[Unit 1: Introduction]

Cloud Computing (CSC-458)

Jagdish Bhatta

Central Department of Computer Science & Information Technology

Tribhuvan University

Introduction:

What is Cloud?

The term *cloud* has been used historically as a metaphor for the Internet and has become a familiar cliché. This usage was originally derived from its common depiction in network diagrams as an outline of a cloud, used to represent the transport of data across carrier backbones (which owned the cloud) to an endpoint location on the other side of the cloud.

The cloud itself is a set of hardware, networks, storage, services, and interfaces that enable the delivery of computing as a service. Cloud services include the delivery of software, infrastructure, and storage over the internet (either as separate components or a complete platform) based on user demand.

The world of the cloud has lots of participants:

✓ The **end user** doesn't really have to know anything about the underlying technology. In small businesses, for example, the cloud provider becomes the de facto data center. In larger organizations, the IT organization oversees the inner workings of both internal resources and external cloud resources.

✓ **Business management** needs to take responsibility for overall governance of data or services living in a cloud. Cloud service providers must provide a predictable and guaranteed service level and security to all their constituents.

✓ The **cloud service provider** is responsible for IT assets and maintenance.

Overall, the cloud embodies the following four basic characteristics:

✓ **Elasticity and the ability to scale up and down**
✓ **Self-service provisioning and automatic deprovisioning**
✓ **Application programming interfaces (APIs)**
✓ **Billing and metering of service usage in a pay-as-you-go model**

Elasticity and scalability:

The service provider can't anticipate how customers will use the service. One customer might use the service three times a year during peak selling seasons, whereas another might use it as a primary development platform for all of its applications. Therefore, the service needs to be available all the time (7 days a week, 24 hours a day) and it has to be designed to scale upward for high periods of demand and downward for lighter ones. ***Scalability*** **also means that an application can scale when additional users are added and** when the application requirements change**. This ability to scale is achieved by providing *elasticity*.** Think about the rubber band and its properties. If you're holding together a dozen pens with a rubber band, you probably have to fold it in half. However, if you're trying to keep 100 pens together, you will have to stretch that rubber band. Why can a single rubber band accomplish both tasks? Simply, it is elastic and so is the cloud. : **Elasticity refers to the ability to flex to meet the needs and preferences of users on a near real-time basis, in response to supply and demand triggers.** In the cloud context, elasticity refers to the ability of a service or an infrastructure to adjust to meet fluctuating service demands by automatically provisioning or de-provisioning resources or by moving the service to be executed on another part of the system.

Self-service provisioning:

Customers can easily get cloud services without going through a lengthy process. The customer simply requests an amount of computing, storage, software, process, or other resources from the service provider. Contrast this on-demand response with the process at a typical data center. When a department is about to implement a new application, it has to submit a request to the data center for additional computing hardware, software, services, or process resources. The data center gets similar requests from departments across the company and must sort through all requests and evaluate the availability of existing resources versus the need to purchase new hardware. After new hardware is purchased, the data center staff has to configure the data center for the new application. These internal procurement processes can take a long time, depending on company policies. Of course, nothing is as simple as it might appear. **While the on-demand provisioning capabilities of cloud services eliminate many time delays**, an organization still needs to do its homework. These services aren't free; needs and requirements must be determined before capability is automatically provisioned.

Application programming interfaces (APIs):

Cloud services need to have standardized APIs. These interfaces provide the instructions on how two application or data sources can communicate with each other. A standardized interface lets the customer more easily link a cloud service, such as a customer relationship management system with a financial accounts management system, without having to resort to custom programming.

Billing and metering of services:

A cloud environment needs a built-in service that bills customers. And, of course, to calculate that bill, usage has to be *metered* (tracked). Even free cloud services (such as Google's Gmail or Zoho's Internet-based office applications) are metered.

Cloud Computing:

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

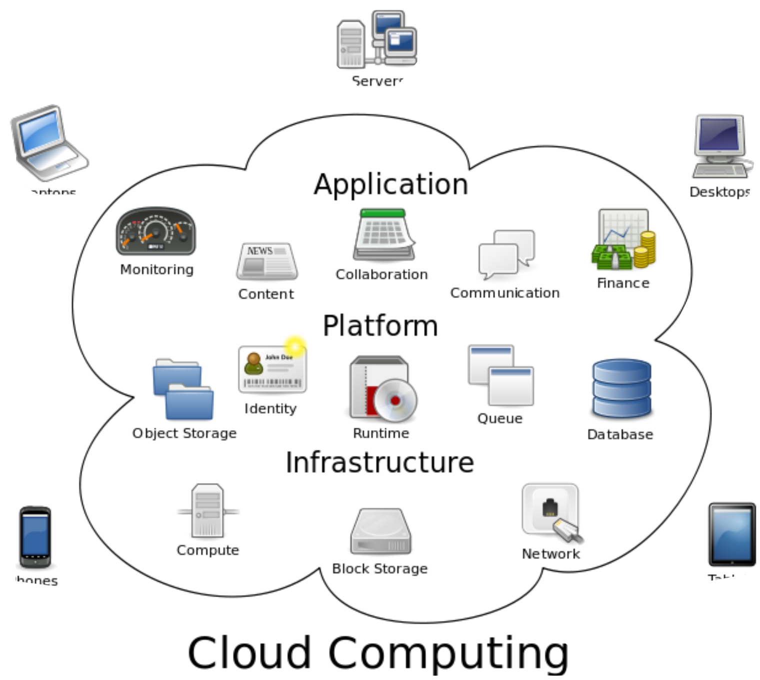
Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams.

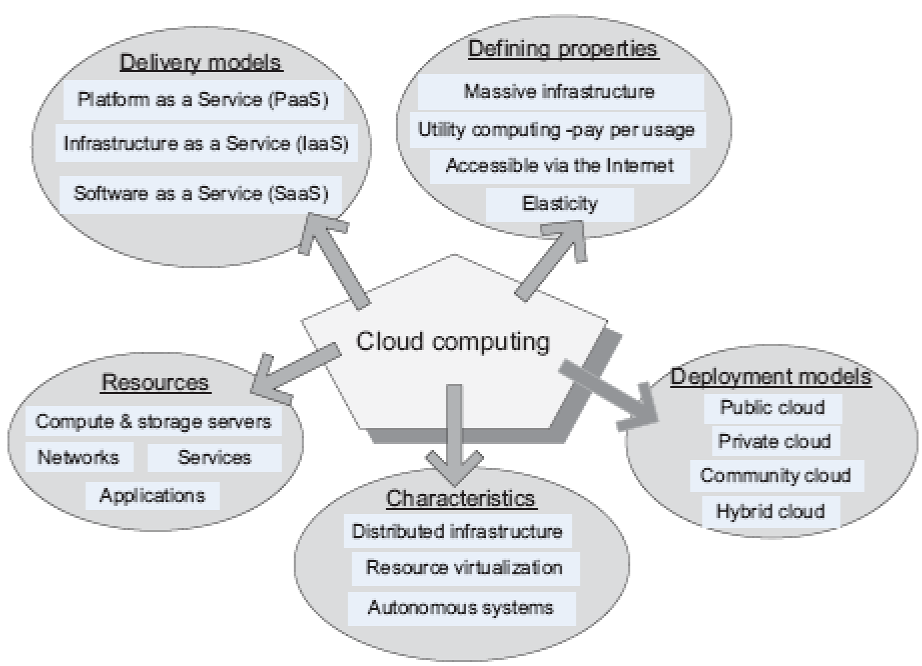
A cloud service has three distinct characteristics that differentiate it from traditional hosting.

- **It is sold on demand,** typically by the minute or the hour;
- **It is elastic** -- a user can have as much or as little of a service as they want at any given time; and
- **The service is fully managed by the provider** (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

A more tempered view of cloud computing considers it the delivery of computational resources from a location other than the one from which you are computing.

The service consumer no longer has to be at a PC, use an application from the PC, or purchase a specific version that's configured for smartphones, PDAs, and other devices. The consumer does not own the infrastructure, software, or platform in the cloud. He/She has lower upfront costs, capital expenses, and operating expenses. He/She does not care about how servers and networks are maintained in the cloud. The consumer can access multiple servers anywhere on the globe without knowing which ones and where they are located.


Servers
Desktops
Application
Monitoring
Content
Collaboration
Communication
Finance
Platform
Object Storage
Identity
Runtime
Queue
Database
Infrastructure
Compute
Block Storage
Network
Cloud Computing


Delivery models
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
Software as a Service (SaaS)
Defining properties
Massive infrastructure
Utility computing -pay per usage
Accessible via the Internet
Elasticity
Cloud computing
Resources
Compute & storage servers
Networks
Services
Applications
Deployment models
Public cloud
Private cloud
Community cloud
Hybrid cloud
Characteristics
Distributed infrastructure
Resource virtualization
Autonomous systems

<u>Emergence of Cloud Computing:</u>

The origin of the term *cloud computing* is obscure, but it appears to derive from the practice of using drawings of stylized clouds to denote networks in diagrams of computing and communications systems. The word *cloud* is used as a metaphor for the Internet, based on the standardized use of a cloud-like shape to denote a network on telephony schematics and later to depict the Internet in computer network diagrams as an abstraction of the underlying infrastructure it represents. The cloud symbol was used to represent the Internet as early as 1994.

In the 1990s, telecommunications companies, who previously offered primarily dedicated point-to-point data circuits, began offering virtual private network (VPN) services with comparable quality of service but at a much lower cost. By switching traffic to balance utilization as they saw fit, they were able to utilize their overall network bandwidth more effectively. The cloud symbol was used to denote the demarcation point between that which was the responsibility of the provider and that which was the responsibility of the users. Cloud computing extends this boundary to cover servers as well as the network infrastructure.

The underlying concept of cloud computing dates back to the 1950s; when large-scale mainframe became available in academia and corporations, accessible via thin clients /terminal computers. Because it was costly to buy a mainframe, it became important to find ways to get the greatest return on the investment in them, allowing multiple users to share both the physical access to the computer from multiple terminals as well as to share the CPU time, eliminating periods of inactivity, which became known in the industry as time-sharing.

As in the earliest stages, the term "cloud" was used to represent the computing space between the provider and the end user. **In 1997, Professor Ramnath Chellapa of Emory University and the University of South California defined cloud computing as the new "computing paradigm where the boundaries of computing will be determined by**

economic rationale rather than technical limits alone." This has become the basis of what we refer to today when we discuss the concept of cloud computing.

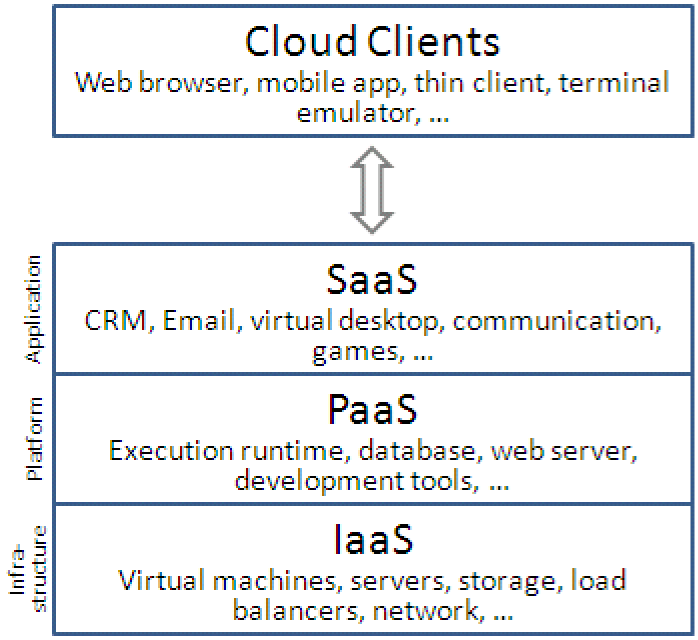
Some people think cloud computing is the next big thing in the world of IT. Others believe it is just another variation of the utility computing model that has been repackaged in this decade as something new and cool.

One of the first milestones for cloud computing was the arrival of Salesforce.com in 1999, which pioneered the concept of delivering enterprise applications via a simple website. The services firm paved the way for both specialist and mainstream software firms to deliver applications over the internet.

The next development was Amazon Web Services in 2002, which provided a suite of cloud-based services including storage, computation and even human intelligence through the Amazon Mechanical Turk. Then in 2006, Amazon launched its Elastic Compute cloud (EC2) as a commercial web service that allows small companies and individuals to rent computers on which to run their own computer applications.

Cloud Based Service Models:

The services given by the service provider to the customers or users through cloud computing technology are said cloud services. Service Provider's server gives both the hardware and software necessary and thus easy in management for both user and the cloud service provider. **Cloud computing providers offer their services according to three fundamental models: Infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) where IaaS is the most basic and each higher model abstracts from the details of the lower models.**



__IaaS__: **In this most basic cloud service model, providers offer computers, as physical or more often as virtual machines, and other resources.** The virtual machines are run as guests by a hypervisor, such as Xen or KVM. Management of pools of hypervisors by the cloud operational support system leads to the ability to scale to support a large number of virtual machines. **Other resources in IaaS clouds include images in a virtual machine image library, raw (block) and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles.** Amies, Alex; Sluiman, Harm; Tong IaaS cloud providers supply these resources on demand from their large pools installed in data centers. For wide area connectivity, the Internet can be used or —in carrier clouds -- dedicated virtual private networks can be configured

To deploy their applications, cloud users then install operating system images on the machines as well as their application software. In this model, it is the cloud user who is responsible for patching and maintaining the operating systems and application software. **Cloud providers typically bill IaaS services on a utility computing basis, that is, cost will reflect the amount of resources allocated and consumed. STaaS -** STorage As A Service. This service comes under IaaS, which manages all the storage services in cloud computing. There are many security issues in this service. They are Data Integrity, Confidentiality, Reliability, etc.

IaaS refers not to a machine that does all the work, but simply to a facility given to businesses that offers users the leverage of extra storage space in servers and data centers.

Examples of IaaS include: Amazon CloudFormation (and underlying services such as Amazon EC2), Rackspace Cloud, Terremark, Windows Azure Virtual Machines, Google Compute Engine, and Joyent.

<u>PaaS</u>**: In the PaaS model, cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server.** Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. **With some PaaS offers, the underlying computer and storage resources scale automatically to match application demand such that cloud user does not have to allocate resources manually.**

Examples of PaaS include: Amazon Elastic Beanstalk, Cloud Foundry, Heroku, Force.com, EngineYard, Mendix, Google App Engine, Windows Azure Compute and OrangeScape.

<u>SaaS</u>**: In this model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. The cloud users do not manage the cloud infrastructure and platform on which the application is running. This eliminates the need to install and run the application on the cloud user's own computers simplifying maintenance and support.** What makes a cloud application different from other applications is its scalability. This can be achieved by cloning tasks onto multiple virtual machines at run-time to meet the changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user who sees only a single access point. **To accommodate a large number of**

cloud users, cloud applications can be multitenant, that is, any machine serves more than one cloud user organization. It is common to refer to special types of cloud based application software with a similar naming convention: desktop as a service, business process as a service, test environment as a service, communication as a service.

The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point.

Examples of SaaS include: Google Apps, Microsoft Office 365, and Onlive.

Grid Computing or Cloud Computing?

Grid computing is often confused with cloud computing. Grid computing is a form of distributed computing that implements a *virtual supercomputer* made up of a cluster of networked or Internetworked computers acting in unison to perform very large tasks. Many cloud computing deployments today are powered by grid computing implementations and are billed like utilities, but cloud computing can and should be seen as an evolved next step away from the grid utility model. There is an ever-growing list of providers that have successfully used cloud architectures with little or no centralized infrastructure or billing systems, such as the peer-to-peer network BitTorrent and the volunteer computing initiative SETI@home.

Service commerce platforms are yet another variation of SaaS and MSPs. This type of cloud computing service provides a centralized service hub that users interact with. Currently, the most often used application of this platform is found in financial trading environments or systems that allow users to order things such as travel or personal services from a common platform (e.g., Expedia.com or Hotels.com), which then coordinates pricing and service delivery within the specifications set by the user.

Cloud computing evolves from grid computing and provides on-demand resource provisioning. Grid computing may or may not be in the cloud depending on what type of users are using it. If the users are systems administrators and integrators, they care how things are maintained in the cloud. They upgrade, install, and virtualize servers and applications. If the users are consumers, they do not care how things are run in the system.

The **difference between grid computing and cloud computing** is hard to grasp because they are not always mutually exclusive. In fact, they are both used to economize computing by maximizing existing resources. However, **the difference between the two lies in the way the tasks are computed in each respective environment**. In a computational grid, one large job is divided into many small portions and executed on multiple machines. This characteristic is fundamental to a grid; not so in a cloud. The computing in cloud is intended to allow the user to avail of various services without investing in the underlying architecture. Cloud services include the delivery of software, infrastructure, and storage over the Internet (either as separate components or a complete platform) based on user demand.

Cloud computing and grid computing are scalable. Scalability is accomplished through load balancing of application instances running separately on a variety of operating systems and connected through Web services. CPU and network bandwidth is allocated and de-allocated on demand. The system's storage capacity goes up and down depending on the number of users, instances, and the amount of data transferred at a given time.

Both computing types involve **multitenancy and multitask**, meaning that many customers can perform different tasks, accessing a single or multiple application instances. Sharing resources among a large pool of users assists in reducing infrastructure costs and peak load capacity. Cloud and grid computing provide service-level agreements (SLAs) for guaranteed uptime availability of, say, 99 percent. If the service slides below the level of the guaranteed uptime service, the consumer will get service credit for receiving data late. The Amazon S3 provides a Web services interface for the storage and retrieval of data in the cloud. Setting a maximum limits the number of objects you can store in S3. You can store an object as small as 1 byte and as large as 5 GB or even several terabytes. S3 uses the concept of buckets as containers for each storage location of your objects. The data is stored securely using the same data storage infrastructure that Amazon uses for its e-commerce Web sites.

While the storage computing in the grid is well suited for data-intensive storage, it is not economically suited for storing objects as small as 1 byte. In a data grid, the amounts of distributed data must be large for maximum benefit. A computational grid focuses on computationally intensive operations. Amazon Web Services in cloud computing offers two types of instances: standard and high-CPU.

(Refer the research article, provided in the class, entitled "Cloud Computing and Grid Computing 360-Degree Compared", *by the authors; Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu, published in 2008).*

Virtualization:

Virtualization is the key to cloud computing, since it is the enabling technology allowing the creation of an intelligent abstraction layer which hides the complexity of underlying hardware or software.

Server virtualization enables different operating systems to share the same hardware and make it easy to move operating systems between different hardware, all while the applications are running.

Storage virtualization does the same thing for data. Storage virtualization creates the abstraction layer between the applications running on the servers, and the storage they use to store the data.

Virtualizing the storage and incorporating the intelligence for provisioning and protection at the virtualization layer enables companies to use any storage they want, and not be locked into any individual vendor. Storage virtualization makes storage a commodity. All this makes for some interesting ways for companies to reduce their costs.

Any discussion of cloud computing typically begins with virtualization. **Virtualization is critical to cloud computing because it simplifies the delivery of services by providing a platform for optimizing complex IT resources in a scalable manner, which is what makes cloud computing so cost effective.**

Virtualization can be applied very broadly to just about everything you can imagine including memory, networks, storage, hardware, operating systems, and applications. Virtualization has three characteristics that make it ideal for cloud computing:

- **Partitioning:** In virtualization, you can use partitioning to support many applications and operating systems in a single physical system.
- **Isolation:** Because each virtual machine is isolated, each machine is protected from crashes and viruses in the other machines. What makes virtualization so important for the cloud is that it decouples the software from the hardware.
- **Encapsulation:** Encapsulation can protect each application so that it doesn't interfere with other applications. Using encapsulation, a virtual machine can be represented (and even stored) as a single file, making it easy to identify and present to other applications.

To understand how virtualization helps with cloud computing, you must understand its many forms. In essence, in all cases, a resource actually emulates or imitates another resource. Here are some examples:

- **Virtual memory:** Disks have a lot more space than memory. PCs can use virtual memory to borrow extra memory from the hard disk. Although virtual disks are slower than real memory, if managed right, the substitution works surprisingly well.
- **Software:** There is virtualization software available that can emulate an entire computer, which means 1 computer can perform as though it were actually 20 computers. Using this kind of software you might be able to move from a data center with thousands of servers to one that supports as few as a couple of hundred.

To manage the various aspects of virtualization in cloud computing most companies use ***hypervisors, an operating system*** *that act as traffic cop managing the various virtualization tasks in the cloud to ensure that they make the things happen in an orderly* *manner.* Because in cloud computing you need to support many different operating environments, the hypervisor becomes an ideal delivery mechanism by allowing you to show the same application on lots of different systems. Because hypervisors can load multiple operating systems, they are a very practical way of getting things virtualized quickly and efficiently.

Cloud Computing Deployment Models (Types):

Public: The infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services. It is also referred to as 'external' Cloud that describes the conventional meaning of Cloud computing: scalable, dynamically provisioned, often virtualized resources available over the internet from an off-site third party provider, which divides up resources and bills its customers on a 'utility' basis.

Public cloud applications, storage, and other resources are made available to the general public by a service provider. These services are free or offered on a pay-per-use model. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure and offer access only via Internet (direct connectivity is not offered).

Private: The infrastructure is operated solely for an organization; it may be managed by the organization or a third party and may exist on or off the premises of the organization. It is also referred to as 'corporate' or 'internal' Cloud, term used to denote a proprietary computing architecture providing hosted services on private networks.

A private cloud could provide the computing resources needed for a large organization, e.g., a research institution, a university, or a corporation. There are some arguments that a private cloud does not support utility computing when the user pays as it consumes resources.

Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and it will require the organization to reevaluate decisions about existing resources. When it is done right, it can have a positive impact on a business, but every one of the steps in the project raises security issues that must be addressed in order to avoid serious vulnerabilities

Community: **Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.),** whether managed internally or by a third-party and hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized

Hybrid: **Here, the infrastructure is a composition of two or more clouds (private, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds). Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and bursts into a public cloud when the demand for computing capacity spikes. The advantage of such a hybrid cloud deployment is that an organization only pays for extra compute resources when they are needed.** Experts recommend cloud bursting for high performance, non-critical applications that handle non-sensitive information. An application can be deployed locally and then burst to the cloud to meet peak demands, or the application can be moved to the public cloud to free up local resources for business-critical applications. Cloud bursting works best for applications that don’t depend on a complex application delivery infrastructure or integration with other applications, components and systems internal to the data center.

By utilizing "hybrid cloud" architecture, companies and individuals are able to obtain degrees of fault tolerance combined with locally immediate usability without dependency on internet connectivity. Hybrid cloud architecture requires both on-premises resources and off-site (remote) server-based cloud infrastructure.

Hybrid clouds lack the flexibility, security and certainty of in-house applications. Hybrid cloud provides the flexibility of in house applications with the fault tolerance and scalability of cloud based services.

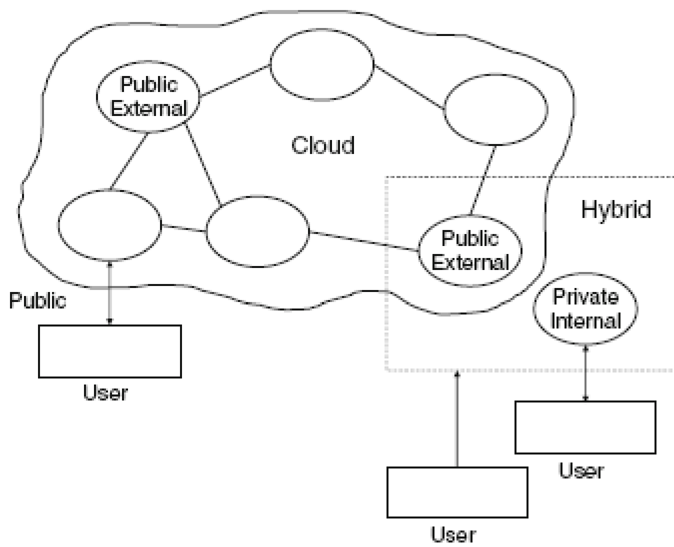


Fig: Types of Cloud Deployment Models

<u>Ethical issues in cloud computing:</u>

Cloud computing is based on a paradigm shift with profound implications on computing ethics. **The main elements of this shift are:**

(i) the control is relinquished to third party services;

(ii) the data is stored on multiple sites administered by several organizations; and

(iii) multiple services interoperate across the network.

Unauthorized access, data corruption, infrastructure failure, or unavailability are some of the risks related to relinquishing the control to third party services; moreover, it is difficult to identify the source of the problem and the entity causing it. Systems can span the boundaries of multiple organizations and cross the security borders, a process called *deperimeterisation*. As a result of de-perimeterisation "not only the border of the organizations IT infrastructure blurs, also the border of the accountability becomes less clear".

The complex structure of cloud services can make it difficult to determine who is responsible in case something undesirable happens. In a complex chain of events or systems, many entities contribute to an action with undesirable consequences, some of them have the opportunity to prevent these consequences, and therefore no one can be held responsible, the so-called "problem of many hands."

Ubiquitous and unlimited data sharing and storage among organizations test the selfdetermination of information, the right or ability of individuals to exercise personal control over the collection, use and disclosure of their personal data by others; this tests the confidence and trust in todays evolving information society. Identity fraud and theft are made possible by the unauthorized access to personal data in circulation and by new forms of dissemination through social networks and they could also pose a danger to cloud computing.

The question of what can be done proactively about ethics of cloud computing does not have easy answers as many undesirable phenomena in cloud computing will only appear in time. But the need for rules and regulations for the governance of cloud computing are obvious. The term *governance* means the manner in which something is governed or regulated, the method of management, the system of regulations. Explicit attention to ethics must be paid by governmental organizations providing research funding; private companies are less constraint by ethics oversight and governance arrangements are more conducive to profit generation.

Accountability is a necessary ingredient of cloud computing; adequate information about how data is handled within the cloud and about allocation of responsibility are key elements to enforcing ethics rules in cloud computing. Recorded evidence allows us to assign responsibility; but there can be tension between privacy and accountability and it is important to establish what is being recorded, and who has access to the records.

Unwanted dependency on a cloud service provider, the so-called *vendor lock-in*, is a serious concern and the current standardization efforts at NIST attempt to address

this problem. Another concern for the users is a future with only a handful of companies which dominate the market and dictate prices and policies

<u>Benefits of using cloud models:</u>

Because customers generally do not own the infrastructure used in cloud computing environments, they can forgo capital expenditure and consume resources as a service by just paying for what they use. Many cloud computing offerings have adopted the utility computing and billing model described above, while others bill on a subscription basis. **By sharing computing power among multiple users, utilization rates are generally greatly improved, because cloud computing servers are not sitting dormant for lack of use.** This factor alone can reduce infrastructure costs significantly and accelerate the speed of applications development.

A beneficial side effect of using this model is that computer capacity increases dramatically, since customers do not have to engineer their applications for peak times, when processing loads are greatest. Adoption of the cloud computing model has also been enabled because of the greater availability of increased high-speed bandwidth. With greater enablement, though, there are other issues one must consider, especially legal ones.

The following are some of the possible benefits for those who offer cloud computing-based services and applications:

- **Cost Savings** — Companies can reduce their capital expenditures and use operational expenditures for increasing their computing capabilities. This is a lower barrier to entry and also requires fewer in-house IT resources to provide system support.

- **Scalability/Flexibility** — Companies can start with a small deployment and grow to a large deployment fairly rapidly, and then scale back if necessary. Also, the flexibility of cloud computing allows companies to use extra resources at peak times, enabling them to satisfy consumer demands.

- **Reliability** — Services using multiple redundant sites can support business continuity and disaster recovery.

- **Maintenance** — Cloud service providers do the system maintenance, and access is through APIs that do not require application installations onto PCs, thus further reducing maintenance requirements.

- **Mobile Accessible** — Mobile workers have increased productivity due to systems accessible in an infrastructure available from anywhere.

Characteristics of Cloud Computing:

Cloud computing has a variety of characteristics, with the main ones being:

- **Shared Infrastructure** — Uses a virtualized software model, enabling the sharing of physical services, storage, and networking capabilities. The cloud infrastructure, regardless of deployment model, seeks to make the most of the available infrastructure across a number of users.

- **Dynamic Provisioning / on demand self service** — Allows for the provision of services based on current demand requirements. This is done automatically using software automation, enabling the expansion and contraction of service capability, as needed. This dynamic scaling needs to be done while maintaining high levels of reliability and security.

- **Broad Network Access** — Needs to be accessed across the internet from a broad range of devices such as PCs, laptops, and mobile devices, using standards-based APIs (for example, ones based on HTTP). Deployments of services in the cloud include everything from using business applications to the latest application on the newest smartphones.

- **Multi-Tenant Capable -** The resources (e.g., network, storage and compute power) can be shared among multiple enterprise clients, thereby lowering overall expense. Resource virtualization is used to enforce isolation and aid in security.

- **Rapid Elasticity** - The consumer should have the ability to rapidly (often automatically) increase or decrease the computing resources needed to carry out their work.

- **Managed Metering / Measured Service -** Uses metering for managing and optimizing the service and to provide reporting and billing information. In this way, consumers are billed for services according to how much they have actually used during the billing period. In short, cloud computing allows for the sharing and scalable deployment of services, as needed, from almost any location, and for which the customer can be billed based on actual usage.

Evolution of cloud computing:

Cloud computing can be seen as an innovation in different ways. From a technological perspective it is an advancement of computing, applying virtualization concepts to utilize hardware more efficiently. Yet a different point of view is to look at cloud computing from an IT deployment perspective. In this sense cloud computing has the potential to revolutionize the way, how computing resources and applications are provided, breaking up traditional value chains and making room for new business models. In the following section we are going to describe the emergence of cloud computing from both perspectives.

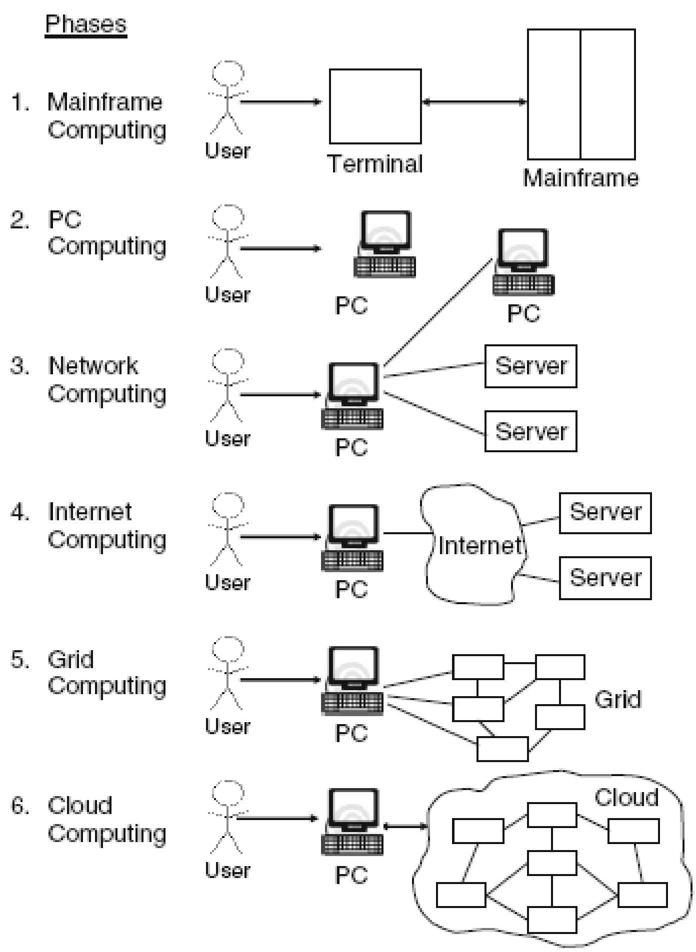


Fig: Evolution of Computing Paradigms from mainframe to cloud computing

Challenges for cloud computing

The development of efficient cloud applications inherits the challenges posed by the natural imbalance between computing, I/O, and communication bandwidths of physical systems; these challenges are greatly amplified due the scale of the system, its distributed nature, and by the fact that virtually all applications are data-intensive. Though cloud computing infrastructures attempt to automatically distribute and balance the load, the application developer is still left with the responsibility to place the data close to the processing site and to identify optimal storage for the data. One of the main advantages of cloud computing, the shared infrastructure, could also have a negative impact as perfect performance isolation is nearly impossible to reach in a real system, especially when the system is heavily loaded. The performance of virtual machines fluctuates based on the load, the infrastructure services, the environment including the other users. Reliability is also a major concern; node failures are to be expected whenever a large numbers of nodes cooperate for the computations. Choosing an optimal instance, in terms of performance isolation, reliability, and security, from those offered by the cloud infrastructure is another critical factor to be considered. Of course, cost considerations play also a role in the choice of the instance type. Many applications consist of multiple stages; in turn, each stage may involve multiple instances running in parallel on the systems of the cloud and communicating among them. Thus, efficiency, consistency, and communication scalability of communication are major concerns for an application developer. Indeed, due to shared networks and unknown topology, cloud infrastructures exhibit inter-node latency and bandwidth fluctuations which affect the application performance.

The following are some of the notable challenges associated with cloud computing, and although some of these may cause a slowdown when delivering more services in the cloud, most also can provide opportunities, if resolved with due care and attention in the planning stages.

- **Security and Privacy** — Perhaps two of the more “hot button” issues surrounding cloud computing relate to storing and securing data, and monitoring the use of the cloud by the service providers. These issues are generally attributed to slowing the deployment of cloud services. These challenges can be addressed, for example, by storing the information internal to the organization, but allowing it to be used in the cloud. For this to occur, though, the security mechanisms between organization and the cloud need to be robust and a Hybrid cloud could support such a deployment.

- **Lack of Standards** — Clouds have documented interfaces; however, no standards are associated with these, and thus it is unlikely that most clouds will be interoperable. The Open Grid Forum is developing an Open Cloud Computing Interface to resolve this issue and the Open Cloud Consortium is working on cloud computing standards and practices. The findings of these groups will need to mature, but it is not known whether they will address the needs of the people deploying the services and the specific interfaces these services need. However, keeping up to date on the latest standards as they evolve will allow them to be leveraged, if applicable.

- **Continuously Evolving** — User requirements are continuously evolving, as are the requirements for interfaces, networking, and storage. This means that a "cloud," especially a public one, does not remain static and is also continuously evolving.

- **Compliance Concerns** — The Sarbanes-Oxley Act (SOX) in the US and Data Protection directives in the EU are just two among many compliance issues affecting cloud computing, based on the type of data and application for which the cloud is being used. The EU has a legislative backing for data protection across all member states, but in the US data protection is different and can vary from state to state. As with security and privacy mentioned previously, these typically result in Hybrid cloud deployment with one cloud storing the data internal to the organization.

Distributed Computing in Grid and Cloud :

Distributed computing is a field of computer science that studies distributed systems. **A distributed system consists of multiple autonomous computers that communicate through a computer network. The computers interact with each other in order to achieve a common goal.**

Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.

The word *distributed* in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing. While there is no single definition of a distributed system, the following defining properties are commonly used:

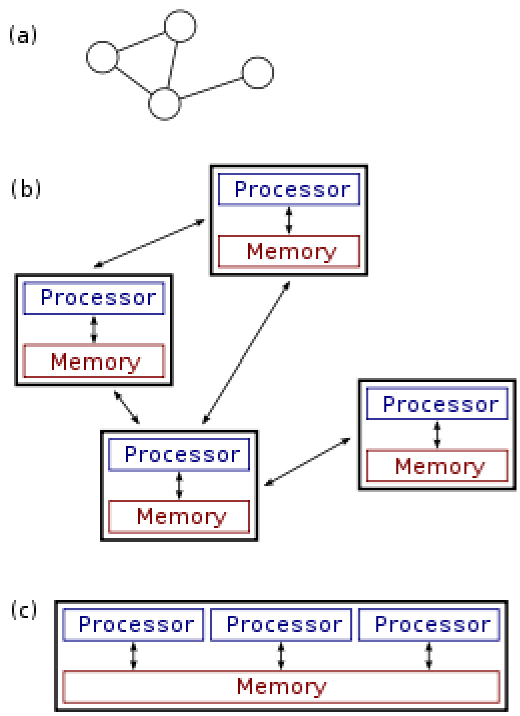
- There are several autonomous computational entities, each of which has its own local memory.
- The entities communicate with each other by message passing

Parallel Vs. Distributed Computing

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

- In parallel computing, all processors may have access to a shared memory to exchange information between processors.
- In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

The figure below illustrates the difference between distributed and parallel systems. Figure (a) is a schematic view of a typical distributed system; as usual, the system is represented as a network topology in which each node is a computer and each line connecting the nodes is a communication link. Figure (b) shows the same distributed system in more detail: each computer has its own local memory, and information can be exchanged only by passing messages from one node to another by using the available communication links. Figure (c) shows a parallel system in which each processor has a direct access to a shared memory.



The last decade, the term 'Grid' has been a key topic in the field of high performance/distributed computing. The Grid has emerged as a new field of distributed computing, focusing on secure sharing of computational and storage resources among dynamic sets of people and organizations who own these resources. This sharing of resources can give people not only computational capabilities and data storage capabilities that cannot be provided by a single supercomputing center, but it also allows them to share data in a transparent way.

Grid Computing can be defined as applying resources from many computers in a network to a single problem, usually one that requires a large number of processing cycles or access to large amounts of data.

At its core, Grid Computing enables devices-regardless of their operating characteristics-to be virtually shared, managed and accessed across an enterprise, industry or workgroup. This virtualization of resources places all of the necessary access, data and processing power at the fingertips of those who need to rapidly solve complex business problems, conduct compute-intensive research and data analysis, and operate in real-time.

Distributed computing was one of the first real instances of cloud computing. Long before Google or Amazon, there was SETI@Home. Proposed in 1995 and launched in 1999, this program uses the spare capacity of internet connected machines to search for extraterrestrial intelligence. This is sort of the cloud in reverse.

A more recent example would be software like Hadoop. Written in Java, Hadoop is a scalable, efficient, distributed software platform designed to process enormous amounts of data. Hadoop can scale to thousands of computers across many clusters.

Distributed computing is nothing more than utilizing many networked computers to partition (split it into many smaller pieces) a question or problem and allow the network to solve the issue piecemeal.

Another instance of distributed computing, for storage instead of processing power, is bittorrent. A torrent is a file that is split into many pieces and stored on many computers around the internet. When a local machine wants to access that file, the small pieces are retrieved and rebuilt.

As the cloud computing buzzword has evolved, distributed computing has fallen out of that particular category of software. Even though distributed computing might take advantage of the internet, it doesn't follow the other tenants of cloud computing, mainly the automatic and instant scalability of resources.

That's not to say that a distributed system couldn't be built to be a cloud environment. Bittorrent, or any P2P system, comes very close to a cloud storage. It would require some additional protections like file ownership and privacy across all nodes but it could probably be done. Privacy like that is not quite what P2P is all about though.

The Cloud Computing paradigm originates mainly from research on distributed computing and virtualization, as it is based on principles, techniques and technologies developed in these areas.