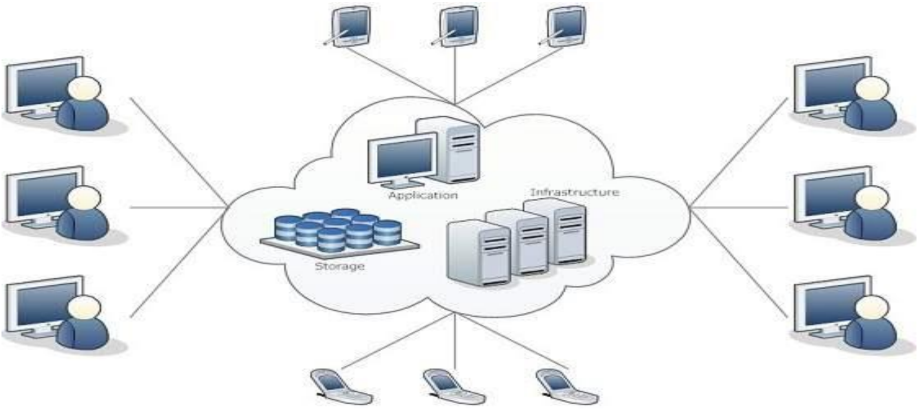
Cloud Computing

The term **Cloud** refers to a **Network** or **Internet.** In other words, we can say that Cloud is something, which is present at remote location. Cloud can provide services over public and private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM) execute on cloud.

Cloud Computing refers to **manipulating, configuring,** and **accessing** the hardware and software resources remotely. It offers online data storage, infrastructure, and application.



Cloud computing offers **platform independency,** as the software is not required to be installed locally on the PC. Hence, the Cloud Computing is making our business applications **mobile** and **collaborative.**

Basic Concepts

There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. Following are the working models for cloud computing:

- Deployment Models
- Service Models

Deployment Models

Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid, and Community.

Public Cloud

The **public cloud** allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness.

Private Cloud

The **private cloud** allows systems and services to be accessible within an organization. It is more secured because of its private nature.

Community Cloud

The **community cloud** allows systems and services to be accessible by a group of organizations.

Hybrid Cloud

The **hybrid cloud** is a mixture of public and private cloud, in which the critical activities are performed using private cloud while the non-critical activities are performed using public cloud.

Service Models

Cloud computing is based on service models. These are categorized into three basic service models which are -

- Infrastructure-as–a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

The Infrastructure-as-a-Service (IaaS) is the most basic level of service. Each of the service models inherit the security and management mechanism from the underlying model.

Infrastructure-as-a-Service (IaaS)

IaaS provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.

Platform-as-a-Service (PaaS)

PaaS provides the runtime environment for applications, development and deployment tools, etc.

Software-as-a-Service (SaaS)

SaaS model allows to use software applications as a service to end-users.

Grid Computing

At its most basic level, grid computing is a computer network in which each computer's resources are shared with every other computer in the system. Processing power, memory and data storage are all community resources that authorized users can tap into and leverage for specific tasks. A grid computing system can be as simple as a collection of similar computers running on the same operating system or as complex as inter-networked systems comprised of every computer platform you can think of.The grid computing concept isn't a new one. It's a special kind of distributed computing. In distributed computing, different computers within the same network share one or more resources. In the ideal grid computing system, every resource is shared, turning a computer network into a powerful supercomputer. With the right user interface, accessing a grid computing system would look no different than accessing a local machine's resources. Every authorized computer would have access to enormous processing power and storage capacity.Though the concept isn't new, it's also not yet perfected. Computer scientists, programmers and engineers are still working on creating, establishing and implementing standards and protocols. Right now, many existing grid computer systems rely on proprietary software and tools. Once people agree upon a reliable set of standards and protocols, it will be easier and more efficient for organizations to adopt the grid computing model.

Grid computing systems work on the principle of pooled resources. A grid computing system uses the concept: share the load across multiple computers to complete tasks more efficiently and quickly. Normally, a computer can only operate within the limitations of its own resources. There's an upper limit to how fast it can complete an operation or how much information it can store. Most computers are upgradeable, which means it's possible to add more power or capacity to a single computer, but that's still just an incremental increase in performance.Grid computing systems link computer resources together in a way that lets someone use one computer to access and leverage the collected power of all the computers in the system. To the individual user, it's as if the user's computer has transformed into a supercomputer.

Grid Computing Lexicon

Some of the terms related to grid computing:

- **Cluster**: A group of networked computers sharing the same set of resources.
- **Extensible Markup Language** (**XML**): A computer language that describes other data and is readable by computers. **Control nodes** (a node is any device connected to a network that can transmit, receive and reroute data) rely on XML languages like the **Web Services Description Language** (**WSDL**). The information in these languages tells the control node how to handle data and applications.
- **Hubs**: A point within a network where various devices connect with one another.
- **Integrated Development Environment** (**IDE**): The tools and facilities computer programmers need to create applications for a platform. The term for an application testing ground is **sandbox**.
- **Interoperability**: The ability for software to operate within completely different environments. For example, a computer network might include both PCs and Macintosh computers. Without interoperable software, these computers wouldn't be able to work together because of their different operating systems and architecture.
- **Open standards**: A technique of creating publically available standards. Unlike proprietary standards, which can belong exclusively to a single entity, anyone can adopt and use an open standard. Applications based on the same open standards are easier to integrate than ones built on different proprietary standards.

- **Parallel processing**: Using multiple CPUs to solve a single computational problem. This is closely related to shared computing, which leverages untapped resources on a network to achieve a task.
- **Platform**: The foundation upon which developers can create applications. A platform can be an operating system, a computer's architecture, a computer language or even a Web site.
- **Server farm**: A cluster of servers used to perform tasks too complex for a single server.
- **Server virtualization**: A technique in which a software application divides a single physical server into multiple exclusive server platforms (the virtual servers). Each virtual server can run its own operating system independently of the other virtual servers. The operating systems don't have to be the same system -- in other words, a single machine could have a virtual server acting as a Linux server and another one running a Windows platform. It works because most of the time, servers aren't running anywhere close to full capacity. Grid computing systems need lots of servers to handle various tasks and virtual servers help cut down on hardware costs.
- **Service**: In grid computing, a service is any software system that allows computers to interact with one another over a network.
- **Simple Object Access Protocol (SOAP):** A set of rules for exchanging messages written in XML across a network. Microsoft is responsible for developing the protocol.
- **State:** In the IT world, a state is any kind of persistent data. It's information that continues to exist in some form even after being used in an application. For example, when you select books to go into an Amazon.com shopping cart, the information is **stateful** -- Amazon keeps track of your selection as you browse other areas of the Web site. Stateful services make it possible to create applications that have multiple steps but rely on the same core data.
- **Transience:** The ability to activate or deactivate a service across a network without affecting other operations.

Working of Grid Computing

Several companies and organizations are working together to create a standardized set of rules called **protocols** to make it easier to set up grid computing environments. It's possible to create a grid computing system right now and several already exist. But what's missing is an agreed-upon approach. That means that two different grid computing systems may not be compatible with one another, because each is working with a unique set of protocols and tools.

In general, a grid computing system requires:

- **At least one computer, usually a server, which handles all the administrative duties for the system.** Many people refer to this kind of computer as a control node. Other application and Web servers (both physical and virtual) provide specific services to the system.
- **A network of computers running special grid computing network software.** These computers act both as a point of interface for the user and as the resources the system will tap into for different applications. Grid computing systems can either include several computers of the same make running on the same operating system (called a homogeneous system) or a hodgepodge of different computers running on every operating system imaginable (a heterogeneous system). The network can be anything from a hardwired system where every computer connects to the system with physical wires to an open system where computers connect with each other over the Internet.
- **A collection of computer software called middleware.** The purpose of middleware is to allow different computers to run a process or application across the entire network of machines. Middleware is the workhorse of the grid computing system. Without it, communication across the system would be impossible. Like software in general, there's no single format for middleware.

  If middleware is the workhorse of the grid computing system, the control node is the dispatcher. The control node must prioritize and schedule tasks across the network. It's the control node's job to determine what resources each task will be able to access. The control node must also monitor the system to make sure that it doesn't become overloaded. It's also important that each user connected to the network doesn't experience a drop in his or her computer's performance. A grid computing system should tap into unused computer resources without impacting everything else.

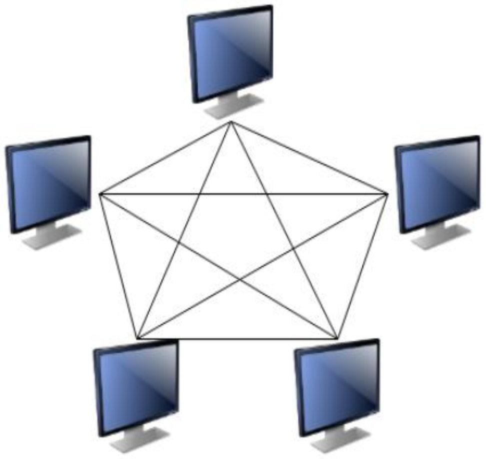
The potential for grid computing applications is limitless, providing everyone agrees on standardized protocols and tools. That's because without a standard format, third-party developers -- independent programmers who want to create applications on the grid computing platform -- often lack the ability to create applications that work on different systems. While it's possible to make different versions of the same application for different systems, it's time consuming and many developers don't want to do the same work twice. A standardized set of protocols means that developers could concentrate on one format while creating applications.

Peer-to-peer (P2P) Computing

Peer-to-peer (**P2P**) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.

Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model in which the consumption and supply of resources is divided. Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual community thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers, yet that are beneficial to all the peers.

The peer to peer computing architecture contains nodes that are equal participants in data sharing. All the tasks are equally divided between all the nodes. The nodes interact with each other as required as share resources.



Characteristics of Peer to Peer Computing

The different characteristics of peer to peer networks are as follows:

- Peer to peer networks are usually formed by groups of a dozen or less computers. These computers all store their data using individual security but also share data with all the other nodes.
- The nodes in peer to peer networks both use resources and provide resources. So, if the nodes increase, then the resource sharing capacity of the peer to peer network increases. This is different than client server networks where the server gets overwhelmed if the nodes increase.
- Since nodes in peer to peer networks act as both clients and servers, it is difficult to provide adequate security for the nodes. This can lead to denial of service attacks.

- Most modern operating systems such as Windows and Mac OS contain software to implement peer to peer networks.

Advantages of Peer to Peer Computing

Some advantages of peer to peer computing are as follows:

- Each computer in the peer to peer network manages itself. So, the network is quite easy to set up and maintain.
- In the client server network, the server handles all the requests of the clients. This provision is not required in peer to peer computing and the cost of the server is saved.
- It is easy to scale the peer to peer network and add more nodes. This only increases the data sharing capacity of the system.
- None of the nodes in the peer to peer network are dependent on the others for their functioning.

Disadvantages of Peer to Peer Computing

Some disadvantages of peer to peer computing are as follows:

- It is difficult to backup the data as it is stored in different computer systems and there is no central server.
- It is difficult to provide overall security in the peer to peer network as each system is independent and contains its own data.

Autonomic Computing

Autonomic computing is a computer's ability to manage itself automatically through adaptive technologies that further computing capabilities and cut down on the time required by computer professionals to resolve system difficulties and other maintenance such as software updates. The move toward autonomic computing is driven by a desire for cost reduction and the need to lift the obstacles presented by computer system complexities to allow for more advanced computing technology.

The autonomic computing initiative (ACI), which was developed by IBM, demonstrates and advocates networking **computer systems that do not involve a lot of human intervention other than defining input rules.** The ACI is derived from the autonomic nervous system of the human body.

IBM has defined the four areas of automatic computing:

- Self-Configuration.
- Self-Healing (error correction).
- Self-Optimization (automatic resource control for optimal functioning).
- Self-Protection (identification and protection from attacks in a proactive manner).
- AC was **designed to mimic the human body's nervous system**-in that the autonomic nervous system acts and reacts to stimuli independent of the individual's

conscious input-an autonomic computing environment functions with a high level of **artificial intelligence** while remaining invisible to the users. Just as the human body acts and responds without the individual controlling functions (e.g., internal temperature rises and falls, breathing rate fluctuates, glands secrete hormones in response to stimulus), the autonomic computing environment operates organically in response to the input it collects.

IBM has set forth eight conditions that define an autonomic system:

1. The system must know itself in terms of what resources it has access to, what its capabilities and limitations are and how and why it is connected to other systems.
2. The system must be able to automatically configure and reconfigure itself depending on the changing computing environment.
3. The system must be able to optimize its performance to ensure the most efficient computing process.
4. The system must be able to work around encountered problems by either repairing itself or routing functions away from the trouble.
5. The system must detect, identify and protect itself against various types of attacks to maintain overall system security and integrity.
6. The system must be able to adapt to its environment as it changes, interacting with neighboring systems and establishing communication protocols.
7. The system must rely on open standards and cannot exist in a proprietary environment.
8. The system must anticipate the demand on its resources while keeping transparent to users.

Autonomic computing is one of the building blocks of ***pervasive computing***, an anticipated future computing model in which tiny – even invisible – computers will be all around us, communicating through increasingly interconnected networks leading to the concept of The Internet of Everything (IoE). Many industry leaders are researching various components of autonomic computing.

BENEFITS

The main benefit of autonomic computing is **reduced TCO (Total Cost of Ownership)**. Breakdowns will be less frequent, thereby drastically reducing maintenance costs. Fewer personnel will be required to manage the systems. The most immediate benefit of autonomic computing will be reduced deployment and maintenance cost, time and increased stability of IT systems through automation, Another benefit of this technology is that **it provides server consolidation to maximize system availability**, and minimizes cost and human effort to manage large server farms.