handy when several users work on a particular file in real-time and a file suddenly gets corrupted.

8. Automation

Automation is an essential characteristic of cloud computing. The ability of cloud computing to automatically install, configure, and maintain a cloud service is known as automation in cloud computing. In simple terms, it is the process of making the most of technology and reducing manual effort. However, to achieve automation in the cloud ecosystem is not so easy. It requires the installation and deployment of virtual machines, servers, and large storage. Upon successful deployment, these resources require constant maintenance as well.

9. Resilience

Resilience in cloud computing means the ability of the service to quickly recover from any disruption. A cloud's resilience is measured by how fast its servers, databases, and network system restarts and recovers from any kind of harm or damage. Availability is another major characteristic of cloud computing. Since cloud services can be accessed remotely, there is no geographic restriction or limitation when it comes to utilizing cloud resources.

10. Large Network Access

A big part of the cloud characteristics is its ubiquity. The client can access the cloud data or transfer the data to the cloud from any place just with a device and internet connection. These capacities are accessible everywhere in the organization and get to with the help of the internet. Cloud providers save that large network access by monitoring and guaranteeing different

measurements that reflect how clients access cloud resources and data: latency, access time, data throughput, etc.

11. Work From Any Location

Remote working is one of the primary **features of cloud computing. It** enables users to work remotely or from any location on the planet. Users will be able to access corporate data via mobile devices such as laptops and smartphones. It also ensures that every user can connect quickly. Employees that work remotely or live in another location can communicate with one another and do their jobs.

12. Multi-Tenancy

One of the best **characteristics of Cloud Computing** is its Multi-Tenancy. Multi-Tenancy is a software architecture that allows a single program instance to serve several user groups. It signifies that numerous cloud provider customers are sharing the same computing resources. Although they share the same computing resources, each Cloud customer's data is kept completely separate and safe.

13. Flexibility

The company must scale as the competition develops. Users that use the traditional hosting technique will have to switch service providers. Users benefit from greater freedom when they host their data in the cloud. Scaling does not require the server to be restarted and can be done anytime. There are a variety of payment alternatives also available for the users. This implies businesses won't have to spend too much on resources they don't need.

14. Service Excellence

Cloud computing ensures that users receive the highest level of service possible. The benefits outlined in Service Level Agreements must include continuous availability and comprehensive resources, performance, and bandwidth. Any compromise on these services will result in a loss of clients and a decrease in popularity.

15. Comfortable Payment Structure

The payment structure is critical since it aids in the cost-cutting process. Because of the additional functionality, cloud computing choices come with a range of prices. Users find the payment option to be simple to use, and it allows them to save time when making payments regularly.

(Just need to learn 6/7 characters for exam purpose)

1.3 Types of cloud and its Cloud services

Types of cloud deployment models:

The cloud deployment model identifies the specific type of cloud environment based on ownership, scale, and access, as well as the cloud's nature and purpose. The location of the servers you're utilizing and who controls them are defined by a cloud deployment model. It specifies how your cloud infrastructure will look, what you can change, and whether you will be given services or will have to create everything yourself. Relationships between the infrastructure and your users are also defined by cloud deployment types.

Different types of cloud computing deployment models are:

1. Public cloud
2. Private cloud
3. Hybrid cloud
4. Community cloud

1. Public Cloud

The public cloud makes it possible for anybody to access systems and services. The public cloud may be less secure as it is open for everyone. The public cloud is one in which cloud infrastructure services are provided over the internet to the general people or major industry groups. The infrastructure in this cloud model is owned by the entity that delivers the cloud services, not by the consumer. It is a type of cloud hosting that allows customers and users to easily access systems and services. This form of cloud computing is an excellent example of cloud hosting, in which service providers supply services to a variety of customers. In this arrangement, storage backup and retrieval services are given for free, as a subscription, or on a per-use basis. Example: Google App Engine etc.

Advantages of the public cloud model:

- **Minimal Investment:** Because it is a pay-per-use service, there is no substantial upfront fee, making it excellent for enterprises that require immediate access to resources.
- **No setup cost:** The entire infrastructure is fully subsidized by the cloud service providers, thus there is no need to set up any hardware.
- **Infrastructure Management is not required:** Using the public cloud does not necessitate infrastructure management.
- **No maintenance:** The maintenance work is done by the service provider (Not users).
- **Dynamic Scalability:** To fulfill your company's needs, on-demand resources are accessible.

2. Private Cloud

The private cloud deployment model is the exact opposite of the public cloud deployment model. It's a one-on-one environment for a single user (customer). There is no need to share your hardware with anyone else. The distinction between private and public cloud is in how you handle all of the hardware. It is also called the "internal cloud" & it refers to the ability to access systems and services within a given border or organization. The cloud platform is implemented in a cloud-based secure environment that is protected by powerful firewalls and under the supervision of an organization's IT department.

The private cloud gives the greater flexibility of control over cloud resources.

Advantages of the private cloud model:

- **Better Control:** You are the sole owner of the property. You gain complete command

over service integration, IT operations, policies, and user behavior.

- **Data Security and Privacy:** It's suitable for storing corporate information to which only authorized staff have access. By segmenting resources within the same infrastructure, improved access and security can be achieved.
- **Supports Legacy Systems:** This approach is designed to work with legacy systems that are unable to access the public cloud.
- **Customization:** Unlike a public cloud deployment, a private cloud allows a company to tailor its solution to meet its specific needs.

3. Hybrid cloud

By bridging the public and private worlds with a layer of proprietary software, hybrid cloud computing gives the best of both worlds. With a hybrid solution, you may host the app in a safe environment while taking advantage of the public cloud's cost savings. Organizations can move data and applications between different clouds using a combination of two or more cloud deployment methods, depending on their needs.

Advantages of the hybrid cloud model:

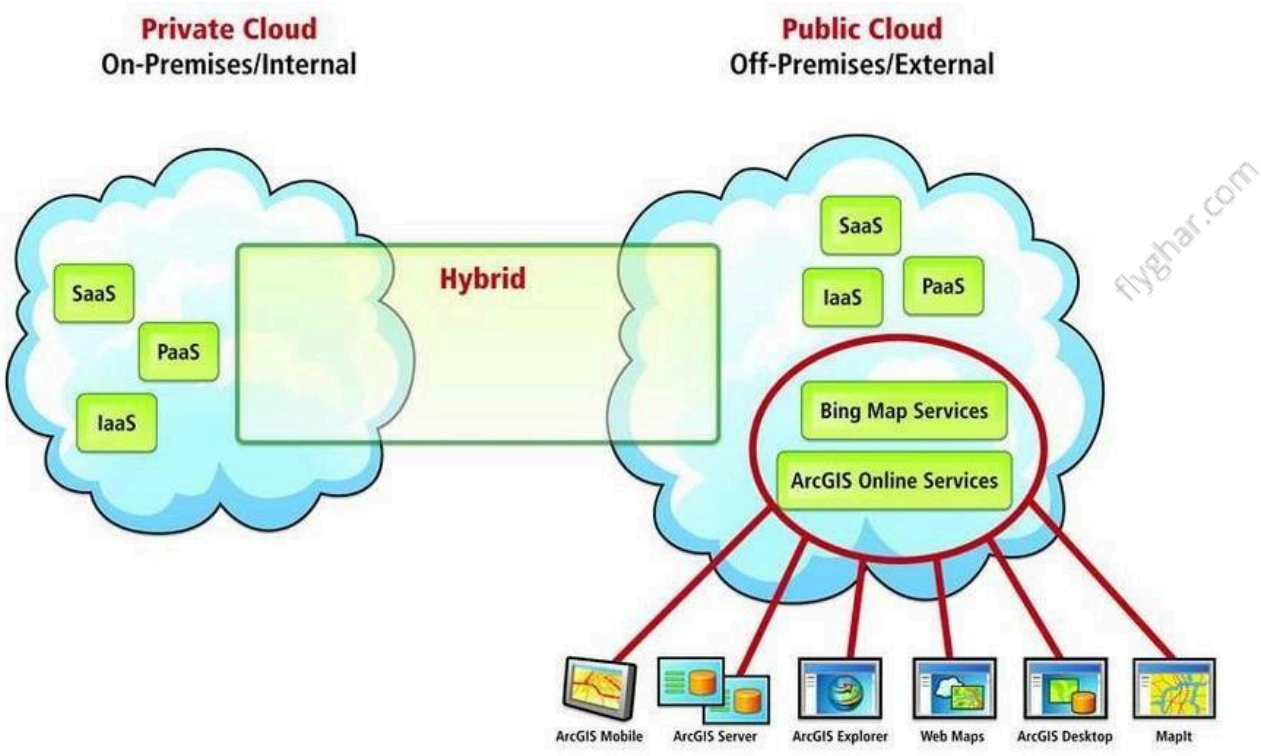
- **Flexibility and control:** Businesses with more flexibility can design personalized solutions that meet their particular needs.
- **Cost:** Because public clouds provide for scalability, you'll only be responsible for paying for the extra capacity if you require it.
- **Security:** Because data is properly separated, the chances of data theft by attackers are considerably reduced.

4. Community cloud

It allows systems and services to be accessible by a group of organizations. It is a distributed system that is created by integrating the services of different clouds to address the specific needs of a community, industry, or business. The infrastructure of the community could be shared between the organization which has shared concerns or tasks. It is generally managed by a third party or by the combination of one or more organizations in the community.

Advantages of the community cloud model:

- **Cost Effective:** It is cost-effective because the cloud is shared by multiple organizations or communities.
- **Security:** Community cloud provides better security.
- **Shared resources:** It allows you to share resources, infrastructure, etc. with multiple organizations.
- **Collaboration and data sharing:** It is suitable for both collaboration and data sharing.



1.4 Benefits and Challenges of Cloud Computing

Benefits of Cloud Computing:

Cost Savings

Cost saving is one of the biggest Cloud Computing benefits. It helps you to save substantial capital cost as it does not need any physical hardware investments. Also, you do not need trained personnel to maintain the hardware. The buying and managing of equipment is done by the cloud service provider.

Strategic edge

Cloud computing offers a competitive edge over your competitors. It is one of the best advantages of Cloud services that helps you to access the latest applications any time without spending your time and money on installations.

High Speed

Cloud computing allows you to deploy your service quickly in fewer clicks. This faster deployment allows you to get the resources required for your system within fewer minutes. **Back-up and restore data**

Once the data is stored in a Cloud, it is easier to get the back-up and recovery of that, which is otherwise very time taking process on-premise.

Automatic Software Integration

In the cloud, software integration is something that occurs automatically. Therefore, you don't need to take additional efforts to customize and integrate your applications as per your preferences.

Reliability

Reliability is one of the biggest benefits of Cloud hosting. You can always get instantly updated about the changes.

Mobility

Employees who are working on the premises or at the remote locations can easily access all the could services. All they need is an Internet connectivity.

Unlimited storage capacity

The cloud offers almost limitless storage capacity. At any time you can quickly expand your storage capacity with very nominal monthly fees.

Collaboration

The cloud computing platform helps employees who are located in different geographies to collaborate in a highly convenient and secure manner.

Quick Deployment

Last but not least, cloud computing gives you the advantage of rapid deployment. So, when you decide to use the cloud, your entire system can be fully functional in very few minutes. Although, the amount of time taken depends on what kind of technologies are used in your business.

Other Important Benefits of Cloud Computing

Apart from the above, some other Cloud Computing advantages are:

- On-Demand Self-service
- Multi-tenancy
- Offers Resilient Computing
- Fast and effective virtualization
- Provide you low-cost software
- Offers advanced online security
- Location and Device Independence
- Always available, and scales automatically to adjust to the increase in demand
- Allows pay-per-use
- Web-based control & interfaces
- API Access available.

Disadvantages of Cloud Computing

Here, are significant challenges of using Cloud Computing:

Performance Can Vary

When you are working in a cloud environment, your application is running on the server which simultaneously provides resources to other businesses. Any greedy behavior or DDOS attack on your tenant could affect the performance of your shared resource.

Technical Issues

Cloud technology is always prone to an outage and other technical issues. Even, the best cloud service provider companies may face this type of trouble despite maintaining high standards of maintenance.

Security Threat in the Cloud

Another drawback while working with cloud computing services is security risk. Before adopting cloud technology, you should be well aware of the fact that you will be sharing all your company's sensitive information to a third-party cloud computing service provider.

Hackers might access this information.

Downtime

Downtime should also be considered while working with cloud computing. That's because your cloud provider may face power loss, low internet connectivity, service maintenance, etc.

Internet Connectivity

Good Internet connectivity is a must in cloud computing. You can't access cloud without an internet connection. Moreover, you don't have any other way to gather data from the cloud.

Lower Bandwidth

Many cloud storage service providers limit bandwidth usage of their users. So, in case if your organization surpasses the given allowance, the additional charges could be significantly costly

Lacks of Support

Cloud Computing companies fail to provide proper support to the customers. Moreover, they want their user to depend on FAQs or online help, which can be a tedious job for non-technical persons.

1.5 Applications of Cloud Computing: (Assignment to students)

1.6 Requirements of Cloud Computing: (Assignment to students)

1.7 Short notes on (a) Cloud Storage (b) Cloud Adoption

1.8 Cloud and dynamic infrastructure

Cloud and dynamic infrastructure:

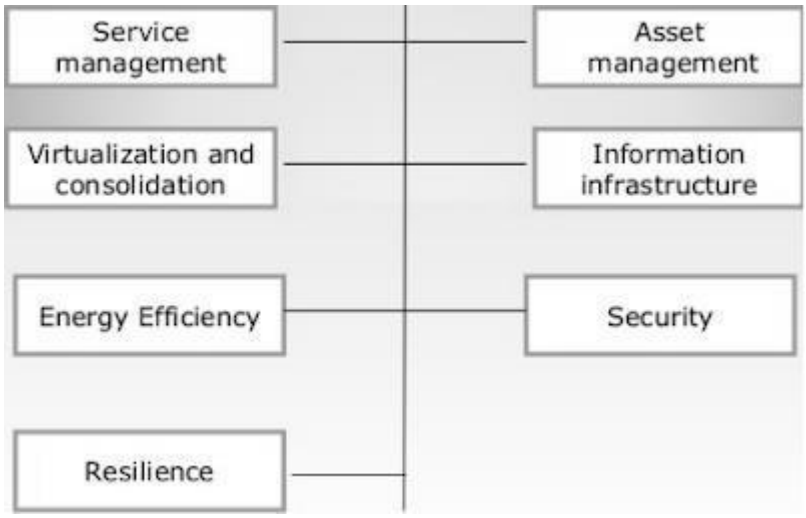


Fig: Cloud and Dynamic Infrastructure

1. **Service management**: This type of special facility or a functionality is provided to the cloud IT services by the cloud service providers. This facility includes visibility, automation and control to delivering the first class IT services.

2. **Asset-Management**: In this the assets or the property which is involved in providing the cloud services are getting managed.

3. **Virtualization and consolidation:** Consolidation is an effort to reduce the cost of a technology by improving its operating efficiency and effectiveness. It means migrating from large number of resources to fewer one, which is done by virtualization technology.

4. **Information Infrastructure**: It helps the business organizations to achieve the following : Information compliance, availability of resources retention and security objectives.

5. **Energy-Efficiency**: Here the IT infrastructure or organization sustainable. It means it is not likely to damage or effect any other thing.

6. **Security**: This cloud infrastructure is responsible for the risk management. Risk management Refers to the risks involved in the services which are being provided by the cloud-service providers.

7. **Resilience**: This infrastructure provides the feature of resilience means the services are resilient. It means the infrastructure is safe from all sides. The IT operations will not be easily get affected.

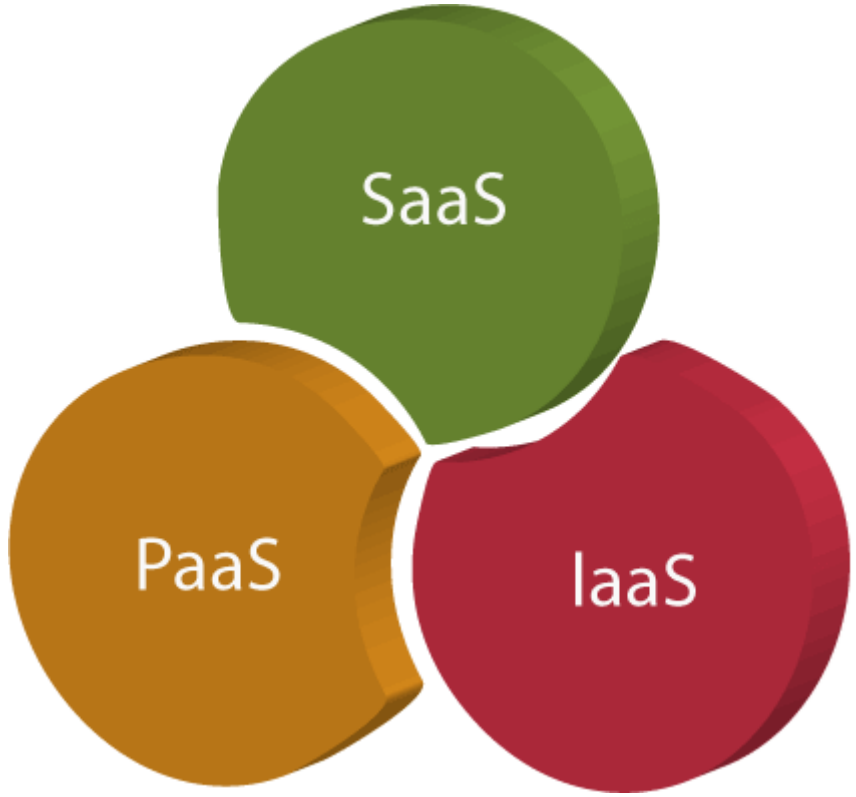
END OF CHAPTER-1

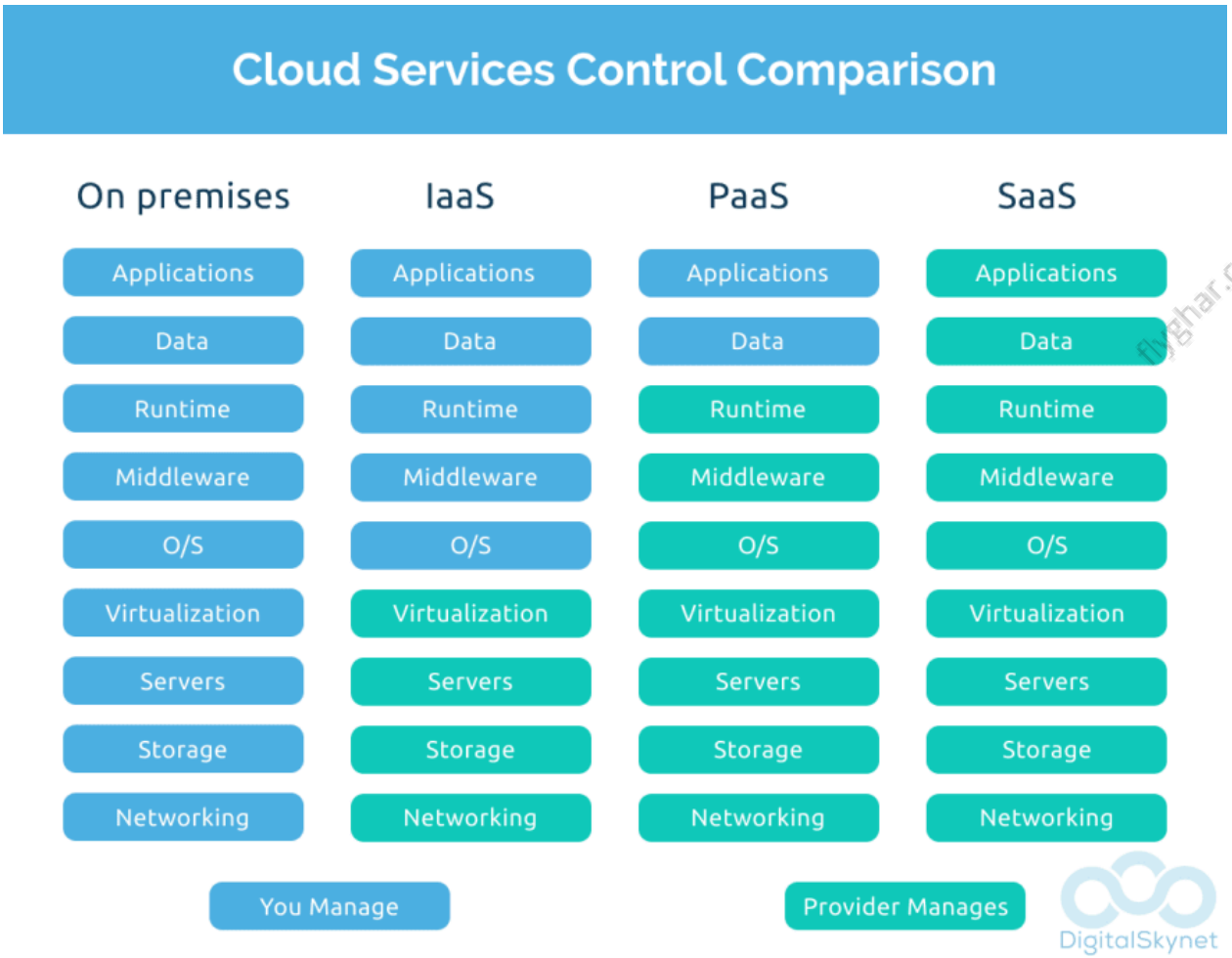
Unit 2: Cloud Computing Architecture

2.1 Cloud Service Models

There are the following three types of cloud service models -

1. **Infrastructure as a Service (IaaS)**
2. **Platform as a Service (PaaS)**
3. **Software as a Service (SaaS)**





1. Infrastructure as a Service | IaaS

Iaas is also known as **Hardware as a Service (HaaS)**. It is one of the layers of the cloud computing platform. It allows customers to outsource their IT infrastructures such as servers, networking, processing, storage, virtual machines, and other resources. Customers access these resources on the Internet using a pay-as-per use model.

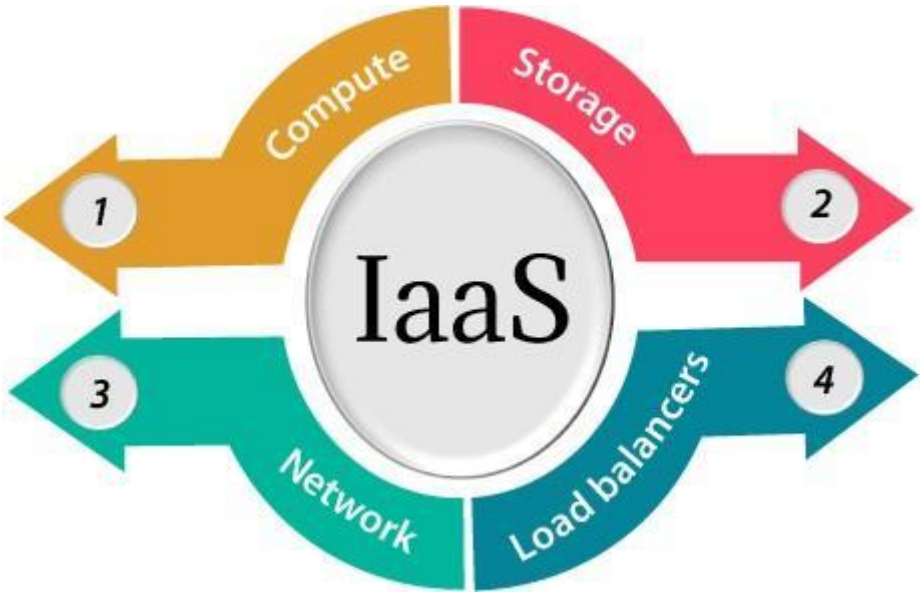
In traditional hosting services, IT infrastructure was rented out for a specific period of time, with pre-determined hardware configuration. The client paid for the configuration and time, regardless of the actual use. With the help of the IaaS cloud computing platform layer, clients can dynamically scale the configuration to meet changing requirements and are billed only for the services actually used.

IaaS cloud computing platform layer eliminates the need for every organization to maintain the IT infrastructure.

IaaS is offered in three models: public, private, and hybrid cloud. The private cloud implies that the infrastructure resides at the customer-premise. In the case of public cloud, it is located at the cloud computing platform vendor's data center, and the hybrid cloud is a combination of the two in which the customer selects the best of both public cloud or private cloud.

IaaS provider provides the following services -

1. **Compute:** Computing as a Service includes virtual central processing units and virtual main memory for the Vms that is provisioned to the end- users.
2. **Storage:** IaaS provider provides back-end storage for storing files.
3. **Network:** Network as a Service (NaaS) provides networking components such as routers, switches, and bridges for the Vms.
4. **Load balancers:** It provides load balancing capability at the infrastructure layer.



Advantages of IaaS cloud computing layer

There are the following advantages of IaaS computing layer -

1. Shared infrastructure

IaaS allows multiple users to share the same physical infrastructure.

2. Web access to the resources

Iaas allows IT users to access resources over the internet.

3. Pay-as-per-use model

IaaS providers provide services based on the pay-as-per-use basis. The users are required to pay for what they have used.

4. Focus on the core business

IaaS providers focus on the organization's core business rather than on IT infrastructure.

5. On-demand scalability

On-demand scalability is one of the biggest advantages of IaaS. Using IaaS, users do not worry about to upgrade software and troubleshoot the issues related to hardware components.

Disadvantages of IaaS cloud computing layer

1. Security

Security is one of the biggest issues in IaaS. Most of the IaaS providers are not able to provide 100% security.

2. Maintenance & Upgrade

Although IaaS service providers maintain the software, but they do not upgrade the software for some organizations.

3. Interoperability issues

It is difficult to migrate VM from one IaaS provider to the other, so the customers might face problem related to vendor lock-in.

Top Iaas Providers who are providing IaaS cloud computing platform

2. Platform as a Service | PaaS

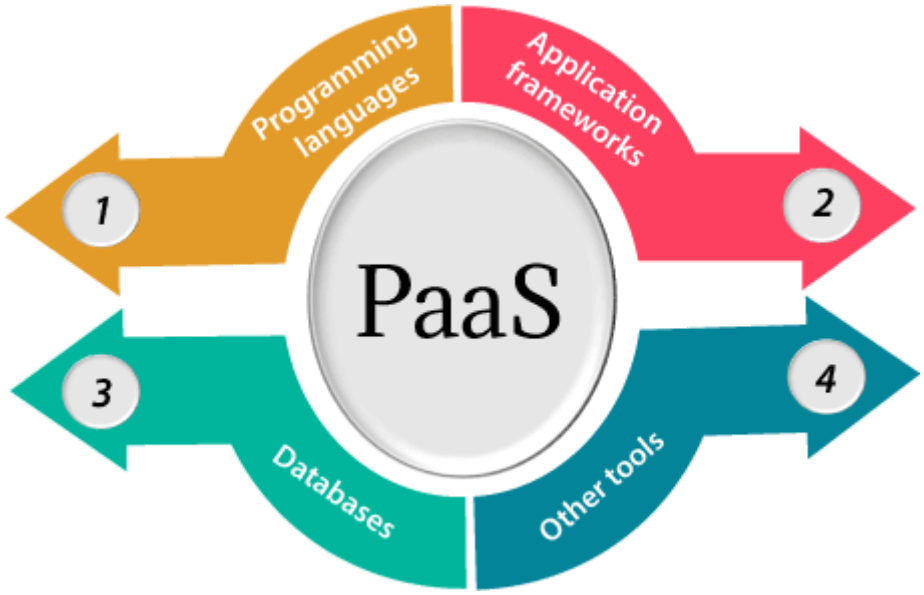
Platform as a Service (PaaS) provides a runtime environment. It allows programmers to easily create, test, run, and deploy web applications. You can purchase these applications from a cloud service provider on a pay-as-per use basis and access them using the Internet

connection. In PaaS, back end scalability is managed by the cloud service provider, so end-users do not need to worry about managing the infrastructure.

PaaS includes infrastructure (servers, storage, and networking) and platform (middleware, development tools, database management systems, business intelligence, and more) to support the web application life cycle.

Example: Google App Engine, Force.com, Joyent, Azure.

PaaS providers provide the Programming languages, Application frameworks, Databases, and Other tools:



1. Programming languages

PaaS providers provide various programming languages for the developers to develop the applications. Some popular programming languages provided by PaaS providers are Java, PHP, Ruby, Perl, and Go.

2. Application frameworks

PaaS providers provide application frameworks to easily understand the application development. Some popular application frameworks provided by PaaS providers are Node.js, Drupal, Joomla, WordPress, Spring, Play, Rack, and Zend.

3. Databases

PaaS providers provide various databases such as ClearDB, PostgreSQL, MongoDB, and Redis to communicate with the applications.

4. Other tools

PaaS providers provide various other tools that are required to develop, test, and deploy the applications.

Advantages of PaaS

There are the following advantages of PaaS -

1) Simplified Development

PaaS allows developers to focus on development and innovation without worrying about infrastructure management.

2) Lower risk

No need for up-front investment in hardware and software. Developers only need a PC and 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 and hence can directly start the projects only.

4) Instant community

PaaS vendors frequently provide online communities where the developer can get the ideas to share experiences and seek advice from others.

5) Scalability

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

Disadvantages of PaaS cloud computing layer

1) Vendor lock-in

One has to write the applications according to the platform provided by the PaaS vendor, so the migration of an application to another PaaS vendor would be a problem.

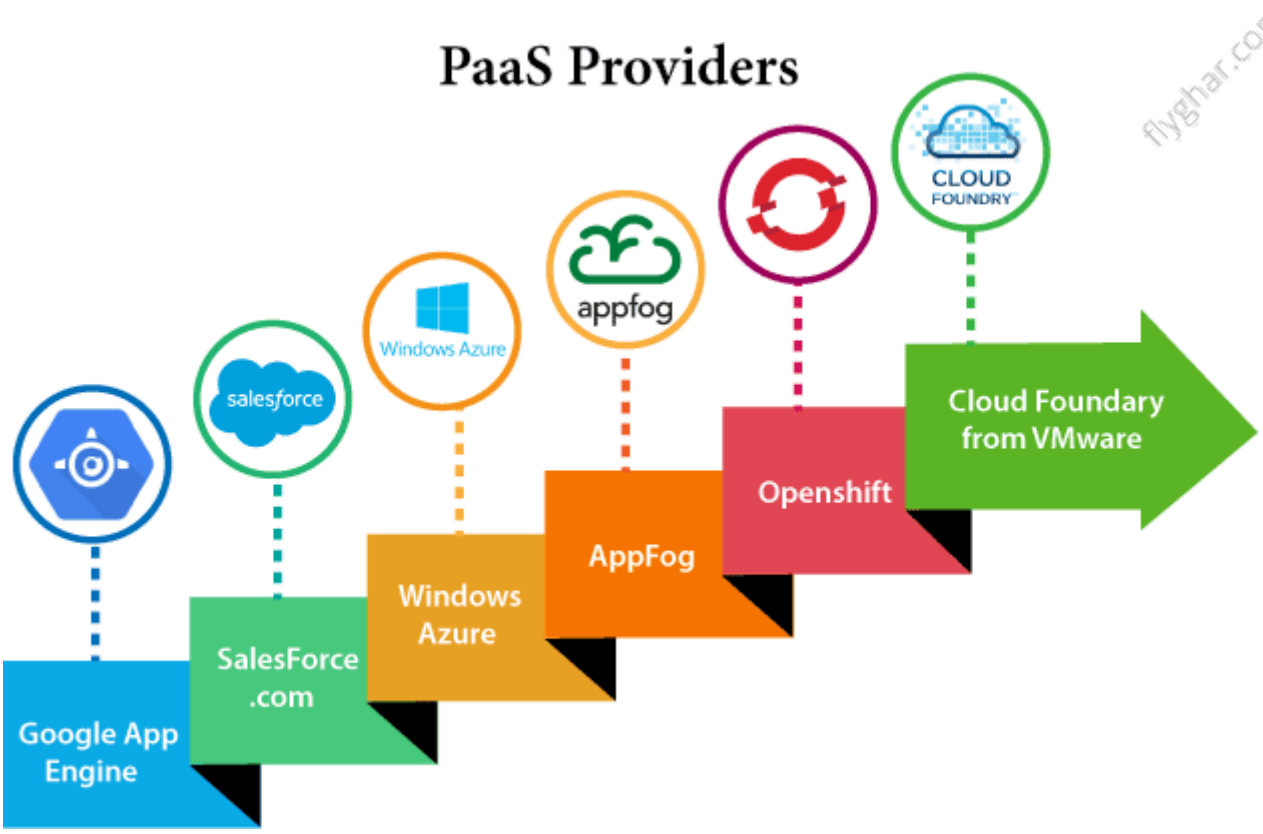
2) Data Privacy

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

3) Integration with the rest of the systems applications

It may happen that some applications are local, and some are in the cloud. So there will be chances of increased complexity when we want to use data which in the cloud with the local data.

Popular PaaS Providers



3. Software as a Service | SaaS

SaaS is also known as "**On-Demand Software**". It is a software distribution model in which services are hosted by a cloud service provider. These services are available to end-users over the internet so, the end-users do not need to install any software on their devices to access these services.

There are the following services provided by SaaS providers -

Business Services - SaaS Provider provides various business services to start-up the business. The SaaS business services include **ERP** (Enterprise Resource Planning), **CRM** (Customer Relationship Management), **billing**, and **sales**.

Document Management - SaaS document management is a software application offered by a third party (SaaS providers) to create, manage, and track electronic documents.

Example: Slack, Samepage, Box, and Zoho Forms.

Social Networks - As we all know, social networking sites are used by the general public, so social networking service providers use SaaS for their convenience and handle the general public's information.

Mail Services - To handle the unpredictable number of users and load on e-mail services, many e-mail providers offering their services using SaaS.

Advantages of SaaS cloud computing layer

1) SaaS is easy to buy

SaaS pricing is based on a monthly fee or annual fee subscription, so it allows organizations to access business functionality at a low cost, which is less than licensed applications.

Unlike traditional software, which is sold as a licensed based with an up-front cost (and often an optional ongoing support fee), SaaS providers are generally pricing the applications using a subscription fee, most commonly a monthly or annually fee.

2. One to Many

SaaS services are offered as a one-to-many model means a single instance of the application is shared by multiple users.

3. Less hardware required for SaaS

The software is hosted remotely, so organizations do not need to invest in additional hardware.

4. Low maintenance required for SaaS

Software as a service removes the need for installation, set-up, and daily maintenance for the organizations. The initial set-up cost for SaaS is typically less than the enterprise software. SaaS vendors are pricing their applications based on some usage parameters, such as a number of users using the application. So SaaS does easy to monitor and automatic updates.

5. No special software or hardware versions required

All users will have the same version of the software and typically access it through the web browser. SaaS reduces IT support costs by outsourcing hardware and software maintenance and support to the IaaS provider.

6. Multidevice support

SaaS services can be accessed from any device such as desktops, laptops, tablets, phones, and thin clients.

7. API Integration

SaaS services easily integrate with other software or services through standard APIs.

8. No client-side installation

SaaS services are accessed directly from the service provider using the internet connection, so do not need to require any software installation.

Disadvantages of SaaS cloud computing layer

1) Security

Actually, data is stored in the cloud, so security may be an issue for some users. However, cloud computing is not more secure than in-house deployment.

2) Latency issue

Since data and applications are stored in the cloud at a variable distance from the end-user, there is a possibility that there may be greater latency when interacting with the application compared to local deployment. Therefore, the SaaS model is not suitable for applications whose demand response time is in milliseconds.

3) Total Dependency on Internet

Without an internet connection, most SaaS applications are not usable.

4) Switching between SaaS vendors is difficult

Switching SaaS vendors involves the difficult and slow task of transferring the very large data files over the internet and then converting and importing them into another SaaS also.

Popular SaaS Providers

2.2 Types of cloud

"Refer to Chapter 1"

2.3 Cloud design and implementation using SOA

What is SOA?

- SOA, or service-oriented architecture, defines a way to make software components reusable and interoperable via service interfaces. Services use common interface standards and an architectural pattern so they can be rapidly incorporated into new applications. This removes tasks from the application developer who previously redeveloped or duplicated existing functionality or had to know how to connect or provide interoperability with existing functions.

Components of SOA:

In SOA, there are mainly three components in SOA:
(i) Service Consumer
(ii) Service Provider
(iii) Service Directory

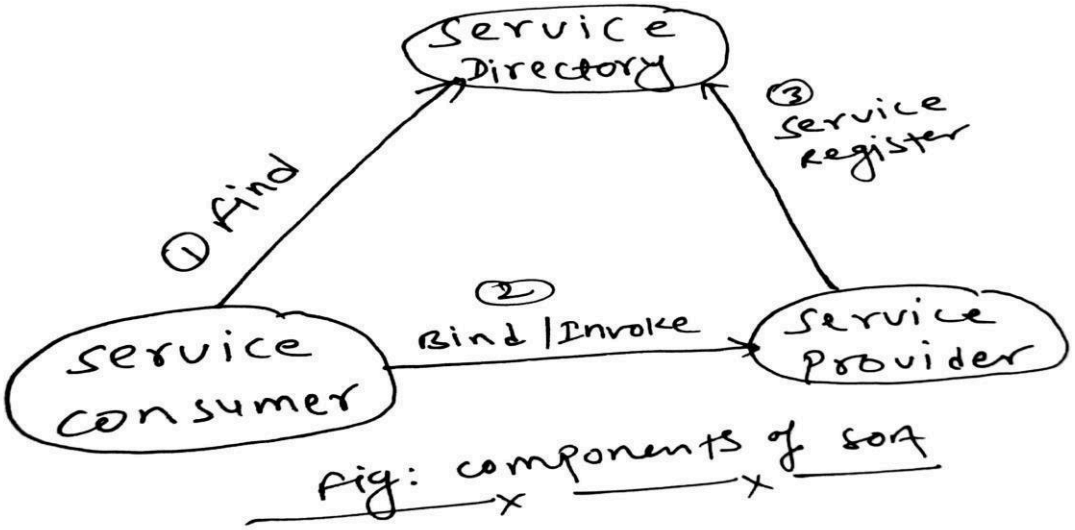


Fig: components of SOA

(i) Service provider: The service provider is the maintainer of the service and the organization that makes available one or more services for others to use. To advertise services, the provider can publish them in a registry, together with a service contract that specifies the nature of the service, how to use it, the requirements for the service, and the fees charged.

(ii) Service Consumer: The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service. The service consumer is an application, service, or some other type of software module that requires a service. It is the entity that initiates the locating of the service in the registry, binding to the service over a transport, and executing the service function. The service consumer executes the service by sending it a request formatted according to the contract.

(iii) Service Registry/Directory: An SOA Service Registry is the core component of SOA Governance. The Service registry allows service providers to discover and communicate with consumers efficiently, creating a link between service providers and service customers. The primary focus of Service Registry is to provide fast, easy access to communication, and to operate among different applications with a limited human intervention. An SOA registry supports the UDDI (Universal Description, Discovery and Integration) specification,
an XML– (Extensible Markup Language) based registry that was developed for the purpose of making systems interoperable. Publication of a service requires proper description of a Web service in terms of business, service, and technical information. Once, the services are published the registry will create the dependencies, associations and versions of these services and metadata.

Service-Oriented Architecture (SOA) Principles

Service-oriented Architecture is as easy as it can be. In SOA, we have *nine design principles* to remain in memory while generating an SOA service:

- **Standardized Service Contract**

  Services adhere to a service-description. A service needs to have some information that defines what the service is about.
- **Loose Coupling**

  Services minimize dependencies on each other. So if the service functionality breaks at several points in time, this should not crush the client application or stop it from running.
- **Service Abstraction**

  Services wrap the logic they encapsulate from the unknown external world. The service shouldn't show how it performs its functionality.
- **Service Reusability**

  Logic is divided into services to maximize re-use.
- **Service Autonomy**

  Services must control the logic they encapsulate.
- **Service Statelessness**

  Services should stay stateless. This determines that services should not keep data from one state to the other. This would be required to be done from each client application.
- **Service Discoverability**

  Services can be discovered (usually in a service registry). We have previously viewed this in the theory of the UDDI, which performs a registry which can contain information about the web service.
- **Service Composability**

  It breaks large problems into tiny problems.
- **Service Interoperability**

Services should use standards that provide different supporters to use the service. This is examined so obviously these days that it is frequently dropped as a principle.

Advantages and Disadvantages of SOA

Given below are the advantages and disadvantages mentioned:

Advantages:

- **Maintenance is Easy:** Editing and updating any service implemented under SOA architecture is easy. You don’t need to update your system. A third party maintains the service, and any amendment in this service won’t have an effect on your system. In most cases, the previous API work because it is functioning before.
- **Quality of Code Improved:** As services run freelance of our system, they have their own variety of code; therefore, our code is prevented from redundancy. Also, our code becomes error-free.
- **Platform Independence:** Services communicate with alternative applications

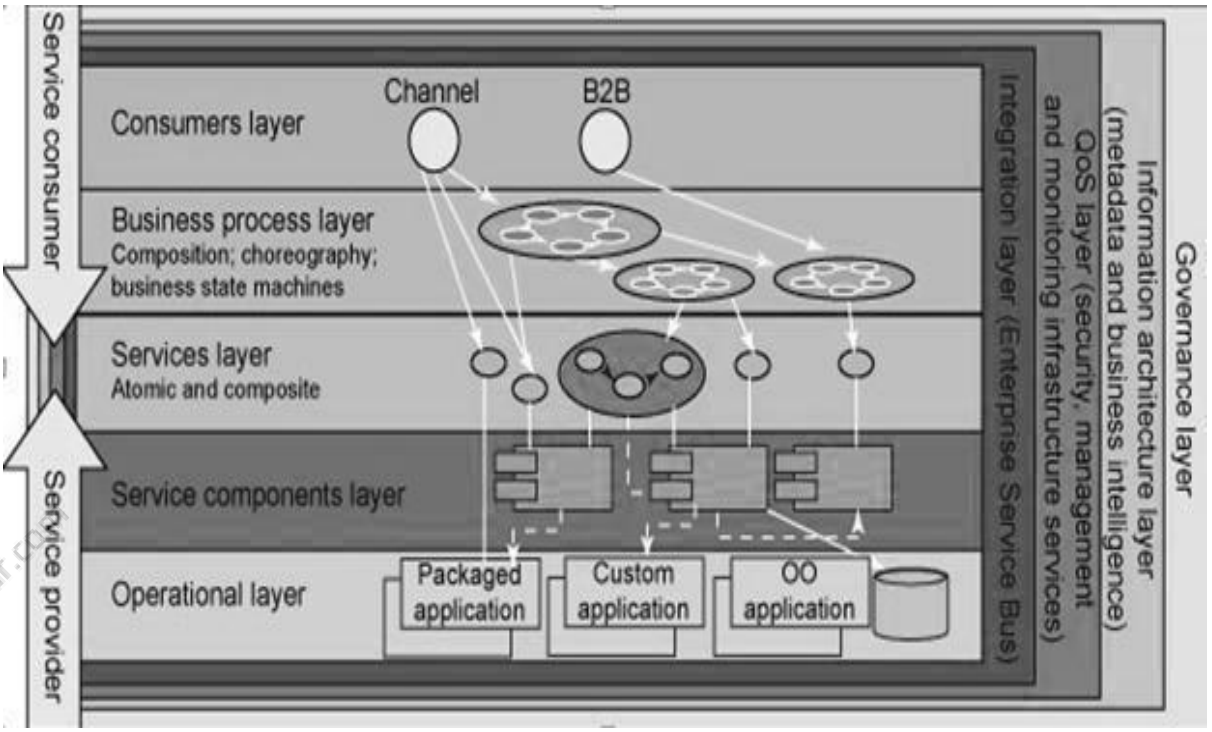
through a common language, which implies it’s freelance of the platform on which that application is running. Services can provide API in different languages, e.g. PHP, JavaScript, etc.

- **Scalable:** If any service obtains several users, it is often simply scalable by attaching additional servers. This will create service out there all time to the users.
- **Reliable:** Services square measure typically tiny size as compared to the full-fledged application. So it's easier to correct and check the freelance services.
- **Same Directory Structure:** Services have an equivalent directory structure so customers can access the service information from an equivalent directory on every occasion. If any service has modified its location, then the additional directory remains the same. This is very helpful for consumers.
- **Independent of Other Services:** Services generated using SOA principles are independent of each other. So services are often utilized by multiple applications at an equivalent time.

Disadvantages:

- **High Bandwidth Server:** Therefore, net service sends and receives messages and knowledge often times, so it simply reaches high requests per day. So it involves a high-speed server with plenty of information measures to run an internet service.
- **Extra Overload:** In SOA, all inputs square measure its validity before it's sent to the service. If you are victimization multiple services, then it'll overload your system with further computation.
- **High Cost:** It is expensive in terms of human resources, development, and technology.

SOA Reference Model:



(Description in Class)

Chapter -4
CLOUD Programming Model

INTRODUCTION:

Usually, there are some certain strategies to recognize where cloud services will use and how it achieves for business goals. Particularly, to reach your goals in business are possible through infrastructure as a service and Platform as a Service, private and public cloud, testing and development, big data analytics course, file storage, disaster recovery, and backup of the cloud. Through the essential servers and computer programming of the cloud, it is efficient to access all the facilities offered by this computing technology. Additionally, cloud course development empowers the user to acquire their application for quick marketing. To mention that, there is no data loss due to hardware failures because of backups in the network. Also, it uses the organization's saving, and remote servers.

Characteristics of Programming Models in Cloud Computing

Customarily, there are main five characteristics of cloud. They are the following:

- On-demand self-service
- Broad Network Access
- Resource Pooling
- Rapid Elasticity
- Measure Service

1. On-demand self-service

With this feature, one can individually establish the automatic server time and network storage capabilities of Oracle Database Cloud Service. This is possible without interacting with the humans and with the service providers.

2. Broad Network Access

In this, the competencies are obtainable over the network and accessed via standard mechanisms that stimulate use by various thin or thick client platforms. For instance, mobile phones, tablets, laptops, and workstations.

3. Resource Pooling

To serve for multiple consumers, it is necessary to pool the resources of provider's computing with multi-talented models. Likewise, this model includes various dynamic resources of physical and virtual that assigns and reassigns as per the demands of the consumers. For example, the resources include memory, storage, processing, and network bandwidth.

4. Rapid Elasticity

In some cases, the capabilities can be automatically released. On the other hand, it is essential for quick inward and outward scaling corresponding with the demand. To the consumer, the capabilities obtainable for provisioning frequently appear to be limitless and can be adopted in any time for any quantity.

5. Measured Service

Specifically, cloud systems habitually control and enhance the usage of the resource via appropriate to the type of service through the level of abstraction. This service includes processing, storage, bandwidth, and active user accounts. To mention that, the usage of the resource is monitor, control, and report that provides transparency for the provider as well as the consumer utilized services.

Programming Language of Cloud Computing

To point out, the cloud is a concept level of the operating system than the language concept. But the Programming Models in Cloud Computing programming includes some computer programming languages for accessing and operating the cloud. Such languages are Perl, learn javascript, AJAX, ASP, Java, PHP and learn MySQL. Additionally, if you wish to implement a new programming language that is specifically designed to operate the cloud should optimize for cloud computing and must be easy to learn, efficient, modern, fast, and powerful. As a result, cloud computing programming, the latest technology is much useful to improve your career and use to access and operate the cloud.

4.1Thread Programming:

modern software. Programmers use multiple threads of control for a variety of reasons: to build responsive servers that interact with multiple clients, to run computations in parallel on a multiprocessor for performance, and as a structuring mechanism for implementing rich user interfaces. In general, threads are useful whenever the software needs to manage a set of tasks with varying interaction latencies, exploit multiple physical resources, or execute largely independent tasks in response to multiple external events. Developing parallel applications requires an understanding of the problem and its logical structure. Understanding the dependencies and the correlation of tasks within an application is fundamental for designing the right program structure and to introduce parallelism where appropriate. Decomposition is a useful technique that helps to understand whether a problem is divided into components (or tasks) that can be executed concurrently. The two main decomposition/partitioning techniques used area: domain and functional decompositions.

Working Mechanism of Multithreading:

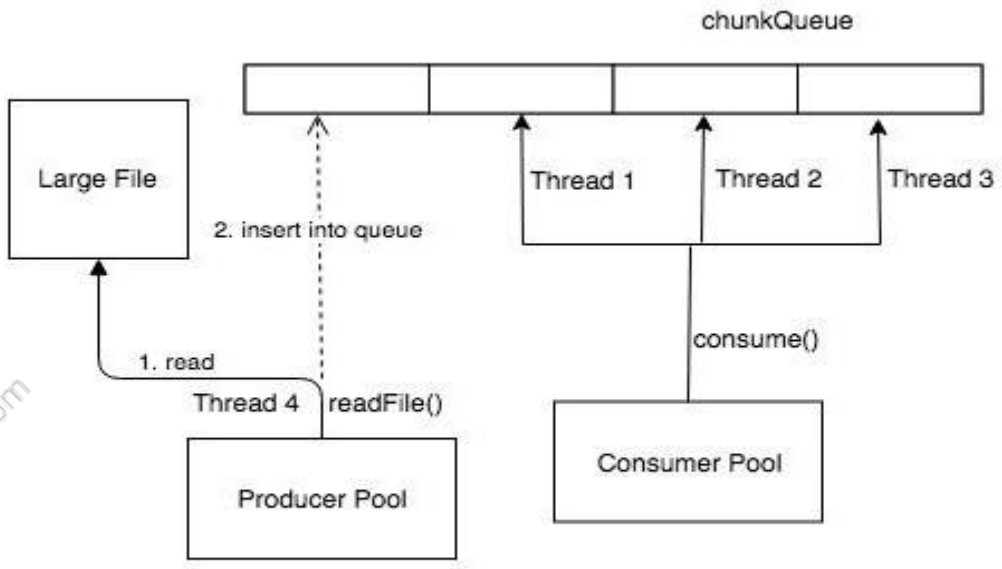


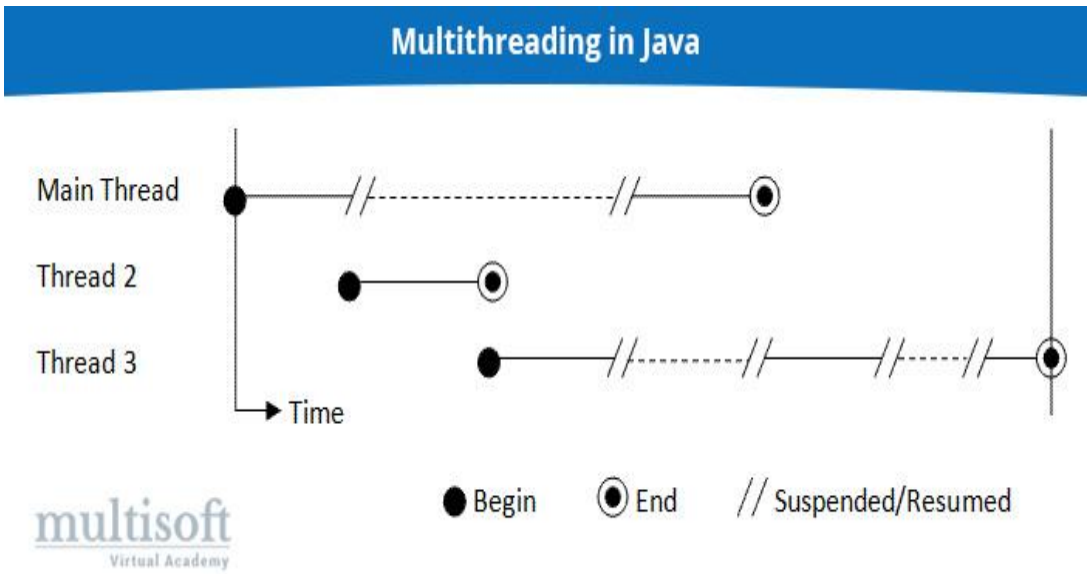
Fig: Producer & Consumer with a blocking queue

The model shown in the figure above uses a blocking queue. The producer reads a large file which is the dataset and divides the file into chunks. Each chunk are inserted to chunk queue.

The consumer pool consisting of a number of worker threads consumes the chunk of data and process them. When the worker threads in the consumer pool are done with processing the main thread aggregates the result and writes to an output file.

Advantages and Disadvantages of Multithreding:

We are heading towards the critical shortage of time where multitasking has become synonymous with efficiency. Similarly, in the programming world also, a program performing a single task is not sufficient and good enough. An efficient program is the one that lets you execute multiple tasks using the same program. Such an efficient program can be created using the Java programming language since Java is a multithreaded programming language. Also, Java is the first programming language that introduced the concept of multithreading.



The primary function of multithreading is to simultaneously run or execute multiple tasks. These tasks are represented as threads in a Java program and have a separate execution path. Also, handling of multithreaded Java programs is easy because you can decide the sequence in which execution of Java threads take place.

Following are some of the common advantages of multithreading:

- Enhanced performance by decreased development time
- Simplified and streamlined program coding
- Improvised GUI responsiveness
- Simultaneous and parallelized occurrence of tasks
- Better use of cache storage by utilization of resources
- Decreased cost of maintenance
- Better use of CPU resource

Multithreading does not only provide you with benefits, it has its disadvantages too. Let us go through some common disadvantages:

- Complex debugging and testing processes

- Overhead switching of context
- Increased potential for deadlock occurrence
- Increased difficulty level in writing a program
- Unpredictable results

4.2 MAP Reduce Programming:

In MapReduce Programming Model implementation, we read the data file and split the data in each line using mapper function. The output of mapper is passed to partitioner which aggregates the values associated with the key. The Reducer function then uses that output from the partitioner to merge and aggregate the overall result.

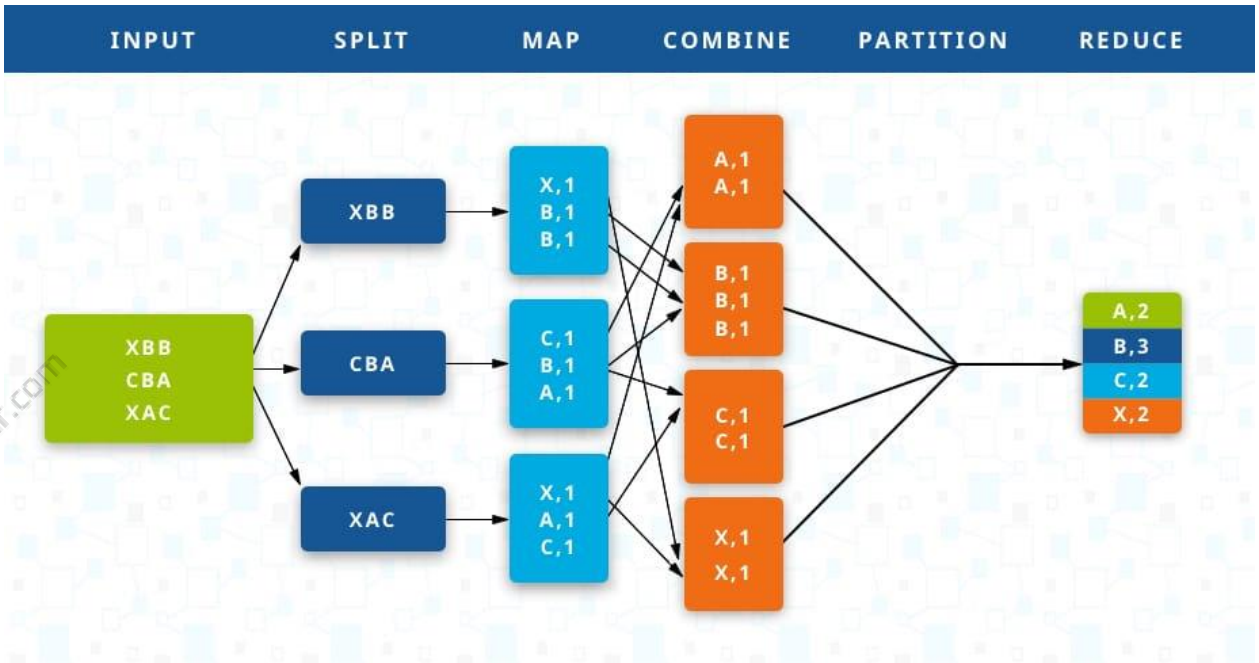
MapReduce is a programming model or pattern within the Hadoop framework that is used to access big data stored in the Hadoop File System (HDFS). It is a core component, integral to the functioning of the Hadoop framework.

MapReduce facilitates concurrent processing by splitting petabytes of data into smaller chunks, and processing them in parallel on Hadoop commodity servers. In the end, it aggregates all the data from multiple servers to return a consolidated output back to the application.

How MapReduce Works

At the crux of MapReduce are two functions: Map and Reduce. They are sequenced one after the other.

- The Map function takes input from the disk as <key,value> pairs, processes them, and produces another set of intermediate <key,value> pairs as output.
- The Reduce function also takes inputs as <key,value> pairs, and produces <key,value> pairs as output.



The types of keys and values differ based on the use case. All inputs and outputs are stored in the HDFS. While the map is a mandatory step to filter and sort the initial data, the reduce function is optional.

<k1, v1> -> Map() -> list(<k2, v2>)
<k2, list(v2)> -> Reduce() -> list(<k3, v3>)

Mappers and Reducers are the Hadoop servers that run the Map and Reduce functions respectively. It doesn't matter if these are the same or different servers.

Map

The input data is first split into smaller blocks. Each block is then assigned to a mapper for processing.

For example, if a file has 100 records to be processed, 100 mappers can run together to process one record each. Or maybe 50 mappers can run together to process two records each. The Hadoop framework decides how many mappers to use, based on the size of the data to be processed and the memory block available on each mapper server.

Reduce

After all the mappers complete processing, the framework shuffles and sorts the results before passing them on to the reducers. A reducer cannot start while a mapper is still in progress. All the map output values that have the same key are assigned to a single reducer, which then aggregates the values for that key.

Combine and Partition

There are two intermediate steps between Map and Reduce.

Combine is an optional process. The combiner is a reducer that runs individually on each mapper server. It reduces the data on each mapper further to a simplified form before passing it downstream.

This makes shuffling and sorting easier as there is less data to work with. Often, the combiner class is set to the reducer class itself, due to the cumulative and associative functions in the reduce function. However, if needed, the combiner can be a separate class as well.

Partition is the process that translates the <key, value> pairs resulting from mappers to another set of <key, value> pairs to feed into the reducer. It decides how the data has to be presented to the reducer and also assigns it to a particular reducer.

The default partitioner determines the hash value for the key, resulting from the mapper, and assigns a partition based on this hash value. There are as many partitions as there are reducers. So, once the partitioning is complete, the data from each partition is sent to a specific reducer.

Parallel Processing in Map reduce

Parallel map over input: Input data is processed such that key/value pairs are processed one by one. It is well known that this pattern of a list map is amenable to total data parallelism. That is, in principle, the list map may be executed in parallel at the granularity level of single elements. Clearly, MAP must be a pure function so that the order of processing key/value pairs does not affect the result of the map phase and communication between the different threads can be avoided.

Parallel grouping of intermediate data: The grouping of intermediate data by key, as needed for the reduce phase, is essentially a sorting problem. Various parallel sorting models exist. If we assume a distributed map phase, then it is reasonable to anticipate grouping to be aligned with distributed mapping. That is, grouping could be performed for any fraction of intermediate data and distributed grouping results could be merged centrally, just as in the case of a parallelmerge- all strategy. Parallel map over groups: Reduction is performed for each group (which is a key with a list of values) separately. Again, the pattern of a list map applies here; total data parallelism is admitted for the reduce phase— just as much as for the map phase.

Parallel reduction per group: Let us assume that REDUCE defines a proper reduction That is, REDUCE reveals itself as an operation that collapses a list into a single value by means of an associative operation and its unit. Then, each application of REDUCE can be massively parallelized by computing sub-reductions in a tree-like structure while applying the associative operation at the nodes. If the binary operation is also commutative, then the order of combining results from sub-reductions can be arbitrary. Given that we already parallelize reduction at the granularity of groups, it is non-obvious that parallel reduction of the values per key could be attractive.

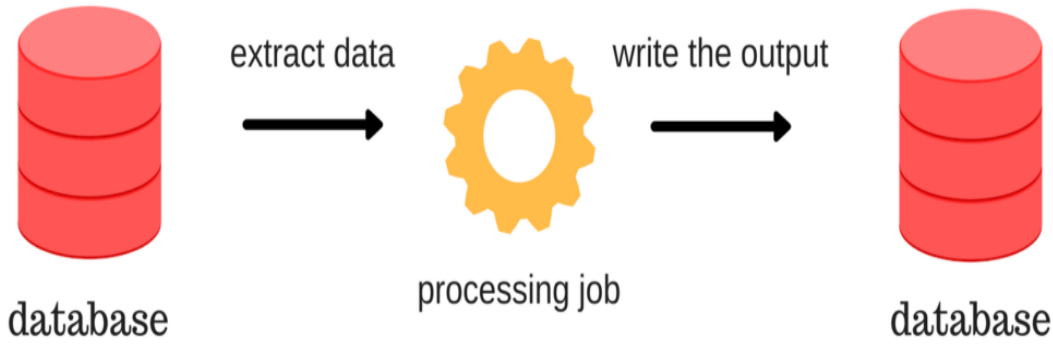
Enterprise batch processing using Map-Reduce

Today, the volume of data is often too big for a single server – node – to process. Therefore, there was a need to develop code that runs on multiple nodes. Writing distributed systems is an endless array of problems, so people developed multiple frameworks to make our lives easier. MapReduce is a framework that allows the user to write code that is executed on multiple nodes without having to worry about fault tolerance, reliability, synchronization or availability.

Batch processing

There are a lot of use cases for a system described in the introduction, but the focus of this post will be on data processing – more specifically, batch processing. Batch processing is an automated job that does some computation, usually done as a periodical job. It runs the processing code on a set of inputs, called a batch. Usually, the job will read the batch data from a database and store the result in the same or different database.

An example of a batch processing job could be reading all the sale logs from an online shop for a single day and aggregating it into statistics for that day (number of users per country, the average spent amount, etc.). Doing this as a daily job could give insights into customer trends.



MapReduce

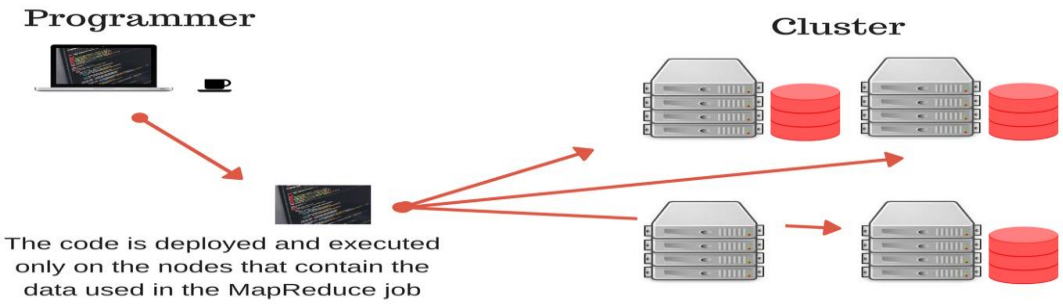
MapReduce is a programming model that was introduced in a white paper by Google in 2004. Today, it is implemented in various data processing and storing systems (Hadoop, Spark, MongoDB, …) and it is a foundational building block of most big data batch processing systems.

For MapReduce to be able to do computation on large amounts of data, it has to be a distributed model that executes its code on multiple nodes. This allows the computation to handle larger amounts of data by adding more machines – horizontal scaling. This is different from vertical scaling, which implies increasing the performance of a single machine.

Execution

In order to decrease the duration of our distributed computation, MapReduce tries to reduce shuffling (moving) the data from one node to another by distributing the computation so that it is done on the same node where the data is stored. This way, the data stays on the same node, but the code is moved via the network. This is ideal because the code is much smaller than the data.

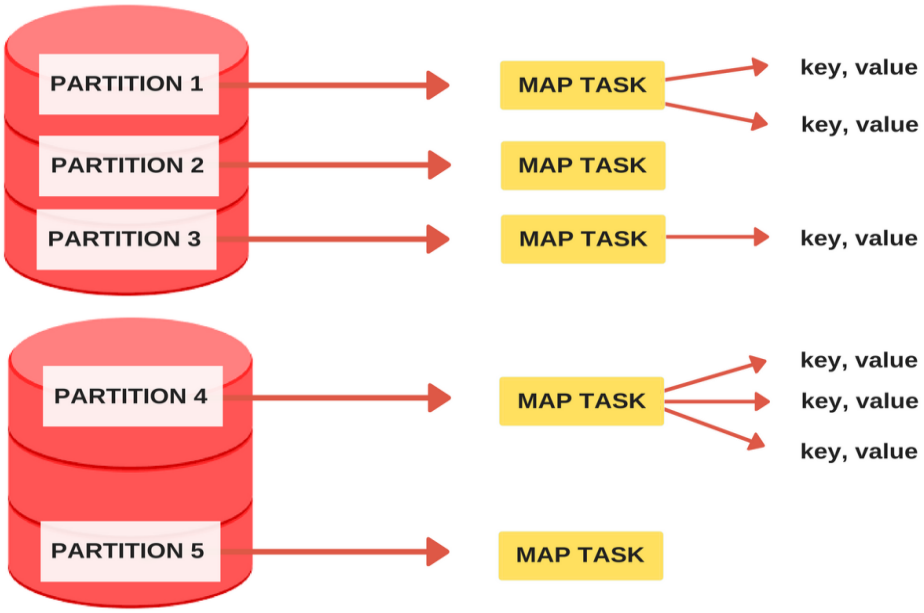
To run a MapReduce job, the user has to implement two functions, map and reduce, and those implemented functions are distributed to nodes that contain the data by the MapReduce framework. Each node runs (executes) the given functions on the data it has in order the minimize network traffic (shuffling data).



The computation performance of MapReduce comes at the cost of its expressivity. When writing a MapReduce job we have to follow the strict interface (return and input data structure) of the map and the reduce functions. The map phase generates key-value data pairs from the input data (partitions), which are then grouped by key and used in the reduce phase by the reduce task. Everything except the interface of the functions is programmable by the user.

Map

Hadoop, along with its many other features, had the first open-source implementation of MapReduce. It also has its own distributed file storage called HDFS. In Hadoop, the typical input into a MapReduce job is a directory in HDFS. In order to increase parallelization, each directory is made up of smaller units called partitions and each partition can be processed separately by a map task (the process that executes the map function). This is hidden from the user, but it is important to be aware of it because the number of partitions can affect the speed of execution.

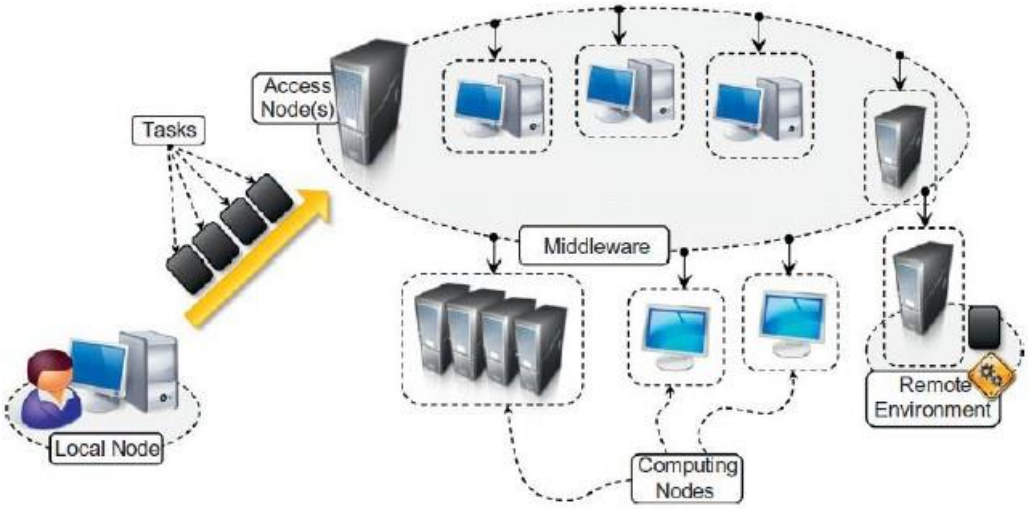


The map task (mapper) is called once for every input partition and its job is to extract key-value pairs from the input partition. The mapper can generate any number of key-value pairs from a single input (including zero, see the figure above). The user only needs to define the code inside the mapper. Below, we see an example of a simple mapper that takes the input partition and outputs each word as a key with value 1.

4.3 Task Programming:

Task computing is a wide area of distributed system programming encompassing several different models of architecting distributed applications, A task represents a program, which require input files and produce output files as a result of its execution.
Applications are then constituted of a collection of tasks. These are submitted for execution and their output data are collected at the end of their execution.

A task identifies one or more operations that produce a distinct output and that can be isolated as a single logical unit.
Inpractice,a task is represented as a distinct unit of code,or a program, that can be separated and executed in are mote runtime environment.
Multithreaded programming is mainly concerned with providing a support for parallelism within a single machine. Task computing provides distribution by harnessing the compute power of several computing nodes. Hence, the presence of a distributed infrastructure is explicit in this model.
Now clouds have emerged as an attractive solution to obtain a huge computing power on demand for the execution of distributed applications. To achieve it, suitable middleware is needed. A reference scenario for task computing is depicted in **Figure.**



The middleware is a software layer that enables the coordinated use of multiple resources, which are drawn from a datacentre or geographically distributed networked computers.
A user submits the collection of tasks to the access point(s) of the middleware, which will take care of scheduling and monitoring the execution of tasks.
Each computing resource provides an appropriate runtime environment.
Task submission is done using the APIs provided by the middleware, whether a Web or programming language interface.
Appropriate APIs are also provided to monitor task status and collect their results upon completion.
It is possible to identify a set of common operations that the middleware needs to support the creation and execution of task-based applications. These operations are:
• Coordinating and scheduling tasks for execution on a set of remote nodes
• Moving programs to remote nodes and managing their dependencies
• Creating an environment for execution of tasks on the remote nodes
• Monitoring each task's execution and informing the user about its status
• Access to the output produced by the task.

Characterizing a task

A task represents a component of an application that can be logically isolated and executed separately.
A task canbe represented by different elements:
• A shell script composing together the execution of several applications
• A single program
• A unit of code(aJava/C11/.NET class)that executes within the context of a specific runtime environment.
A task is characterized by input files, executable code (programs, shell scripts, etc.), and output files.
The runtime environment in which tasks execute is the operating system or an equivalent sandboxed environment.
A task may also need specific software appliances on the remote execution nodes.

Computing categories

These categories provide an overall view of the characteristics of the problems. They implicitly impose requirements on the infrastructure and the middleware.
Applications falling into this category are:

1 High-performance computing
2 High-throughput computing
3 Many-task computing

1 High-performance computing

High-performance computing(HPC) is the use of distributed computing facilities for solving problems that need large computing power.

The general profile of HPC applications is constituted by a large collection of compute-intensive tasks that need to be processed in a short period of time.
The metric stoe valuate HPC systems are floating-point operations per second(FLOPS), nowtera-FLOPSorevenpeta-FLOPS,whichidentifythenumberoffloating-point operations per second.
Ex: supercomputers and clusters are specifically designed to support HPC applications that are developed to solve "Grand Challenge" problems in science and engineering.

2 High-throughput computing

High-throughput computing(HTC) is the use of distributed computing facilities for applications requiring large computing power over along period of time.
HTC systems need to be robust and to reliably operate over a long time scale.
The general profile of HTC applications is that they are made up of a large number of tasks of which the execution can last for a considerable amount of time.
Ex: scientific simulations or statistical analyses.
It is quite common to have independent tasks that can be scheduled in distributed resources because they do not need to communicate.
HTC systems measure their performance in terms of jobs completed per month.

3 Many-task computing

MTC denotes high-performance computations comprising multiple distinct activities coupled via file system operations.
MTC is the heterogeneity of tasks that might be of different nature: Tasks may be small or large, single-processoror multiprocessor, compute-intensive or data-intensive, static or dynamic, homogeneous or heterogeneous.
MTC applications includes loosely coupled applications that are communication-intensive but not naturally expressed using the message-passing interface.
It aims to bridge the gap between HPC and HTC. MTC is similar to HTC, but it concentrates on the use of many computing resources over a short period of time to accomplish many computational tasks.

Chapter-5 "Cloud Security"

Cloud security, also known as *cloud computing security*, is the practice of protecting cloud-based data, applications and infrastructure from cyber attacks and cyber threats. A cloud service providing vendor combines different parameters to ensure the consumers of their service are getting cloud security as services. The cloud security consists of:

(1) Set of policies
(2) Controls
(3) Procedures
(4) Technologies.

This all things work together to protect cloud based system, data and infrastructures. Level of cloud security also depends upon the individual cloud service provider or the cloud security solutions.

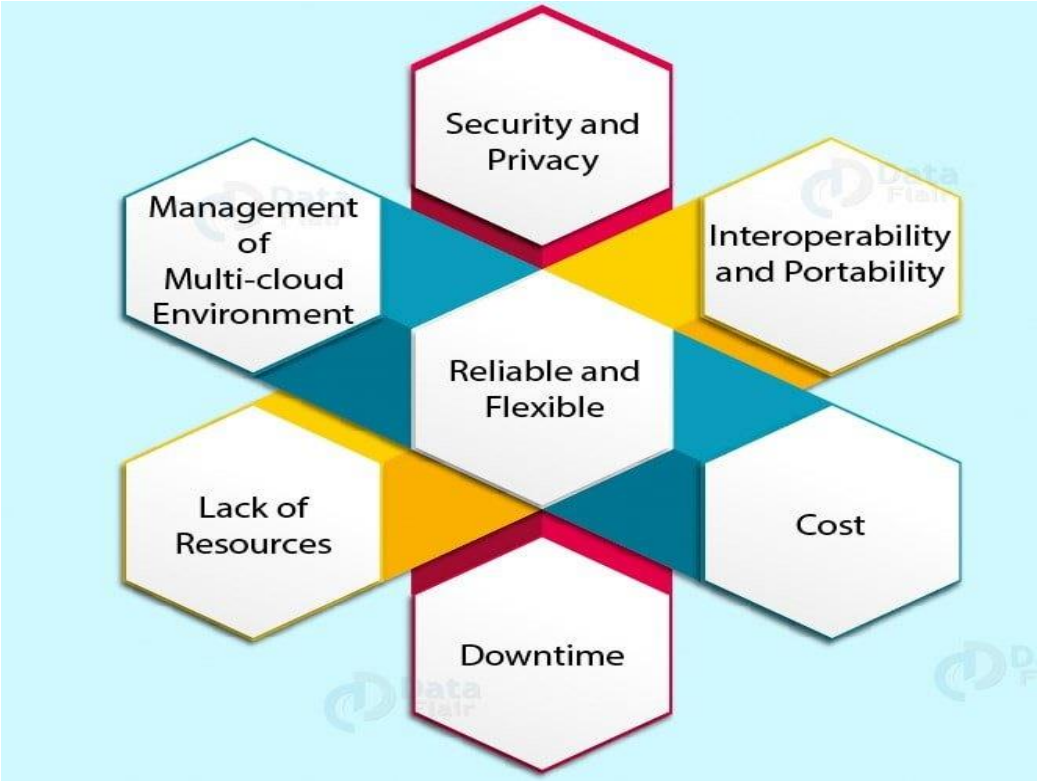
Cloud Security Challenges/Risks/Issues:

The Cloud Computing is getting implemented in almost all companies as the companies are in need to store the data. A tremendous amount of data generate and store by the companies. So there are lots of security issues faced by them.

To improve the cloud computing management the companies can include establishment to simplify and automate the process.

The Major Challenges are:

- Security & Privacy
- Interoperability & Portability
- Reliable and flexible
- Cost
- Downtime
- Lack of resources
- Management of Multi-Cloud Environment



i. Security and Privacy of Cloud

The data store in the cloud must secure and provide full confidentiality. The customers rely on the cloud provider so much. This means that the cloud provider should take necessary security measures to secure the data of the customers.

Securities are also the responsibility of the customer as they should provide a strong password, should not share the password with anyone, and regularly change the password when we did. If the data is outside the firewall there may be some issues which can eliminate by the cloud provider.

Hacking and malware are also one of the major problems as it can affect multiple customers. Hacking can lead to data loss; disrupt the encrypted file system and many other problems.

ii. Interoperability and Portability

The customer must be provided with the services of migration in and out of the cloud. There should be no bond period as it can create a hindrance for the customers. The cloud should have the ability to provide facilities on the premises.

One of the Cloud challenges is remote access which can eliminate by the cloud provider so that the customer can access the cloud from anywhere security.

iii. Reliable and Flexible

Reliability and flexibility are also one of the challenges of cloud customers and it can eliminate in a way that the data provided to the cloud should not leak and the host should provide the reliability to the customers. To eliminate this challenge the services provided by the third party should be monitored and supervision should be done on performance, robustness and business dependency.

iv. Cost

Cloud computing is affordable but modifying the cloud to the customer's demand can be sometimes expensive.

Moreover, it can cause hindrance to the small-scale organization is modifying the cloud as per their demand can sometimes cost more. In addition, transferring of data from the Cloud to the premises can also sometimes be costly.

v. Downtime

Downtime is the common challenges of cloud computing as no cloud provider guarantees a platform that is free from downtime. Internet connection also plays an important role as if a company has an untrustworthy internet connection then there may be a problem as they can face downtime.

vi. Lack of resources

Lack of resources and expertise is also one of the major challenges faced by the cloud industry and many companies are hoping to overcome this challenge by hiring more workers which are more experienced. These workers will not only help to eliminate the challenges of the companies but also they will train existing staff to benefit the company. Today many IT workers are working to boost the cloud computing expertise and CEO of the company is finding it difficult as the workers are not much skilled.

It believes that workers with knowledge of the latest development and the technologies related to it will become more valuable in business.

vii. Management of Multi-Cloud Environment

Companies nowadays do not use a single cloud instead they are using multiple clouds. On an average company are using 4.8 different public and private clouds due to which their management is hindered. When a company uses multi-cloud there are so many complexities faced by the IT team. This Cloud challenge can eliminate by training employees, utilization of proper tools, and doing research.

So, this was all about Risk and Challenges of Cloud Computing. Hope you like our explanation.

Software As Service (Saas Security)

The seven security issues which one should discuss with a cloud-computing vendor:

1. Privileged user access —inquire about who has specialized access to data, and about the hiring and management of such administrators.
2. Regulatory compliance—make sure that the vendor is willing to undergo external audits and/or security certifications.
3. Data location—does the provider allow for any control over the location of data?

4. Data segregation —make sure that encryption is available at all stages, and that these encryption schemes were designed and tested by experienced professionals.
5. Recovery —Find out what will happen to data in the case of a disaster. Do they offer complete restoration? If so, how long would that take?
6. Investigative support —Does the vendor have the ability to investigate any inappropriate or illegal activity?
7. Long-term viability —What will happen to data if the company goes out of business? How will data be returned, and in what format?

To address the security issues listed above, SaaS providers will need to incorporate and enhance security practices used by the managed service providers and develop new ones as the cloud computing environment evolves. The baseline security practices for the SaaS environment as currently formulated are discussed in the following sections.

- **Security Management (People)**: One of the most important actions for a security team is to develop a formal charter for the security organization and program. This will foster a shared vision among the team of what security leadership is driving toward and expects, and will also foster "ownership" in the success of the collective team. The charter should be aligned with the strategic plan of the organization or company the security team works for. Lack of clearly defined roles and responsibilities, and agreement on expectations, can result in a general feeling of loss and confusion among the security team about what is expected of them, how their skills and experienced can be leveraged, and meeting their performance goals. Morale among the team and pride in the team is lowered, and security suffers as a result.
- **Security Governance**: A security steering committee should be developed whose objective is to focus on providing guidance about security initiatives and alignment with business and IT strategies. A charter for the security team is typically one of the first deliverables from the steering committee. This charter must clearly define the roles and responsibilities of the security team and other groups involved in performing information security functions. Lack of a formalized strategy can lead to an unsustainable operating model and security level as it evolves. In addition, lack of attention to security governance can result in key needs of the business not being met, including but not limited to, risk management, security monitoring, application security, and sales support. Lack of proper governance and management of duties can also result in potential security risks being left unaddressed and opportunities to improve the business being missed because the security team is not focused on the key security functions and activities that are critical to the business.
- **Risk Management**: Effective risk management entails identification of technology assets; identification of data and its links to business processes, applications, and data stores; and assignment of ownership and custodial responsibilities. Actions should also include maintaining a repository of information assets. Owners have authority and accountability for information assets including protection requirements, and custodians implement confidentiality, integrity, availability, and privacy controls. A formal risk assessment process should be created that allocates security resources linked to business continuity.
- **Risk Assessment**: Security risk assessment is critical to helping the information security organization make informed decisions when balancing the dueling priorities of business utility and protection of assets. Lack of attention to completing formalized risk assessments can contribute to an increase in information security audit findings, can jeopardize certification goals, and can lead to inefficient and ineffective selection of security controls that may not adequately mitigate information security risks to an acceptable level. A formal information security risk management process should proactively assess information security risks as well as plan and manage them on a periodic or as-needed basis. More detailed and technical security risk assessments in the form of threat modeling should also be applied to applications and infrastructure. Doing so can help the product management and engineering groups to be more proactive in designing and testing the security of applications and systems and to collaborate more closely with the internal security team. Threat modeling requires both IT and business process knowledge, as well as technical knowledge of how the applications or systems under review work.

- **Security Monitoring and Incident Response**: Centralized security information management systems should be used to provide notification of security vulnerabilities and to monitor systems continuously through automated technologies to identify potential issues. They should be integrated with network and other systems monitoring processes (e.g., security information management, security event management, security information and event management, and security operations centers that use these systems for dedicated 24/7/365 monitoring). Management of periodic, independent third-party security testing should also be included. Many of the security threats and issues in SaaS center around application and data layers, so the types and sophistication of threats and attacks for a SaaS organization require a different approach to security monitoring than traditional infrastructure and perimeter monitoring. The organization may thus need to expand its security monitoring capabilities to include application- and data-level activities. This may also require subject-matter experts in applications security and the unique aspects of maintaining privacy in the cloud. Without this capability and expertise, a company may be unable to detect and prevent security threats and attacks to its customer data and service stability.
- **Third-Party Risk Management**: As SaaS moves into cloud computing for the storage and processing of customer data, there is a higher expectation that the SaaS will effectively manage the security risks with third parties. Lack of a third-party risk management program may result in damage to the provider's reputation, revenue losses, and legal actions should the provider be found not to have performed due diligence on its third-party vendors.

Security Architecture Design:

A security architecture framework should be established with consideration of processes (enterprise authentication and authorization, access control, confidentiality, integrity, non- repudiation, security management, etc.), operational procedures, technology specifications, people and organizational management, and security program compliance and reporting. A security architecture document should be developed that defines security and privacy principles to meet business objectives. Documentation is required for management controls and metrics specific to asset classification and control, physical security, system access controls, network and computer management, application development and maintenance, business continuity, and compliance. A design and implementation program should also be integrated with the formal system development life cycle to include a business case, requirements definition, design, and implementation plans. Technology and design methods should be included, as well as the security processes necessary to provide the following services across all technology layers:

1. Authentication
2. Authorization
3. Availability
4. Confidentiality
5. Integrity
6. Accountability
7. Privacy

The creation of a secure architecture provides the engineers, data center operations personnel, and network operations personnel a common blueprint to design, build, and test the security of the applications and systems. Design reviews of new changes can be better assessed against this architecture to assure that they conform to the principles described in the architecture, allowing for more consistent and effective design reviews.

Data and Applications Security

The ultimate challenge in cloud computing is data-level security, and sensitive data is the domain of the enterprise, not the cloud computing provider. Security will need to move to the data level so that enterprises can be sure their data is protected wherever it goes. For example, with data-level security, the enterprise can specify that this data is not allowed to go outside of the United States. It can also force encryption of certain

types of data, and permit only specified users to access the data. It can provide compliance with the Payment Card Industry Data Security Standard (PCI DSS). True unified end-to-end security in the cloud will likely requires an ecosystem of partners.

Application security is one of the critical success factors for a world-class SaaS company. This is where the security features and requirements are defined and application security test results are reviewed. Application security processes, secure coding guidelines, training, and testing scripts and tools are typically a collaborative effort between the security and the development teams. Although product engineering will likely focus on the application layer, the security design of the application itself, and the infrastructure layers interacting with the application, the security team should provide the security requirements for the product development engineers to implement. This should be a collaborative effort between the security and product development team. External penetration testers are used for application source code reviews, and attack and penetration tests provide an objective review of the security of the application as well as assurance to customers that attack and penetration tests are performed regularly. Fragmented and undefined collaboration on application security can result in lower-quality design, coding efforts, and testing results.

Virtual Machine Security

In the cloud environment, physical servers are consolidated to multiple virtual machine instances on virtualized servers. Not only can data center security teams replicate typical security controls for the data center at large to secure the virtual machines, they can also advise their customers on how to prepare these machines for migration to a cloud environment when appropriate.

Firewalls, intrusion detection and prevention, integrity monitoring, and log inspection can all be deployed as software on virtual machines to increase protection and maintain compliance integrity of servers and applications as virtual resources move from on- premises to public cloud environments. By deploying this traditional line of defense to the virtual machine itself, you can enable critical applications and data to be moved to the cloud securely. To facilitate the centralized management of a server firewall policy, the security software loaded onto a virtual machine should include a bidirectional stateful firewall that enables virtual machine isolation and location awareness, thereby enabling a tightened policy and the flexibility to move the virtual machine from on-premises to cloud resources. Integrity monitoring and log inspection software must be applied at the virtual machine level.

This approach to virtual machine security, which connects the machine back to the mother ship, has some advantages in that the security software can be put into a single software agent that provides for consistent control and management throughout the cloud while integrating seamlessly back into existing security infrastructure investments, providing economies of scale, deployment, and cost savings for both the service provider and the enterprise.

Legal Issues and Aspects

Cloud computing is bringing amazing advantages and benefits companies. But it also brings some challenges. There are several legal issues that must be taken into consideration when moving into the cloud. Let’s see which are the most challenging legal issues around cloud computing and how to overcome them.

Security procedures

The majority of companies which implemented cloud solutions and services do not have security procedures in place. Also, they lack measures to approve or evaluate cloud applications. When adopting the BYOD trend for example, organizations needed these security procedures more than ever. General data security trainings, multiple levels of security, rigorous procedures to use one’s own device and to transfer or copy data are some of the options available to protect data in organizations. The bottom line is that security procedures must be established according to every company’s objectives and work flow.

Third party access issues

Third-party involvement could be a risk. All third parties using a multi-tenant shared cloud are using the same administration interface, so make sure multi-factor authentication and enhanced security is present. Also, look for compliant providers – a business associate agreement (BAA) with third-party vendor who access Protected Health Information (PHI) is necessary to ensure privacy and security requirements. A partnership with a HIPAA solutions provider that signs a BAA is an efficient method to make sure this this goes smoothly and everything is secure. And don't forget to read carefully the terms and conditions before signing up for a cloud based services.

Intellectual Property Rights

Intellectual Property Rights differ from one country to another, so it is not very clear what intellectual property laws will apply in the cloud computing environment. Make sure you are aware of the regulations and rights from the country you store your intellectual work. The provider you choose should know how to protect intellectual property it stores and how to avoid potential infringement pitfalls.

Confidential data theft attacks

Data stored in the cloud might be compromised or breached. Therefore, most cloud computing providers also offer the customer different levels of security protection, which allows for more enhanced security. Encryption might seem to have failed in protecting data from theft attacks, but other methods have been discovered and implemented, including monitoring data access in the cloud to detect abnormal data access patterns. The customer has to understand the cloud provider's disclosure policy and how quickly the breach would be disclosed to them. Most of the U.S. states have security breach disclosure laws requiring the provider to inform the customers when their data has been compromised.

Many of these legal issues and the methods to inform about them or to solve them should be mentioned in the Service Level Agreement. It is essential to understand all the terms of the cloud's provider and to consider the needs and objectives of the enterprise before signing an agreement.

Multi-tenancy issues in cloud computing

Multi-tenancy issues in cloud computing are a growing concern, especially as the industry expands. And big business enterprises have shifted their workload to the cloud. Cloud computing provides different services on the internet. Including giving users access to resources via the internet, such as servers and databases. Cloud computing lets you work remotely with networking and software.

There is no need to be at a specific place to store data. Information or data can be available on the internet. One can work from wherever he wants. Cloud computing brings many benefits for its users or tenants, like flexibility and scalability. Tenants can expand and shrink their resources according to the needs of their workload. Tenants or users do not need to worry about the maintenance of the cloud.

Tenants need to pay for only the services they use. Still, there are some multi-tenancy issues in cloud computing that you must look out for:

Security: This is one of the most challenging and risky issues in multi-tenancy cloud computing. There is always a risk of data loss, data theft, and hacking. The database administrator can grant access to an unauthorized person accidentally. Despite software and cloud computing companies saying that client data is safer than ever on their servers, there are still security risks.

There is a potential for security threats when information is stored using remote servers and accessed via the internet. There is always a risk of hacking with cloud computing. No matter how secure encryption is, someone can always decrypt it with the proper knowledge. A hacker getting access to a multitenant cloud system can gather data from many businesses and use it to his advantage. Businesses need high-level trust when putting data on remote servers and using resources provided by the cloud company to run the software.

The multi-tenancy model has many new security challenges and vulnerabilities. These new security challenges and vulnerabilities require new techniques and solutions. For example, a tenant gaining access to someone else's data and it's returned to the wrong tenant, or a tenant affecting another in terms of resource sharing.

Performance: SaaS applications are at different places, and it affects the response time. SaaS applications usually take longer to respond and are much slower than server applications. This slowness affects the overall performance of the systems and makes them less efficient. In the competitive and growing world of cloud computing, lack of performance pushes the cloud service providers down. It is significant for multi-tenancy cloud service providers to enhance their performance.

Less Powerful: Many cloud services run on web 2.0, with new user interfaces and the latest templates, but they lack many essential features. Without the necessary and adequate features, multi-tenancy cloud computing services can be a nuisance for clients.

Noisy Neighbor Effect: If a tenant uses a lot of the computing resources, other tenants may suffer because of their low computing power. However, this is a rare case and only happens if the cloud architecture and infrastructure are inappropriate.

Interoperability: users remain restricted by their cloud service providers. Users can not go beyond the limitations set by the cloud service providers to optimize their systems. For example, users can not interact with other vendors and service providers and can't even communicate with the local applications.
This prohibits the users from optimizing their system by integrating with other service providers and local applications. Organizations can not even integrate with their existing systems like the on-premise data centers.

Monitoring: constant monitoring is vital for cloud service providers to check if there is an issue in the multi-tenancy cloud system. Multi-tenancy cloud systems require continuous monitoring, as computing resources get shared with many users simultaneously. If any problem arises, it must get solved immediately not to disturb the system's efficiency.

However, monitoring a multi-tenancy cloud system is very difficult as it is tough to find flaws in the system and adjust accordingly.

Chapter -6

##Web Services in Cloud Computing

The Internet is the worldwide connectivity of hundreds of thousands of computers belonging to many different networks.

A web service is a standardized method for propagating messages between client and server applications on the World Wide Web. A web service is a software module that aims to accomplish a specific set of tasks. Web services can be found and implemented over a network in cloud computing.

The web service would be able to provide the functionality to the client that invoked the web service.

A web service is a set of open protocols and standards that allow data exchange between different applications or systems. Web services can be used by software programs written in different programming languages and on different platforms to exchange data through computer networks such as the Internet. In the same way, communication on a computer can be inter-processed.

Any software, application, or cloud technology that uses a standardized Web protocol (**HTTP or HTTPS**) to connect, interoperate, and exchange data messages over the Internet-usually XML (Extensible Markup Language) is considered a Web service. Is.

Web services allow programs developed in different languages to be connected between a client and a server by exchanging data over a web service. A client invokes a web service by submitting an XML request, to which the service responds with an XML response.

- Web services functions
- It is possible to access it via the Internet or intranet network.
- XML messaging protocol that is standardized.
- Operating system or programming language independent.
- Using the XML standard is self-describing.

A simple location approach can be used to detect this.

Web Service Components

XML and HTTP is the most fundamental web service platform. All typical web services use the following components:

1. SOAP (Simple Object Access Protocol)

SOAP stands for "Simple Object Access Protocol". It is a transport-independent messaging protocol. SOAP is built on sending XML data in the form of SOAP messages. A document known as an XML document is attached to each message.

Only the structure of an XML document, not the content, follows a pattern. The great thing about web services and SOAP is that everything is sent through HTTP, the standard web protocol.

Every SOAP document requires a root element known as an element. In an XML document, the root element is the first element.

The "envelope" is divided into two halves. The header comes first, followed by the body. Routing data, or information that directs the XML document to which client it should be sent, is contained in the header. The real message will be in the body.

2. UDDI (Universal Description, Search, and Integration)

UDDI is a standard for specifying, publishing and searching online service providers. It provides a specification that helps in hosting the data through web services. UDDI provides a repository where WSDL files can be hosted so that a client application can search the WSDL file to learn about the various actions provided by the web service. As a result, the client application will have full access to UDDI, which acts as the database for all WSDL files.

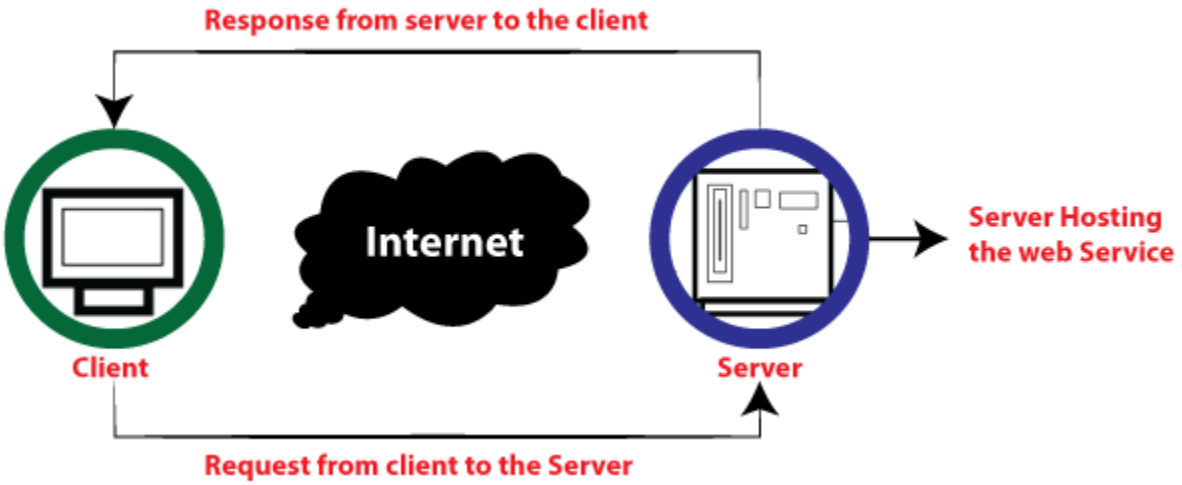
The UDDI Registry will keep the information needed for online services, such as a telephone directory containing the name, address, and phone number of a certain person so that client applications can find where it is.

3. WSDL (Web Services Description Language)

The client implementing the web service must be aware of the location of the web service. If a web service cannot be found, it cannot be used. Second, the client application must understand what the web service does to implement the correct web service. WSDL, or Web Service Description Language, is used to accomplish this. A WSDL file is another XML-based file that describes what a web service does with a client application. The client application will understand where the web service is located and how to access it using the WSDL document.

How does web service work?

The diagram shows a simplified version of how a web service would function. The client will use requests to send a sequence of web service calls to the server hosting the actual web service.



Remote procedure calls are used to perform these requests. The calls to the methods hosted by the respective web service are known as Remote Procedure Calls (RPC). Example: Flipkart provides a web service that displays the prices of items offered on Flipkart.com. The front end or presentation layer can be written in .NET or Java, but the web service can be communicated using a programming language.

The data exchanged between the client and the server, XML, is the most important part of web service design. XML (Extensible Markup Language) is a simple, intermediate language understood by various programming languages. It is the equivalent of HTML.

As a result, when programs communicate with each other, they use XML. It forms a common platform for applications written in different programming languages to communicate with each other.

Web services employ SOAP (Simple Object Access Protocol) to transmit XML data between applications. The data is sent using standard HTTP. A SOAP message is data sent from a web service to an application. An XML document is all that is contained in a SOAP message. The client application that calls the web service can be built in any programming language as the content is written in XML.

Features of Web Service

Web services have the following characteristics:

(a) XML-based: A web service's information representation and record transport layers employ XML. There is no need for networking, operating system, or platform bindings when using XML. At the mid-level, web offering-based applications are highly interactive.

(b) Loosely Coupled: The subscriber of an Internet service provider may not necessarily be directly connected to that service provider. The user interface for a web service provider may change over time without affecting the user's ability to interact with the service provider. A

strongly coupled system means that the decisions of the mentor and the server are inextricably linked, indicating that if one interface changes, the other must be updated.

A loosely connected architecture makes software systems more manageable and easier to integrate between different structures.

(c) Ability to be synchronous or asynchronous: Synchronicity refers to the client's connection to the execution of the function. Asynchronous operations allow the client to initiate a task and continue with other tasks. The client is blocked, and the client must wait for the service to complete its operation before continuing in synchronous invocation.

Asynchronous clients get their results later, but synchronous clients get their effect immediately when the service is complete. The ability to enable loosely connected systems requires asynchronous capabilities.

(d) Coarse Grain: Object-oriented systems, such as Java, make their services available differently. At the corporate level, an operation is too great for a character technique to be useful. Building a Java application from the ground up requires the development of several granular strategies, which are then combined into a coarse grain provider that is consumed by the buyer or service.

Corporations should be coarse-grained, as should the interfaces they expose. Building web services is an easy way to define coarse-grained services that have access to substantial business enterprise logic.

(e) Supports remote procedural calls: Consumers can use XML-based protocols to call procedures, functions, and methods on remote objects that use web services. A web service must support the input and output framework of the remote system.

Enterprise-wide component development Over the years, JavaBeans (EJBs) and .NET components have become more prevalent in architectural and enterprise deployments. Several RPC techniques are used to both allocate and access them.

A web function can support RPC by providing its services, similar to a traditional role, or translating incoming invocations into an EJB or .NET component invocation.

(f) Supports document exchanges: One of the most attractive features of XML for communicating with data and complex entities.

##Aneka in Cloud Computing

Aneka includes an extensible set of APIs associated with programming models like MapReduce.

These APIs support different cloud models like a private, public, hybrid Cloud.

Manjrasoft focuses on creating innovative software technologies to simplify the development and deployment of private or public cloud applications. Our product plays the role of an application platform as a service for multiple cloud computing.

- Multiple Structures:
- Aneka is a software platform for developing cloud computing applications.
- In Aneka, cloud applications are executed.
- Aneka is a pure PaaS solution for cloud computing.
- Aneka is a cloud middleware product.
- Manya can be deployed over a network of computers, a multicore server, a data center, a virtual cloud infrastructure, or a combination thereof.

Multiple containers can be classified into three major categories:

- Textile services
- Foundation Services
- Application Services

1. Textile Services:

Fabric Services defines the lowest level of the software stack that represents multiple containers. They provide access to resource-provisioning subsystems and monitoring features implemented in many.

2. Foundation Services:

Fabric Services are the core services of Manya Cloud and define the infrastructure management features of the system. Foundation services are concerned with the logical management of a distributed system built on top of the infrastructure and provide ancillary services for delivering applications.

3. Application Services:

Application services manage the execution of applications and constitute a layer that varies according to the specific programming model used to develop distributed applications on top of Aneka.

There are mainly two major components in multiple technologies:

The SDK (Software Development Kit) includes the Application Programming Interface (API) and tools needed for the rapid development of applications. The Anka API supports three popular cloud programming models: **Tasks, Threads** and **MapReduce**;

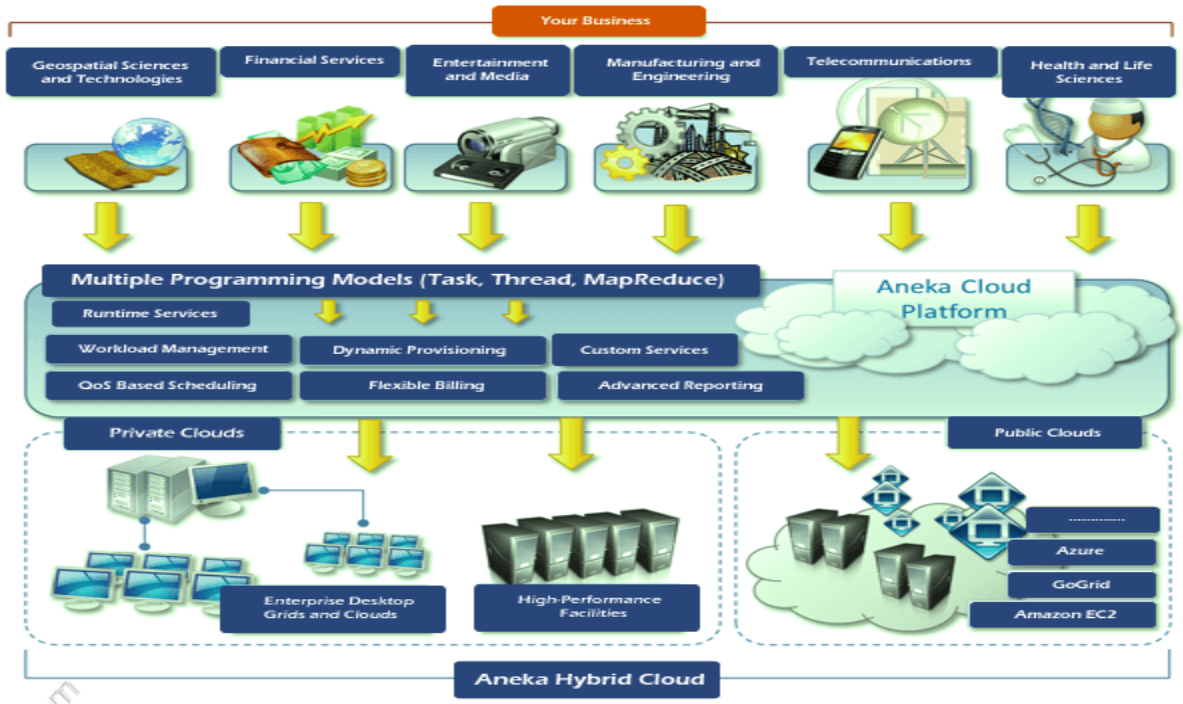
And

A runtime engine and platform for managing the deployment and execution of applications on a private or public cloud.

One of the notable features of Aneka Pass is to support the provision of private cloud resources from desktop, cluster to a virtual data center using **VMware, Citrix Zen Server**, and public cloud resources such as **Windows Azure, Amazon EC2**, and **GoGrid cloud service**.

Aneka's potential as a Platform as a Service has been successfully harnessed by its users and customers in three different areas, including **engineering, life sciences, education,** and **business intelligence.**

Architecture of Aneka



Aneka is a platform and framework for developing distributed applications on the Cloud. It uses desktop PCs on-demand and CPU cycles in addition to a heterogeneous network of servers or datacenters. Aneka provides a rich set of APIs for developers to transparently exploit such resources and express the business logic of applications using preferred programming abstractions.

System administrators can leverage a collection of tools to monitor and control the deployed infrastructure. It can be a public cloud available to anyone via the Internet or a private cloud formed by nodes with restricted access.

A multiplex-based computing cloud is a collection of physical and virtualized resources connected via a network, either the Internet or a private intranet. Each resource hosts an instance of multiple containers that represent the runtime environment where distributed applications are executed. The container provides the basic management features of a single node and takes advantage of all the other functions of its hosting services.

Services are divided into clothing, foundation, and execution services. Foundation services identify the core system of Anka middleware, which provides a set of infrastructure features to enable Anka containers to perform specific and specific tasks. Fabric services interact directly with nodes through the Platform Abstraction Layer (PAL) and perform hardware profiling and dynamic resource provisioning. Execution services deal directly with scheduling and executing applications in the Cloud.

One of the key features of Aneka is its ability to provide a variety of ways to express distributed applications by offering different programming models; Execution services are mostly concerned with providing middleware with the implementation of these models. Additional services such as persistence and security are inverse to the whole stack of services hosted by the container.

At the application level, a set of different components and tools are provided to

- Simplify the development of applications (SDKs),
- Port existing applications to the Cloud, and
- Monitor and manage multiple clouds.

An Aneka-based cloud is formed by interconnected resources that are dynamically modified according to user needs using resource virtualization or additional CPU cycles for desktop machines. A common deployment of Aneka is presented on the side. If the deployment identifies a private cloud, all resources are in-house, for example, within the enterprise.

This deployment is enhanced by connecting publicly available on-demand resources or by interacting with several other public clouds that provide computing resources connected over the Internet.

##Google App Engine

Google App Engine (GAE) is a platform-as-a-service product that provides web app developers and enterprises with access to Google's scalable hosting and tier 1 internet service.

GAE requires that applications be written in Java or Python, store data in Google Bigtable and use the Google query language. Noncompliant applications require modification to use GAE.

GAE provides more infrastructure than other scalable hosting services, such as Amazon Elastic Compute Cloud (EC2). GAE also eliminates some system administration and development tasks to make writing scalable applications easier.

Google provides GAE free up to a certain amount of use for the following resources:

- processor (CPU)
- storage
- application programming interface (API) calls
- concurrent requests

Users exceeding the per-day or per-minute rates can pay for more of these resources.

Amazon Elastic Beanstalk is a rival platform as a service to Google App Engine. It is compatible with different programming languages and runtimes than GAE.

How is GAE used?

GAE is a fully managed, serverless platform that is used to host, build and deploy web applications. Users can create a GAE account, set up a software development kit and write application source code. They can then use GAE to test and deploy the code in the cloud.

One way to use GAE is building scalable mobile application back ends that adapt to workloads as needed. Application testing is another way to use GAE. Users can route traffic to different application versions to A/B test them and see which version performs better under various workloads.

Serverless platforms like Google App Engine are inherently scalable and allow for quick software deployments. See 10 common uses.

What are GAE's key features?

Key features of GAE include the following:

API selection. GAE has several built-in APIs, including the following five:

- **Blobstore** for serving large data objects;
- **GAE Cloud Storage** for storing data objects;
- **Page Speed Service** for automatically speeding up webpage load times;
- **URL Fetch Service** to issue HTTP requests and receive responses for efficiency and scaling; and
- **Memcache** for a fully managed in-memory data store.

Managed infrastructure. Google manages the back-end infrastructure for users. This approach makes GAE a serverless platform and simplifies API management.

Several programming languages. GAE supports a number of languages, including GO, PHP, Java, Python, NodeJS, .NET and Ruby. It also supports custom runtimes.

Support for legacy runtimes. GAE supports legacy runtimes, which are versions of programming languages no longer maintained. Examples include Python 2.7, Java 8 and Go 1.11.

Application diagnostics. GAE lets users record data and run diagnostics on applications to gauge performance.

Security features. GAE enables users to define access policies with the GAE firewall and managed Secure Sockets Layer/Transport Layer Security certificates for free.
Traffic splitting. GAE lets users route requests to different application versions.
Versioning. Applications in Google App Engine function as a set of microservices that refer back to the main source code. Every time code is deployed to a service with the corresponding GAE configuration files, a version of that service is created.

Google App Engine benefits and challenges

GAE extends the benefits of cloud computing to application development, but it also has drawbacks.

Benefits of GAE

- **Ease of setup and use**. GAE is fully managed, so users can write code without considering IT operations and back-end infrastructure. The built-in APIs enable users to build different types of applications. Access to application logs also facilitates debugging and monitoring in production.
- **Pay-per-use pricing**. GAE's billing scheme only charges users daily for the resources they use. Users can monitor their resource usage and bills on a dashboard.
- **Scalability.** Google App Engine automatically scales as workloads fluctuate, adding and removing application instances or application resources as needed.
- **Security**. GAE supports the ability to specify a range of acceptable Internet Protocol (IP) addresses. Users can allowlist specific networks and services and blocklist specific IP addresses.

GAE challenges

- **Lack of control**. Although a managed infrastructure has advantages, if a problem occurs in the back-end infrastructure, the user is dependent on Google to fix it.
- **Performance limits**. CPU-intensive operations are slow and expensive to perform using GAE. This is because one physical server may be serving several separate, unrelated app engine users at once who need to share the CPU.
- **Limited access.** Developers have limited, read-only access to the GAE filesystem.
- **Java limits.** Java apps cannot create new threads and can only use a subset of the Java runtime environment standard edition classes.

Examples of Google App Engine

One example of an application created in GAE is an Android messaging app that stores user log data. The app can store user messages and write event logs to the Firebase Realtime Database and use it to automatically synchronize data across devices.
Java servers in the GAE flexible environment connect to Firebase and receive notifications from it. Together, these components create a back-end streaming service to collect messaging log data.
GAE can be used in many different application contexts. Additional sample application code in GitHub includes the following:

- a Python application that uses Blobstore;
- a program that uses MySQL connections from GAE to Google Cloud Platform SQL; and
- code that shows how to set up unit tests in GAE.

GAE lets users structure applications as microservices. Learn the difference between service-oriented architecture and microservices and the benefits of each.

Amazon Web Services (AWS)

Amazon Web Services (AWS)

is a cloud computing platform which was introduced in 2002. It offers a wide range of cloud services such as Infrastructure as a Service (IaaS)

, Platform as a Service (PaaS)

, and Software as a Service (SaaS)

AWS provides the largest community with millions of active customers as well as thousands of partners globally. Most of the organizations use AWS to expand their business by moving their IT management to the AWS.

Flexibility, security, scalability, and better performance are some important features of AWS.

Microsoft Azure

Microsoft Azure

is also called as **Windows Azure**. It is a worldwide cloud platform which is used for building, deploying, and managing services. It supports multiple programming languages such as **Java**

, Nodejs

, C

, and **C#**

. The advantage of using Microsoft Azure is that it allows us to a wide variety of services without arranging and purchasing additional hardware components.

Microsoft Azure provides several computing services, including servers, storage, databases, software, networking, and analytics over the Internet

.

Google Cloud Platform (GCP)

Google Cloud Platform (GCP)

is introduced by **Google** in 2011. It allows us to use Google's products such as **Google search engine, Gmail, YouTube**, etc. Most of the companies use this platform to easily build, move, and deploy applications on the cloud. It allows us to access these applications using a high-speed internet

connection. The advantage of GCP is that it supports various databases such as **SQL, MYSQL, Oracle, Sam**, and **more**.

Google Cloud Platform (GCP) provides various cloud computing services, including computing, data analytics, data storage, and machine learning.

Difference between AWS, Azure, and Google Cloud Platform (GCP)

Although AWS, Microsoft Azure, and Google cloud platforms offer various high-level features in terms of computing, management, storage, and other services, but there are also some differences between these three.

The below table shows the difference between AWS, Azure, and Google Cloud Platform -

Parameter	AWS	Azure	Google Cloud Platform
App Testing	It uses device farm	It uses DevTest labs	It uses Cloud Test labs.
API Management	Amazon API gateway	Azure API gateway	Cloud endpoints.
Kubernetes Management	EKS	Kubernetes service	Kubernetes engine
Git Repositories	AWS source repositories	Azure source repositories	Cloud source repositories.
Data warehouse	Redshift	SQL warehouse	Big Query
Object Storage	S3	Block Blobs and files	Google cloud storage.
Relational DB	RDS	Relational DBs	Google Cloud SQL
Block Storage	EBS	Page Blobs	Persistent disks
Marketplace	AWS	Azure	G suite
File Storage	EFS	Azure Files	ZFS and Avere
Media Services	Amazon Elastic transcoder	Azure media services	Cloud video intelligence API
Virtual network	VPC	VNet	Subnet
Pricing	Per hour	Per minute	Per minute

Maximum processors in VM	128	128	96
Maximum memory in VM (GiB)	3904	3800	1433
Catching	ElasticCache	RedisCache	CloudCDN
Load Balancing Configuration	Elastic Load Balancing	Load Balancer Application Gateway	Cloud Load Balancing
Global Content Delivery Networks	CloudFront	Content Delivery Network	Cloud Interconnect

Business and consumer application of cloud application – Accounting Application

In terms of applications for businesses in cloud computing, accounting software is a real-time application of cloud computing that helps in real-time day-to-day accounting services. Cloud accounting software such as Sage and Quickbooks comes with the same functionality as desktop accounting, just moving everything to the cloud. With business and consumer applications of cloud applications, it gives business access to its accounts from a mobile device. By keeping your bookkeeping up to date, you can achieve real-time reporting.

Business and consumer application of cloud application

There are many business and consumer applications of cloud applications for small, medium and large businesses. Statistics show that many enterprises have embraced the advantages that cloud computing offers their consumers and to further improve their business goals.

What is business and consumer application of cloud application? This cloud computing has different options to offer various users access to data wherever there is an internet connection. What is business and consumer application of cloud applicaiton? In terms of business and consumer application of cloud application, business owners use different types of cloud services to store their data and provide consumer services and this can be private, public or hybrid. With business and consumer application of cloud application, the cloud gives small businesses the chance to grow quickly, and apart from that, the technology has other advantages such as efficiency, cost-effectiveness, collaboration, and high-speed connectivity.

When it comes to business and consumer applications of cloud applications, employees know that they are able to access customer information via **cloud-based CRM software** easily. With business and consumer applications of cloud applications, consumers and businesses can conveniently access data from their mobile devices wherever they are. With applications for

businesses in clouding computing, there are still businesses that are hesitant when it comes to committing to cloud-computing solutions but they soon discover the many business advantages, some of which are –

- Security
- Cost savings
-
- Mobility
- Increased collaboration
- Quality control
- Loss prevention
- Disaster recovery

With applications for businesses in cloud computing, cost savings is one of the biggest advantages of cloud computing. Once you're on the cloud, you have easy access to your company's data. With applications for business in cloud computing, many companies worry that they will have to pay for a whole lot of features that they don't want, little realizing that most cloud-computing services are pay as you go.

Other massively important applications for business in clouding computing is security. Many organizations naturally have security concerns in the 21st century and they worry about the expense of adopting a cloud-computing solution. A cloud host's job is to carefully monitor security and it is far more efficient than your regular in-house system.

Cloud computing is recognized for its value amongst many different industries. Applications of cloud computing play a big role in our daily lives even though we are scarcely aware of it. Applications for businesses in cloud computing has become all important with the pandemic. It has increased the demand for real-time applications of cloud computing. Applications for businesses in cloud computing have become more important to ensure the ongoing progress of a business.

LOAD BALANCING IN CLOUD COMPUTING

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine. Balancing the load of virtual machines uniformly means that anyone of the available machine is not idle or partially loaded while others are heavily loaded. Load balancing is one of the important factors to heighten the working performance of the cloud service provider. The benefits of distributing the workload includes increased resource utilization ratio which further leads to enhancing the overall performance thereby achieving maximum client satisfaction.

In cloud computing, if users are increasing load will also be increased, the increase in the number of users will lead to poor performance in terms of resource usage, if the cloud provider is not configured with any good mechanism for load balancing and also the capacity of cloud servers would not be utilized properly. This will confiscate or seize the performance of heavy loaded node. If some good load balancing technique is implemented, it will equally divide the load (here term equally defines low load on heavy loaded node and more load on node with less load now) and thereby we can maximize resource utilization. One of the crucial issue of cloud computing is to divide the workload dynamically.

2.1 Goals of Load Balancing

- Substantial improvement in performance
- Stability maintenance of the system
- Increase flexibility of the system so as to adapt to the modifications.
- Build a fault tolerant system by creating backups.

2.2 Classification of Load Balancing Algorithm

a) Sender Initiated: In this sender initiates the process; the client sends request until a receiver is assigned to him to receive his workload
b) Receiver Initiated: The receiver initiates the process; the receiver sends a request to acknowledge a sender who is ready to share the workload
c) Symmetric: It is a combination of both sender and receiver initiated type of load balancing algorithm.

Based on the current state of the system they are classified as:

1. Static Load Balancing

In the static load balancing algorithm the decision of shifting the load does not depend on the current state of the system. It requires knowledge about the applications and resources of the system. The performance of the virtual machines is determined at the time of job arrival. The master processor assigns the workload to other slave processors according to their performance. The assigned work is thus performed by the slave processors and the result is returned to the master processor.
Static load balancing algorithms are not preemptive and therefore each machine has at least one task assigned for itself. Its aims in minimizing the execution time of the task and limit communication overhead and delays. This algorithm has a drawback that the task is assigned to the processors or machines only after it is created and that task cannot be shifted during its execution to any other machine for balancing the load.The four different types of Static load balancing techniques are Round Robin algorithm, Central Manager algorithm, Threshold algorithm and randomized algorithm.

2. Dynamic Load Balancing

In this type of load balancing algorithms the current state of the system is used to make any decision for load balancing, thus the shifting of the load is depend on the current state of the system. It allows for processes to move from an over utilized machine to an under utilized machine dynamically for faster execution.
This means that it allows for process preemption which is not supported in Static load balancing approach. An important advantage of this approach is that its decision for balancing the load is based on the current state of the system which helps in improving the overall performance of the system by migrating the load dynamically.

2.3 Traditional Computing V/S Cloud Computing Environment

There are many similarities as well as differences between traditional scheduling algorithms and the scheduling of VM resources in cloud computing environment.
First of all the major difference between cloud computing environment and traditional computing environment is the target of scheduling. In traditional computing environment, it mainly schedules process or task so the granularity and the transferred data is small; whereas in cloud computing environment, the scheduling target is VM resources so the granularity is large and the transferred data is large as well.
Secondly, in cloud computing environment, compared with the deployment time of VMs, the time of scheduling algorithm can almost be neglected.

2.4. Need of Load Balancing

We can balance the load of a machine by dynamically shifting the workload local to the machine to remote nodes or machines which are less utilized. This maximizes the user satisfaction, minimizing response time, increasing resource utilization, reducing the number of job rejections and raising the performance ratio of the system.

Load balancing is also needed for achieving Green computing in clouds [5]. The factors responsible for it are:

1. Limited Energy Consumption: Load balancing can reduce the amount of energy consumption by avoiding over hearting of nodes or virtual machines due to excessive workload.
2. Reducing Carbon Emission: Energy consumption and carbon emission are the two sides of the same coin. Both are directly proportional to each other. Load balancing helps in reducing energy consumption which will automatically reduce carbon emission and thus achieve Green Computing .

III. LOAD BALANCING ALGORITHMS

The paper describes about three load balancing algorithms which are Round robin algorithm, equally spread current execution load and Throttled Load balancing.

 Round Robin: Round robin use the time slicing mechanism. The name of the algorithm suggests that it works in the round manner where each node is allotted with a time slice and has to wait for their turn. The time is divided and interval is allotted to each node. Each node is allotted with a time slice in which they have to perform their task. The complicity of this algorithm is less compared to the other two algorithms. An open source simulation performed the algorithm software know as cloud analyst, this algorithm is the default algorithm used in the simulation. This algorithm simply allots the job in round robin fashion which doesn't consider the load on different machines.

 Equally spread current execution load: This algorithm requires a load balancer which monitors the jobs which are asked for execution. The task of load balancer is to queue up the jobs and hand over them to different virtual machines. The balancer looks over the queue frequently for new jobs and then allots them to the list of free virtual server. The balance also maintains the list of task allotted to virtual servers, which helps them to identify that which virtual machines are free and need to be allotted with new jobs. The experimental work for this algorithm is performed using the cloud analyst simulation. The name suggests about this algorithm that it work on equally spreading the execution load on different virtual machine.

 Throttled Load balancing: The Throttled algorithm work by finding the appropriate virtual machine for assigning a particular job. The job manager is having a list of all virtual machines, using this indexed list, it allot the desire job to the appropriate machine. If the job is well suited for a particular machine than that job is, assign to the appropriate machine. If no virtual machines are available to accept jobs then the job manager waits for the client request and takes the job in queue for fast processing.
 ARA(Adaptive Resource Allocation):In ARA algorithm for adaptive resource allocation in cloud systems, which attempts to counteract the deleterious effect of burrstones by allowing some randomness in the decision making process and thus improve overall system performance and availability [1]. The problem with this strategy is that it only considers the Poisson arrival streams as well as the exponentially distributed service time and the fixed number of choice.

The following figure shows the diagrammatical representation of the algorithm used for load balancing in cloud computing environment [8], [9].The figure also shows the three algorithms which are studded in this paper using the cloud analyst simulation tool, this tool is based on cloud sim , the cloud sim provides a GUI inter face which helps to perform the experimental work.

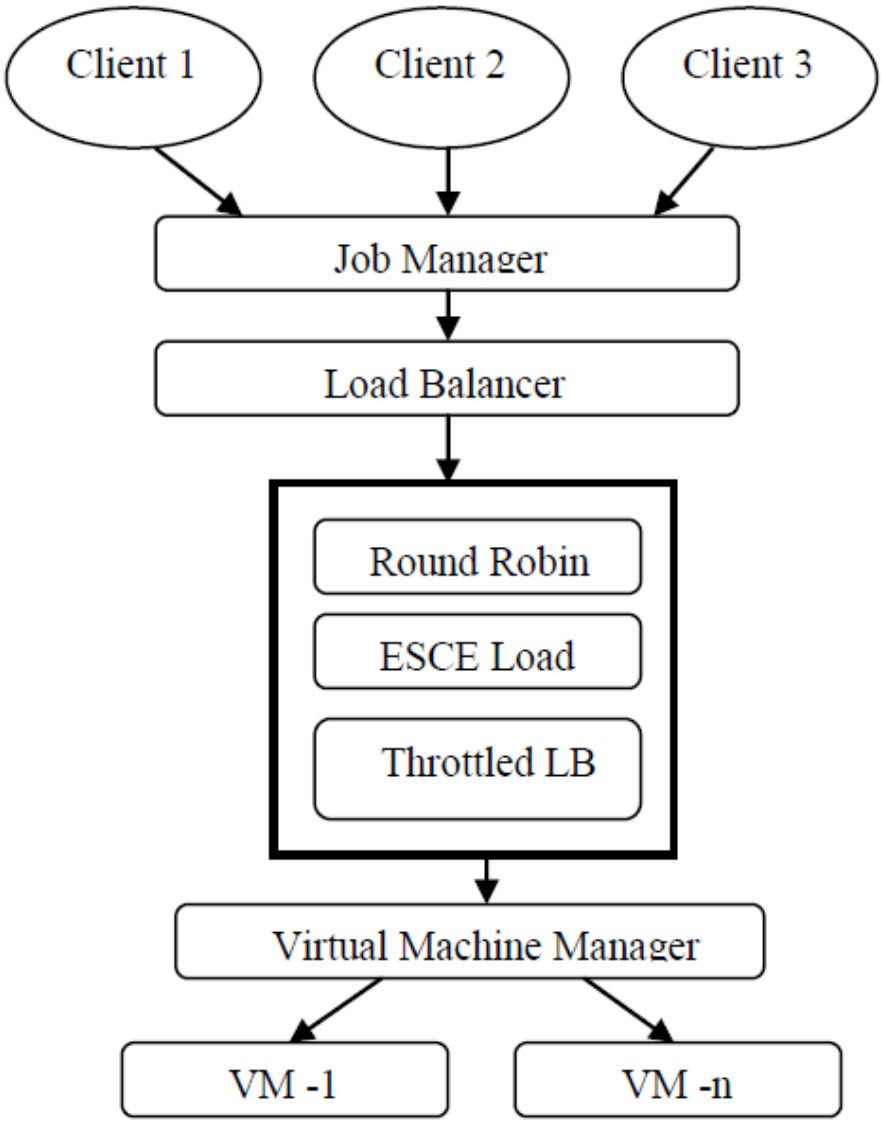


Fig.1 Load Balancing Algorithms Execution

IV. DYNAMIC LOAD BALANCING POLICIES AND STRATEGIES

1. Location Policy: The policy used by a processor or machine for sharing the task transferred by an over loaded machine is termed as Location policy.
2. Transfer Policy: The policy used for selecting a task or process from a local machine for transfer to a remote machine is termed as Transfer policy.
3. Selection Policy: The policy used for identifying the processors or machines that take part in load balancing is termed as Selection Policy.
4. Information Policy: The policy that is accountable for gathering all the information on which the decision of load balancing is based id referred as Information policy.
5. Load estimation Policy: The policy which is used for deciding the method for approximating the total work load of a processor or machine is termed as Load estimation policy.

6. Process Transfer Policy: The policy which is used for deciding the execution of a task that is it is to be done locally or remotely is termed as Process Transfer policy.
7. Priority Assignment Policy: The policy that is used to assign priority for execution of both local and remote processes and tasks is termed as Priority Assignment Policy.
8. Migration Limiting Policy: The policy that is used to set a limit on the maximum number of times a task can migrate from one machine to another machine.