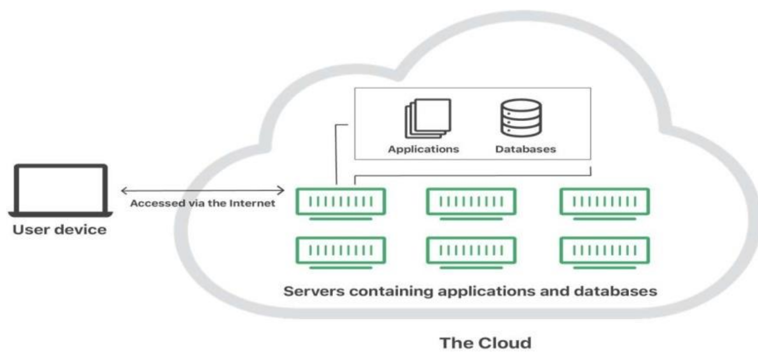
UNIT – I

Cloud Computing Fundamentals: Definition of Cloud computing, Roots of Cloud Computing, Layers and Types of Clouds, Desired Features of a Cloud, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers. Computing Paradigms: High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing.

Introduction to Cloud Computing:

What is the cloud: "The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers. Cloud servers are in data centres all over the world. By using cloud computing, users and companies do not have to manage physical servers themselves or run software applications on their own machines.



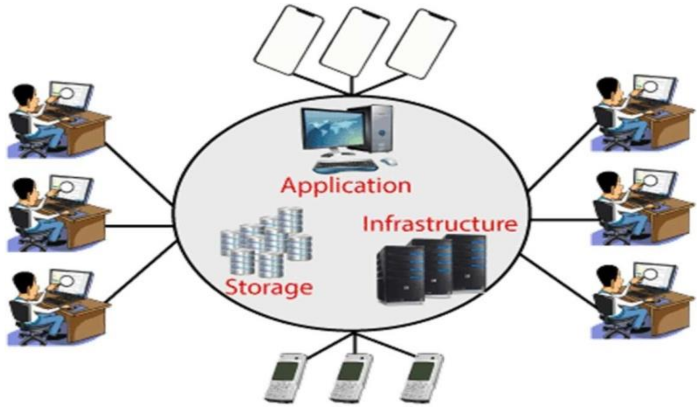
The cloud enables users to access the same files and applications from almost any device, because the computing and storage takes place on servers in a data centre, instead of locallyon the user device. Therefore, a user can log into their Instagram account on a new phone after their old phone breaks and still find their old account in place, with all their photos, videos, and conversation history. It works the same way with cloud email providers likeGmail or Microsoft Office 365, and with cloud storage providers like Dropbox or Google Drive.

For businesses, switching to cloud computing removes some IT costs and overhead: for instance, they no longer need to update and maintain their own servers, as the cloud vendor they are using will do that. This especially makes an impact for small businesses that may not have been able to afford their own internal infrastructure but can outsource their infrastructure needs affordably via the cloud. The cloud can also make it easier for companies to operate internationally, because employees and customers can access the same files and applications from any location.

Definition of Cloud Computing:

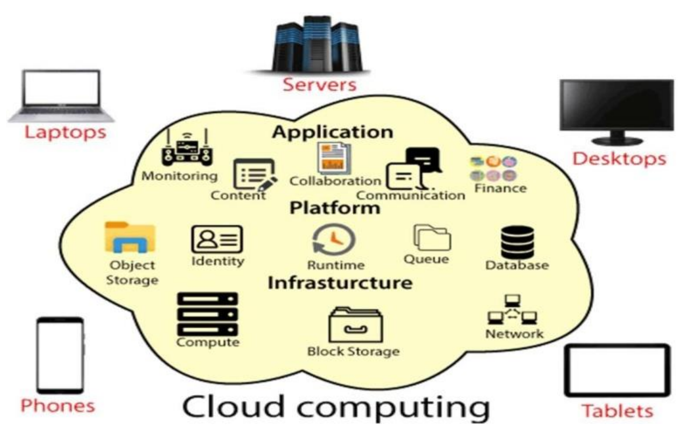
The term "Cloud Computing" refers to services provided by the cloud that is responsible for delivering of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud (Internet).

Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically



Cloud Computing provides an alternative to the on-premises data center. With an on- premises data center, we must 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 doingall the set-up, we become responsible for maintaining it through its entire lifecycle.

However, if we choose Cloud Computing, a cloud vendor is responsible for the hardware purchase and maintenance. They also provide a wide variety of software and platform as a service. We can take any required services on rent. The cloud computing services are charged based on usage.



The cloud environment provides an easily accessible online portal that makes handy for the user to manage the compute, storage, network, and application resources. Some of the cloud service providers are in the following figure.

Advantages of cloud computing:

1. **Cost:** It reduces the huge capital costs of buying hardware and software.
2. **Speed:** Resources can be accessed in minutes, typically within a few clicks.
3. **Scalability:** We can increase or decrease the requirement of resources according to the business requirements.
4. **Productivity:** While using cloud computing, we put less operational effort. We do not need to apply patching, as well as no need to maintain hardware and software. So, in this way, the IT team can be more productive and focus on achieving business goals.
5. **Reliability:** Backup and recovery of data are less expensive and extremely fast for business continuity.
6. **Security:** Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Cloud computing shares characteristics with:

1. **Client–server model**—*Client–server computing* refers broadly to any distributed application that distinguishes between service providers (servers) and service requestors (clients).
2. **Grid computing**—A form of distributed and parallel computing, whereby a 'super and virtual computer' is composed of a cluster of networked, loosely coupled computers acting in concert to perform very large tasks.
3. **Fog computing—**Distributed computing paradigm that provides data, compute, storage and application services closer to the client or near-user edge devices, such asnetwork routers. Furthermore, fog computing handles data at the network level, on smart devices and on the end-user client-side (e.g., mobile devices), instead of sending data to a remote location for processing.
4. **Mainframe computer—**Powerful computers used mainly by large organizations for critical applications, typically bulk data processing such as census; industry and consumer statistics; police and secret intelligence services; enterprise resource planning; and financial transaction processing.
5. **Utility computing—**The packaging of computing resources, such as computation and storage, as a metered service similar to a traditional public utility, such as electricity.

6. **Peer-to-peer—**A distributed architecture without the need for central coordination. Participants are both suppliers and consumers of resources (in contrast to the traditional client-server model).
7. **Green computing—**Study and practice of environmentally sustainable computing or IT.
8. **Cloud sandbox—**A live, isolated computer environment in which a program, code or file can run without affecting the application in which it runs.

Characteristics of Cloud Computing

1. Agility for organizations
2. Cost reductions, Centralization of infrastructure in locations with lower costs.
3. Device and location independence, which means no maintenance, required.
4. Pay-per-use means utilization and efficiency improvements for systems that are often only 10–20% utilized.
5. Performances are being monitored by IT experts i.e., from the service provider end.
6. Productivity increases which results in multiple users who can work on the same data simultaneously.
7. Time may be saved as information does not need to be re-entered when fields are matched
8. Availability improves with the use of multiple redundant sites
9. Scalability and elasticity via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis in near real-time without users having to engineer for peak loads.
10. Self-service interface.
11. Resources thatare abstracted or virtualized.
12. Security can improve due to centralization of data

The National Institute of Standards and Technology's definition of cloud computing identifies "five essential characteristics":

1. On-demand self-service.
2. Broad network access.
3. Resource pooling.
4. Rapid elasticity.
5. Measured service.

ROOTS OF CLOUD COMPUTING

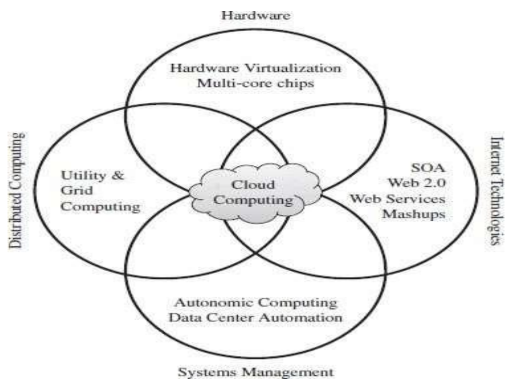
We can track the roots of clouds computing by observing the advancement of several technologies, especially in hardware (virtualization, multi-core chips), Internet technologies (Web services, service-oriented architectures, Web 2.0), distributed computing (clusters, grids), and systems management (autonomic computing, data center automation). Figure 1.1 shows the convergence of technology fields that significantly advanced and contributed to the advent of cloud computing.

The emergence of cloud computing itself is closely linked to the maturity of such technologies. We present a closer look at the technologies that form the base of cloud computing, with the aim of providing a clearer picture of the cloud ecosystem.

1. From Mainframes to Clouds

We are currently experiencing a switch in the IT world, from in-house generated computing power into utility-supplied computing resources delivered over the Internet as Web services. This trend is like what occurred about a century ago when factories, which used to generate their own electric power, realized that it is was cheaper just plugging their machines into the newly formed electric power grid.

Computing delivered as a utility can be defined as “on demand delivery of infrastructure, applications, and business processes in a security-rich, shared, scalable, and based computer environment over the Internet for a fee”



2. SOA, Web Services, Web 2.0, and Mashups

The emergence of Web services (WS) open standards has significantly con- tributed to advances in the domain of software integration. Web services can glue together applications running on different messaging product plat- forms, enabling information from one application to be made available to others, and enabling internal applications to be made available over the Internet.

WS standards have been created on top of existing ubiquitous technologies such as HTTP and XML, thus providing a common mechanism for delivering services, making them ideal for implementing a service-oriented architecture (SOA). The purpose of a SOA is to address requirements of loosely coupled, standards-based, and protocol-independent distributed computing. In a SOA, software resources are packaged as “services,” which are well-defined, self- contained modules that provide standard business functionality and are independent of the state or context of other services. Services are described ina standard definition language and have a published interface

Many service providers, such as Amazon, delicious, Facebook, and Google, make their service APIs publicly accessible using standard protocols such as

SOAP and REST. Consequently, one can put an idea of a fully functional Web application into practice just by gluing pieces with few lines of code.

1. Grid Computing

Grid computing enables aggregation of distributed resources and transparently access to them. Most production grids such as TeraGrid and EGEE seek to share compute and storage resources distributed across different administrative domains, with their focus being speeding up a broad range of scientific applications, such as climate modeling, drug design, and protein analysis.

A key aspect of the grid vision realization has been building standard Web services-based protocols that allow distributed resources to be “discovered, accessed, allocated, monitored, accounted for, and billed for, etc., and in general managed as a single virtual system.” The Open Grid Services Architecture (OGSA) addresses this need for standardization by defining a set of core capabilities and behaviors that address key concerns in grid systems.

Virtualization technology has been identified as the perfect fit to issues that have caused frustration when using grids, such as hosting many dissimilar software applications on a single physical platform. In this direction, some research projects (e.g., Globus Virtual Workspaces) aimed at evolving grids to support an additional layer to virtualize computation, storage, and network resources.

2. Utility Computing

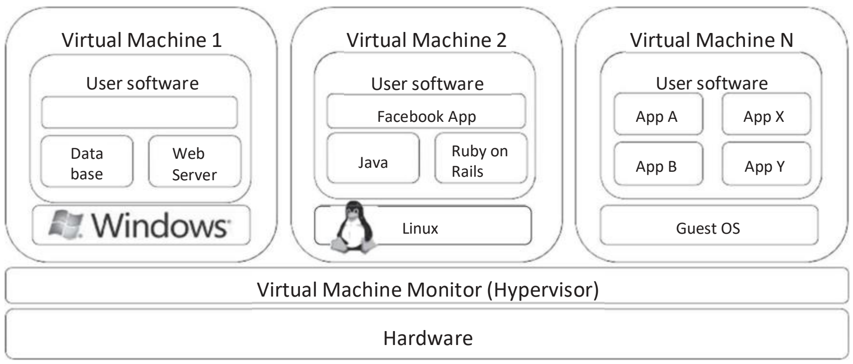
With increasing popularity and usage, large grid installations have faced new problems, such as excessive spikes in demand for resources coupled with strategic and adversarial behavior by users. Initially, grid resource management techniques did not ensure fair and equitable access to resources in many systems. Traditional metrics (throughput, waiting time, and slowdown) failed to capture the more subtle requirements of users. There were no real incentives for users to be flexible about resource requirements or job deadlines, nor provisions to accommodate users with urgent work.

In utility computing environments, users assign a “utility” value to their jobs, where utility is a fixed or time-varying valuation that captures various QoS constraints (deadline, importance, satisfaction). The valuation is the amount they are willing to pay a service provider to satisfy their demands. The service providers then attempt to maximize their own utility, where said utility may directly correlate with their profit. Providers can choose to prioritize high yield (i.e., profit per unit of resource) user jobs, leading to a scenario where shared systems are viewed as a marketplace, where users compete for resources based on the perceived utility or value of their jobs. Further information and comparison of these utility computing environments are available in an extensive survey of these platforms

3. Hardware Virtualization

Cloud computing services are usually backed by large-scale data centers composed of thousands of computers. Such data centers are built to serve many users and host many disparate applications. For this purpose, hardware virtualization can be considered as a perfect fit to overcome most operational issues of data center building and maintenance.

The idea of virtualizing a computer system resources, including processors, memory, and I/O devices, has been well established for decades, aiming at improving sharing and utilization of computer systems. Hardware virtualization allows running multiple operating systems and software stacks on a single physical platform. As depicted in Figure 1.2, a software layer, the virtual machine monitor (VMM), also called a hypervisor, mediates access to the physical hardware presenting to each guest operating system a virtual machine (VM), which is a set of virtual platform interfaces



Email Server Facebook App

FIGURE 1.2. A hardware virtualized server hosting three virtual machines, each one running distinct operating system and user level software stack.

Several VMM platforms exist that are the basis of many utilities or cloud computing environments. The most notable ones, VMWare, Xen, and KVM, are outlined in the following sections.

VMWare ESXi – VMware is a pioneer in the virtualization market. Its ecosys- tem of tools ranges from server and desktop virtualization to high-level management tools. ESXi is a VMM (Virtual Machine Manager) from VMWare. It is a bare-metal hypervisor, meaning that it installs directly on the physical server, whereas others may require a host operating system. It provides advanced virtualization techniques of processor, memory, and I/O. Especially, through memory ballooning and page sharing, it can overcommit memory, thus increasing the density of VMs inside a single physical server.

Xen—The Xen hypervisor started as an open-source project and has served as a base to other virtualization products, both commercial and open-source. It has pioneered the para-virtualization concept, on which the guest operating system, by means of a specialized kernel, can interact with the hypervisor, thus significantly improving performance. In addition to an

open-source distribution, Xen currently forms the base of commercial hypervisors of several vendors, most notably Citrix XenServer and Oracle VM.

KVM—The kernel-based virtual machine (KVM) is a Linux virtualization subsystem. Is has been part of the mainline Linux kernel since version 2.6.20, thus being natively supported by several distributions. In addition, activities such as memory management and scheduling are carried out by existing kernel features, thus making KVM simpler and smaller than hypervisors that take control of the entire machine.

KVM leverages hardware-assisted virtualization, which improves performance and allows it to support unmodified guest operating systems. currently, it supports several versions of Windows, Linux, and UNIX.

4. Virtual Appliances and the Open Virtualization Format

An application combined with the environment needed to run it (operating system, libraries, compilers, databases, application containers, and so forth) is referred to as a "virtual appliance. Packaging application environments in the shape of virtual appliances eases software customization, configuration, and patching and improves portability. Most commonly, an appliance is shaped as a VM disk image associated with hardware requirements, and it can be readily deployed in a hypervisor.

OVF's extensibility has encouraged additions relevant to management of data centers and clouds. Mathews et al. have devised virtual machine contracts (VMC)as an extension to OVF. A VMC aids in communicating and managing the complex expectations that VMs have of their runtime environment and vice versa. A simple example of a VMC is when a cloud consumer wants to specify minimum and maximum amounts of a resource that a VM needs to function. similarly, the cloud provider could express resource limits to bound resource consumption and costs.

5. Autonomic Computing

Autonomic or self-managing, systems rely on monitoring probes and gauges (sensors), on an adaptation engine (autonomic manager) for computing optimizations based on monitoring data, and on effectors to carry out changes on the system. IBM's Autonomic Computing Initiative has contributed to define the four properties of autonomic systems: self-configuration, self- optimization, self-healing, and self-protection. IBM has also suggested a reference model for autonomic control loops of autonomic managers, called MAPE-K (Monitor Analyze Plan Execute—Knowledge)

LAYERS AND TYPES OF CLOUDS

Cloud computing services are divided into three classes, according to the abstractionlevel of the capability provided and the service model of providers, namely:

1. Infrastructure as a Service,
2. Platform as a Service, and Software as a Service.

Infrastructure as a Service

A cloud infrastructure enables on-demand provisioning of servers running several choices of operating systems and a customized software stack. Infrastructure services are considered as the bottom layer of cloud computing systems. Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS).

One of the best examples is Amazon Web Services mainly offers IaaS, which inthe case of its EC2 service means offering VMs with a software stack that can be customized similar to how an ordinary physical server would be customized.

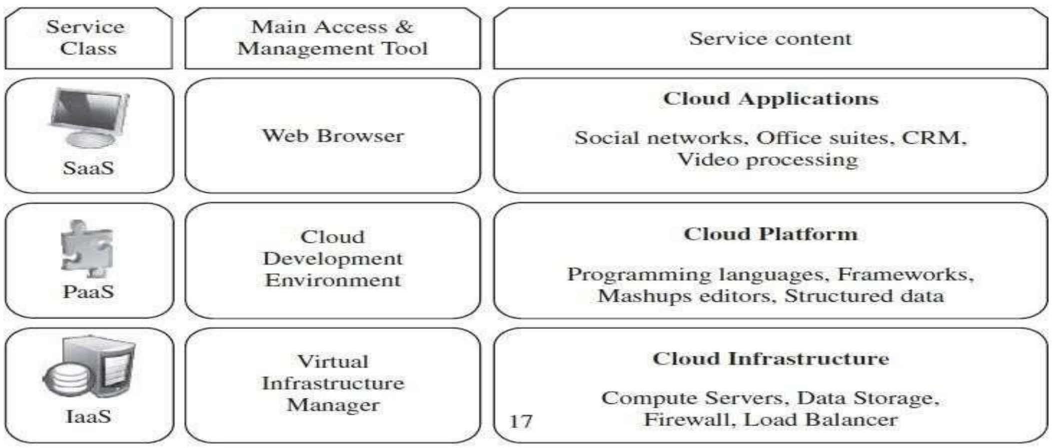


FIGURE 1.3. The cloud computing stack.

Platform as a Service

A *cloud platform* offers an environment on which developers create and deploy applications and do not necessarily need to know how many processors or how much memory that applications will be using. In addition, multiple program- ming models and specialized services (e.g., data access, authentication, and payments) are offered as building blocks to new applications.

Google AppEngine, an example of Platform as a Service, offers a scalable environment for developing and hosting Web applications, which should be written in specific programming languages such as Python or Java, and use the services' own proprietary structured object data store.

Software as a Service

Traditional desktop applications such as word processing and spreadsheet can now be accessed as a service in the Web. This model of delivering applications, known as Software as a Service (SaaS), alleviates the burden of software maintenance for customers and simplifies development and testing for providers.

Salesforce.com, which relies on the SaaS model, offers business productivity applications (CRM) that reside completely on their servers, allowing customers to customize and access applications on demand.

Deployment Models

Although cloud computing has emerged mainly from the appearance of public

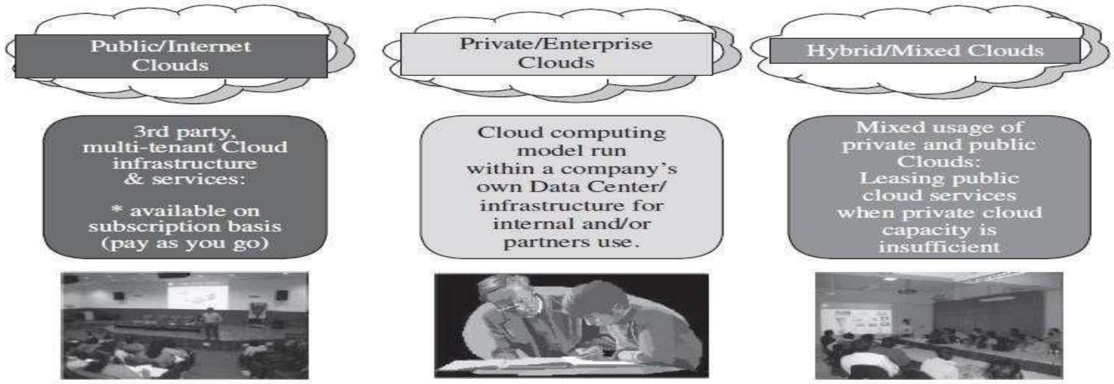


FIGURE 1.4. Types of clouds based on deployment models.

computing utilities, other deployment models, with variations in physical location and distribution, have been adopted. In this sense, regardless of its service class, a cloud can be classified as public, private, community, or hybrid based on model of deployment as shown figure below.

Public cloud & Private cloud: Public cloud as a "cloud made available in a pay-as-you-go manner to the general public". **Private cloud** as "internal data center of a business or other organization, not made available to the general public."

In most cases, establishing a private cloud means restructuring an existing infrastructure by adding virtualization and cloud-like interfaces. This allows users to interact with the local data center while experiencing the same advantages of public clouds, most notably self-service interface, privileged access to virtual servers, and per-usage metering and billing.

A **community cloud** is "shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations)."

A **hybrid cloud** takes shape when a private cloud is supplemented with computing capacity from public clouds. The approach of temporarily renting capacity to handle spikes in load is known as "cloud-bursting"

DESIRED FEATURES OF A CLOUD

Certain features of a cloud are essential to enable services that truly represent the cloud-computing model and satisfy expectations of consumers, and cloud offerings must be having following features:

1. Self-service
2. Per-usage metered and billed
3. Elastic,
4. Customizable.

The following are the features that are explained in detail.

1. **Self-Service**
   Consumers of cloud computing services expect on-demand, nearly instant access to resources. To support this expectation, clouds must allow self-service access so that customers can request, customize, pay, and use services without intervention of human operators.

2. **Per-Usage Metering and Billing**
   Cloud computing eliminates up-front commitment by users, allowing them to request and use only the necessary amount. Services must be priced on a short- term basis (e.g., by the hour), allowing users to release (and not pay for) resources as soon as they are not needed. For these reasons, clouds must implement features to allow efficient trading of service such as pricing, accounting, and billing. Metering should be done accordingly for different types of service (e.g., storage, processing, and bandwidth) and usage promptly reported, thus providing greater transparency.

3. **Elasticity**
   Cloud computing gives the illusion of infinite computing resources available on demand. Therefore, users expect clouds to rapidly provide resources in any quantity at any time. In particular, it is expected that the additional resources can be (a) provisioned, possibly automatically, when an application load increases and
   (b) released when load decreases (scale up and down).

4. **Customization**
   In a multi-tenant cloud a great disparity between user needs is often the case. Thus, resources rented from the cloud must be highly customizable. In the case of infrastructure services, customization means allowing users to deploy specialized virtual appliances and to be given privileged (root) access to the virtual servers. Other service classes (PaaS and SaaS) offer less flexibility and are not suitable for general-purpose computing, but still are expected to provide a certain level of customization.

CLOUD INFRASTRUCTURE MANAGEMENT

A key challenge IaaS providers face when building a cloud infrastructure is managing physical and virtual resources, namely servers, storage, and net- works, in a holistic fashion.

The software toolkit responsible for this orchestration is called a virtual infrastructure manager (VIM). This type of software resembles a traditional operating system—but instead of dealing with a single computer, it aggregates resources from multiple computers, presenting a uniform view to user and applications. Other terms include infrastructure sharing software and virtual infra- structure engine.

There are two categories of tools used to manage cloud they are

5. Cloud toolkits—includes those that "expose a remote and secure interface for creating, controlling and monitoring virtualize resources," but do not specializein VI management.

6. The virtual infrastructure managers—provide advanced features such as automatic load balancing and server consolidation, but do not expose remote cloud-like interfaces.

The availability of a remote cloud-like interface and the ability of managing many users and their permissions are the primary features that would distinguish "cloud toolkits" from "VIMs." However, here we place both categories of tools under the same group(of the VIMs) and, when applicable, we highlight the availability of a remote interface as a feature.

Virtually all VIMs we investigated present a set of basic features related tomanaging the life cycle of VMs, including networking groups of VMs together and setting up virtual disks for VMs. These basic features pretty much define whether a tool can be used in practical cloud deployments or not. On the other hand, only a handful of software present advanced features (e.g., high availability) which allow them to be used in large-scale production clouds.

Features

We now present a list of both basic and advanced features that are usually available in VIMs.

Virtualization Support— The multi-tenancy aspect of clouds requires multiple customers with disparate requirements to be served by a single hardware infrastructure. Virtualized resources (CPUs, memory, etc.) can be sized and resized with certain flexibility. These features make hardware virtualization, the ideal technology to create a virtual infrastructure that partitions a data center among multiple tenants.

Self-Service, On-Demand Resource Provisioning— Self-service access to resources has been perceived as one the most attractive features of clouds. This feature enables users to directly obtain services from clouds, such as spawning the creation of a server and tailoring its software, configurations, and security policies, without interacting with a human system administrator. This capability eliminates the need for more time-consuming, labor-intensive, human- driven procurement processes familiar to many in IT. Therefore, exposing a self-service interface, through which users can easily interact with the system, is a highly desirable feature of a **Vi manager**.

Multiple Backend Hypervisors— Different virtualization models and tools offer different benefits, drawbacks, and limitations. Thus, some Vi managers provide a uniform management layer regardless of the virtualization technology used.

This characteristic is more visible in open-source Vi managers, which usually provide pluggable drivers to interact with multiple hypervisors. In this direction, the aim of **libvirt**is to provide a uniform API that Vi managers can use to manage domains (a VM or container running an instance of an operating system) in virtualized nodes using standard operations that abstract hypervisor specific calls.

Storage Virtualization— Virtualizing storage means abstracting logical storage from physical storage. By consolidating all available storage devices in a data center, it allows creating virtual disks independent from device and location. Storage devices are commonly organized in a storage area network (SAN) and attached to servers via protocols such as Fibre Channel, iSCSI, and NFS; a storage controller provides the layer of abstraction between virtual and physical storage.

In the VI management sphere, storage virtualization support is often restricted tocommercial products of companies such as VMWare and Citrix.

Interface to Public Clouds— Researchers have perceived that extending the capacity of a local in-house computing infrastructure by borrowing resources from public clouds is advantageous. In this fashion, institutions can make good use of their available resources and, in case of spikes in demand, extra load can be offloaded to rented resources.

A VI manager can be used in a hybrid cloud setup if it offers a driver to manage the life cycle of virtualized resources obtained from external cloud providers. To the applications, the use of leased resources must ideally be transparent.

Virtual Networking— Virtual networks allow creating an isolated network on top of a physical infrastructure independently from physical topology and locations. A virtual LAN (VLAN) allows isolating traffic that shares a switched network, allowing VMs to be grouped into the same broadcast domain. Additionally, a VLAN can be configured to block traffic originated from VMs from other networks. Similarly, the VPN (virtual private network) concept is used to describe a secure and private overlay network on top of a public network (most commonly the public Internet)

Support for creating and configuring virtual networks to group VMs placed throughout a data center is provided by most VI managers. Additionally, VI managers that interface with public clouds often support secure VPNs connecting local and remote VMs.

Dynamic Resource Allocation— In cloud infrastructures, where applications have variable and dynamic needs, capacity management and demand prediction are especially complicated. This fact triggers the need for dynamic resource allocation aiming at obtaininga timely match of supply and demand.

Energy consumption reduction and better management of SLAs can be achieved by dynamically remapping VMs to physical machines at regular intervals. Machines that are not assigned any VM can be turned off or put on a low power state.

A number of VI managers include a dynamic resource allocation feature that continuously monitors utilization across resource pools and reallocates avail- able resources among VMs according to application needs.

Virtual Clusters— Several VI managers can holistically manage groups of VMs. This feature is useful for provisioning computing virtual clusters on demand, and interconnected VMs for multi-tier Internet applications.

Reservation and Negotiation Mechanism— When users request computational resourcesto available at a specific time, requests are termed **Advance reservations** (AR), in contrast to best-effort requests, when users request resources whenever available [54]. To support complex requests, such as AR, a VI manager must allow users to "lease" resources expressing more complex terms (e.g., the period of time of a reservation). This is especially useful in clouds on which resources are scarce; since not all requests may be satisfied immediately, they can benefit of VM placement strategies that support queues, priorities,and Advance reservations. Additionally, leases may be negotiated and renegotiated, allowing provider and consumer to modify a lease or present counter proposals until an agreement is reached. This feature is illustrated by the case in which an AR request for a given slot cannot be satisfied, but the provider can offer a distinct slot that is still satisfactory to the user.

High Availability and Data Recovery— The high availability (HA) feature of VI managers aims at minimizing application downtime and preventing business disruption. A few VI managers accomplish this by providing a failover mechanism, which detects failure of both physical and virtual servers and restarts VMs on healthy physical servers. This style of HA protects from host, but not VM, failures.

Data Recovery means data backup in clouds should consider the high data volume involved in VM management. Frequent backup of a large number of VMs, each one with multiple virtual disks attached, should be done with minimal interference in the systems performance. In this sense, some VI managers offer data protection mechanisms that perform incremental backups of VM images. The backup workload is often assigned to proxies, thus offloading production server and reducing network overhead.

Case Studies

In this section, we describe the main features of the most popular VI managers available. Only the most prominent and distinguishing features of each tool are discussed in detail. A detailed side-by-side feature comparison of VI managers is presented in Table 1.1.

Apache VCL: The Virtual Computing project has been incepted in 2004 by researchers at the North Carolina State University as a way to provide customized environments to computer lab users. The software compo- nents that support NCSU's initiative have been released as open-source and incorporated by the Apache Foundation.

Since its inception, the main objective of VCL has been providing desktop (virtual lab) and HPC computing environments anytime, in a flexible cost- effective way and with minimal intervention of IT staff. In this sense, VCL was one of the first projects to create a tool with features such as: self-service Web portal, to reduce administrative burden; advance reservation of capacity, to provide resources during classes; and deployment of customized machine images on multiple computers, to provide clusters on demand.

In summary, Apache VCL provides the following features: (i) multi-platform controller, based on Apache/PHP; (ii) Web portal and XML-RPC interfaces; (iii) support for VMware hypervisors (ESX, ESXi, and Server); (iv) virtual networks; (v) virtual clusters; and (vi) advance reservation of capacity.

AppLogic: AppLogic is a commercial VI manager, the flagship product of 3tera Inc. from California, USA. The company has labeled this product as a Grid OperatingSystem.

AppLogic provides a fabric to manage clusters of virtualized servers, focusing on managing multi-tier Web applications. It views an entire applica- tion as a collection of components that must be managed as a single entity. Several components such as firewalls, load balancers, Web servers, application servers, and database servers can be set up and linked together. Whenever the application is started, the system manufactures and assembles the virtual infrastructure requiredto run it. Once the application is stopped, AppLogic tears down the infrastructure built for it.

AppLogic offers dynamic appliances to add functionality such as Disaster Recovery and Power optimization to applications. The key differential of this approach is that additional functionalities are implemented as another pluggable appliance instead of being added as a core functionality of the VI manager.

In summary, 3tera AppLogic provides the following features: Linux-based controller; CLI and GUI interfaces; Xen backend; Global Volume Store (GVS) storage virtualization; virtual networks; virtual clusters; dynamic resource allocation; high availability; and data protection.

Citrix Essentials: The Citrix Essentials suite is one the most feature complete VI management software available, focusing on management and automation of data centers. Itis essentially a hypervisor-agnostic solution, currently supporting Citrix XenServer and Microsoft Hyper-V. Citrix Essentials provides the following features: Windows- based controller; GUI, CLI, Web portal, and XML-RPC interfaces which support for XenServer and Hyper-V hypervisors. Citrix Storage Link storage virtualization; virtual networks; dynamic resource allocation; three-level high avail- ability (i.e., recovery by VM restart, recovery by activating paused duplicate VM, and running duplicate VM continuously) [58]; data protection with Citrix Consolidated Backup.

Enomaly ECP: The Enomaly Elastic Computing Platform, in its most complete edition, offers most features a service provider needs to build an IaaS cloud.

Most notably, ECP Service Provider Edition offers a Web-based customer dashboard that allows users to fully control the life cycle of VMs. Usage accounting is performed in real time and can be viewed by users. Similar to the functionality of virtual appliance marketplaces, ECP allows providers and users to package and exchange applications.

In summary, Enomaly ECP provides the following features: Linux-based controller; Web portal and Web services (REST) interfaces; Xen back-end; interface to the AmazonEC2 public cloud; virtual networks; virtual clusters (ElasticValet).

Eucalyptus: The Eucalyptus framework was one of the first open-source projects to focus on building IaaS clouds. It has been developed with the intent of providing an open-source implementation nearly identical in functionality to Amazon Web Services APIs. Therefore, users can interact with a Eucalyptus cloud using the same tools they use to access Amazon EC2. It also distinguishes itself from other tools because it provides a storage cloud API— emulating the Amazon S3 API—for storing general user data and VM images.

In summary, Eucalyptus provides the following features: Linux-based con- troller with administration Web portal; EC2-compatible (SOAP, Query) and S3- compatible (SOAP, REST) CLI and Web portal interfaces; Xen, KVM, and VMWare backends; Amazon EBS-compatible virtual storage devices; interface to the Amazon EC2 public cloud; virtual networks.

Nimbus3: The Nimbus toolkit is built on top of the Globus framework. Nimbus provides most features in common with other open-source VI managers, such as an EC2-compatible front-end API, support to Xen, and a backend interface to Amazon EC2. However, it distinguishes from others by providing a Globus Web Services Resource Framework (WSRF) interface. It also provides a backend service, named Pilot, which spawns VMs on clusters managed by a local resource manager (LRM) such as PBS and SGE.

Nimbus' core was engineered around the Spring framework to be easily extensible, thus allowing several internal components to be replaced and also eases the integration with other systems.

In summary, Nimbus provides the following features: Linux-based control- ler; EC2-compatible (SOAP) and WSRF interfaces; Xen and KVM backend and a Pilot program to spawn VMs through an LRM; interface to the Amazon EC2 public cloud; virtual networks; one-click virtual clusters.

OpenNebula: OpenNebula is one of the most feature-rich open-source VI managers. It was initially conceived to manage local virtual infrastructure, but has also included remote interfaces that make it viable to build public clouds. Altogether, four programming APIs are available: XML-RPC and libvirt for local interaction; a subset of EC2 (Query) APIs and the OpenNebula Cloud API (OCA) for public access.

OpenNebula provides the following features: Linux-based controller; CLI, XML- RPC, EC2-compatible Query and OCA interfaces; Xen, KVM, and VMware backend; interface to public clouds (Amazon EC2, ElasticHosts); virtual networks; dynamic resource allocation; advance reserva- tion of capacity.

OpenPEX: OpenPEX (Open Provisioning and EXecution Environment) was constructedaround the notion of using advance reservations as the primary method for allocating VM instances.

OpenPEX provides the following features: multi-platform (Java) controller; Web portal and Web services (REST) interfaces; Citrix XenServer backend; advance reservation of capacity with negotiation.

oVirt: oVirt is an open-source VI manager, sponsored by Red Hat's Emergent Technology group. It provides most of the basic features of other VI managers.

oVirt provides the following features: Fedora Linux-based controller packaged as a virtual appliance; Web portal interface; KVM backend.

Platform ISF: Infrastructure Sharing Facility (ISF) is the VI manager offering from

Platform Computing. The company, mainly through its LSF family of products, has been serving the HPC market for several years.

ISF provides the following features: Linux-based controller packaged as a virtual appliance; Web portal interface; dynamic resource allocation; advance reservation of capacity; high availability.

VMWare vSphere and vCloud: vSphere is VMware's suite of tools aimed at transforming IT infrastructures into private clouds. In the vSphere architecture, servers run on the ESXi platform. A separate server runs vCenter Server, which centralizes control over the entire virtual infrastructure. Through the vSphere Client software, administrators connect to vCenter Server to perform various tasks.

The Distributed Resource Scheduler (DRS) makes allocation decisions based on predefined rules and policies. It continuously monitors the amount of resources available to VMs and, if necessary, makes allocation changes to meet VM requirements. In the storage virtualization realm, vStorage VMFS is a cluster file system to provide aggregate several disks in a single volume. VMFS is especially optimized to store VM images and virtual disks. It supports storage equipment that use Fibre Channel or iSCSI SAN.

vSphere provides the following features: Windows-based controller (vCenter Server); CLI, GUI, Web portal, and Web services interfaces; VMware ESX, ESXi backend; VMware vStorage VMFS storage virtualization; interface to external clouds (VMware vCloud partners); virtual networks (VMWare Distributed Switch); dynamic resource allocation (VMware DRM); high availability; data protection (VMWare Consolidated Backup).

INFRASTRUCTURE AS A SERVICE PROVIDERS

Public Infrastructure as a Service providers commonly offer virtual servers containing one or more CPUs, running several choices of operating systems and a customized software stack. In addition, storage space and communication facilities are often provided.

Features

IAAS offers a set of specialized features that can influence the cost benefit ratio to be experienced by user applications when moved to the cloud.

The most relevant features are:

1. Geographic distribution of data centers.
2. Variety of user interfaces and APIs to access the system.
3. Specialized components and services that aid Particular applications (e.g., load- balancers, firewalls).
4. Choice of virtualization platform and operating systems and
5. Different billing methods and period (e.g., prepaid vs. postpaid, hourly vs. monthly).

Geographic Presence— To improve availability and responsiveness, a provider of worldwide services would typically build several data centers distributed around the world. For example, Amazon Web Services presents the concept of availability zones and regions for its EC2 service. Availability zones are distinct locations that are engineered to be insulated from failures in other availability zones and provide inexpensive, low-latency network connectivity to other availability zones in the same region. Regions, in turn, are geographically dispersed and will be in separate geographic areas or countries.

User Interfaces and Access to Servers— Ideally, a public IaaS provider must provide multiple access means to its cloud, thus catering for various users and their preferences. Different types of user interfaces (UI) provide different levels of abstraction, the most common being graphical user interfaces (GUI), command-line tools (CLI), and Web service (WS) APIs.

GUIs are preferred by end users who need to launch, customize, and monitor a few virtual servers and do not necessarily need to repeat the process several times. On the other hand, CLIs offer more flexibility and the possibility of automating repetitive tasks via scripts(e.g., start and shutdown a number of virtual servers at regular intervals).

Advance Reservation of Capacity— Advance reservations allow users to request for an IaaS provider to reserve resources for a specific time frame in the future, thus ensuring that cloud resources will be available at that time. However, most clouds only support best- effort requests that means users can request server whenever resources are available .

Amazon Reserved Instances is a form of advance reservation of capacity, allowing users to pay a fixed amount of money in advance to guarantee resource availability atanytime during an agreed period and then paying a discounted hourly rate when resources are in use. However, only long periods of 1 to 3 years are offered; therefore, users cannot express their reservations in finer granularities—for example, hours or days.

Automatic Scaling and Load Balancing— Automatic scaling is a highly desirable feature of IaaS clouds. It allows users to set conditions for when they want their applications to scale up and down, based on application-specific metrics such as transactions per second, number of simultaneous users, request latency, and so forth.

When the number of virtual servers is increased by automatic scaling, incoming traffic must be automatically distributed among the available servers. This activity enables applications to promptly respond to traffic increase while also achieving greater fault tolerance.

Service-Level Agreement. Service-level agreements (SLAs) are offered by IaaS providers to express their commitment to delivery of a certain QoS. To customers it serves as awarranty. An SLA usually include availability and performance guarantees. Additionally, metrics must be agreed upon by all parties as well as penalties for violating these expectations.

Most IaaS providers focus their SLA terms on availability guarantees, specifying the minimum percentage of time the system will be available during a certain period. For instance, Amazon EC2 states that "if the annual uptime Percentage for a customer drops below 99.95% for the service year, that customer is eligible to receive a service credit equalto 10% of their bill.3"

Hypervisor and Operating System Choice— Traditionally, IaaS offerings have been based on heavily customized open-source Xen deployments. IaaS providers needed expertise in Linux, networking, virtualization, metering, resource management, and many other low-level aspects to successfully deploy and maintain their cloud offerings.

More recently, there has been an emergence of turnkey IaaS platforms such as VMWare VCloud and Citrix Cloud Center (C3) which have lowered the barrier of entry for IaaS competitors, leading to a rapid expansion in the IaaS marketplace.

Case Studies

Amazon Web Services: Amazon WS4 (AWS) is one of the major players in the cloud computing market. It pioneered the introduction of IaaS clouds in 2006. It offers a variety cloud services, most notably: S3 (storage), EC2 (virtual servers), Cloudfront (content delivery), Cloudfront Streaming (video streaming), Simple DB (structured datastore), RDS (Relational Database), SQS (reliable messaging), and Elastic MapReduce (data processing). The ElasticCompute Cloud (EC2) offers Xen-based virtual servers (instances) that can be instantiated from Amazon Machine Images (AMIs). Instances are available in a variety of sizes, operating systems, architectures, and price. CPU capacity of instances is measured in Amazon Compute Units and, although fixed for each instance, vary among instance types from 1 (small instance) to 20 (high CPU instance). Each instance provides a certain amount of non persistent disk space; a persistence disk service (Elastic Block Storage) allows attaching virtual disks to instances with space up to 1TB. Elasticity can be achieved by combining the Cloud Watch, Auto Scaling and Elastic Load Balancing features, which allow the number of instances to scale up and down automatically based on a set of customizable rules, and traffic to be distributed across available instances. Fixed IP address (Elastic IPs) are not available by default, but can be obtained at an additional cost.

Flexiscale: Flexiscale is a UK-based provider offering services similar in nature to Amazon Web Services. Flexiscale cloud provides the following features: available in UK; Web services (SOAP), Web-based user interfaces; access to virtual server mainly via SSH (Linux) and Remote Desktop (Windows); 100% availability SLA with automatic recoveryof VMs in case of hardware failure; per hour pricing; Linux and Windows operating systems; automatic scaling (horizontal/vertical).

Joyent: Joyent's Public Cloud offers servers based on Solaris containers virtualization technology. These servers, dubbed accelerators, allow deploying various specializedsoftware-stack based on a customized version of Open- Solaris operating system, which include by default a Web-based configuration tool and several pre-installed software, suchas Apache, MySQL, PHP, Ruby on Rails, and Java. Software load balancing is available as an accelerator in addition to hardware load balancers. A notable feature of Joyent's virtual servers is automatic vertical scaling of CPU cores, which means a virtual server can make use of additional CPUs automatically up to the maximum number of cores available in the physical host.

The Joyent public cloud offers the following features: multiple geographic locations inthe United States; Web-based user interface; access to virtual server via SSH and Web- based administration tool; 100% availability SLA; per month pricing; OS-level virtualization Solaris containers; Open- Solaris operating systems; automaticscaling(vertical).

GoGrid: GoGrid, like many other IaaS providers, allows its customers to utilize a range of pre-made Windows and Linux images, in a range of fixed instance sizes. GoGrid also offers "value- added" stacks on top for applications such as high- volume Web serving, e-

Commerce, and database stores. It offers some notable features, such as a "hybrid hosting" facility, which combines traditional dedicated hosts with auto-scaling cloud server infrastructure. As part of its core IaaS offerings, GoGrid also provides free hardware load balancing, auto-scaling capabilities, and persistent storage, features that typically add an additional cost for most other IaaS providers.

Rackspace Cloud Servers: Rackspace Cloud Servers is an IaaS solution that provides fixed size instances in the cloud. Cloud Servers offers a range of Linux- based pre-made images. A user can request different-sized images, where the size is measured by requested RAM, not CPU.

PLATFORM AS A SERVICE PROVIDERS

Public Platform as a Service providers commonly offer a development and deployment environment that allow users to create and run their applications with little or no concern to low-level details of the platform. In addition, specific programming languages and frameworks are made available in the platform, as well as other services such as persistent data storage and in memory caches.

Features

Programming Models, Languages, and Frameworks: Programming models made available by IaaS providers define how users can express their applications using higher levels of abstraction and efficiently run them on the cloud platform.

Each model aims at efficiently solving a particular problem. In the cloud computing domain, the most common activities that require specialized models are: processing of large dataset in clusters of computers (MapReduce model), development of request-based Web services and applications; definition and orchestration of business processes in the form of workflows (Workflow model); and high-performance distributed execution of various computational tasks.

For user convenience, PaaS providers usually support multiple programming languages. Most commonly used languages in platforms include Python and Java (e.g., Google AppEngine), .NET languages (e.g., Microsoft Azure), and Ruby (e.g., Heroku). Force.com has devised its own programming language (Apex) and an Excel-like query language, which provide higher levels of abstraction to key platform functionalities.

A variety of software frameworks are usually made available to PaaS developers, depending on application focus. Providers that focus on Web and enterprise application hosting offer popular frameworks such as Ruby on Rails, Spring, Java EE, and .NET.

Persistence Options: A persistence layer is essential to allow applications to record their state and recover it in case of crashes, as well as to store user data. Web and enterprise application developers have chosen relational databases as the preferred persistence method. These databases offer fast and reliable structured data storage and transaction processing, but may lack scalability to handle several peta bytes of data stored in commodity computers. In the cloud computing domain, distributed storage technologies have emerged, which seek to be robust and highly scalable, at the expense of relational structure and convenient query languages.

CASE STUDIES

Aneka: Aneka is a .NET-based service-oriented resource management and development platform. Each server in an Aneka deployment (dubbed Aneka cloud node) hosts the Aneka container, which provides the base infrastructure that consists of services for persistence, security (authorization, authentication and auditing), and communication (message handling and dispatching). Cloud nodes can be either physical server, virtual machines (Xen Server and VMware are supported), and instances rented from Amazon EC2. The Aneka container can also host any number of optional services that can be added by developers to augment the capabilities of an Aneka Cloud node, thus providing a single, extensible framework for orchestrating various application models.

Several programming models are supported by such task models to enable execution of legacy HPC applications and Map Reduce, which enables a variety of data-mining and search applications. Users request resources via a client to a reservation services manager of the Aneka master node, which manages all cloud nodes and contains scheduling service to distribute request to cloud nodes.

App Engine: Google App Engine lets you run your Python and Java Web applications on elastic infrastructure supplied by Google. App Engine allows your applications to scale dynamically as your traffic and data storage requirements increase or decrease. It gives developers a choice between a Python stack and Java. The App Engine serving architectureis notable in that it allows real-time auto- scaling without virtualization for many common types of Web applications. However, such auto-scaling is dependent on the application developer using a limited subset of the native APIs on each platform, and in some instances you need to use specific Google APIs such as URLFetch, Data store, and mem cache in place of certain native API calls. For example, a deployed App Engine application cannot write to the file system directly (you must use the Google Data store) or open a socket or access another host directly (you must use Google URL fetch service). A Java application cannot create a new Thread either.

Microsoft Azure: Microsoft Azure Cloud Services offers developers a hosted. NET Stack (C#, VB.Net, ASP.NET). In addition, a Java & Ruby SDK for .NET Services is also available. The Azure system consists of a number of elements. The Windows Azure Fabric Controller provides auto-scaling and reliability, and it manages memory resources and load balancing. The .NET Service Bus registers and connects applications together. The .NET Access Control identity providers include enterprise directories and Windows LiveID. Finally, the .NET Workflow allows construction and execution of workflow instances.

Force.com: In conjunction with the Salesforce.com service, the Force.com PaaS allows developers to create add-on functionality that integrates into main Salesforce CRM SaaS application. Force.com offers developers two approaches to create applications that can be deployed on its SaaS plaform: a hosted Apex or Visualforce application. Apex is a proprietary Java-like language that can be used to create Salesforce applications. Visual force is an XML-like syntax for building UIs in HTML, AJAX, or Flex to overlay over the Salesforce hosted CRM system. An application store called App Exchange is also provided, which offers a paid & free application directory.

Heroku: Heroku is a platform for instant deployment of Ruby on Rails Web applications.In the Heroku system, servers are invisibly managed by the platform and are never exposed to users. Applications are automatically dispersed across different CPU cores and servers.

maximizing performance and minimizing contention. Heroku has an advanced logic layer than can automatically route around failures, ensuring seamless and uninterrupted service at all times.

CHALLENGES AND RISKS

Despite the initial success and popularity of the cloud computing paradigm and the extensive availability of providers and tools, a significant number of challenges and risks are inherent to this new model of computing. Providers, developers, and end users must consider these challenges and risks to take good advantage of cloud computing. Issues to be faced include user privacy, data security, data lock-in, availability of service, disaster recovery, performance, scalability, energy- efficiency, and programmability.

Security, Privacy, and Trust: Security and privacy affect the entire cloud computing stack, since there is a massive use of third-party services and infrastructures that are used to host important data or to perform critical operations. In this scenario, the trust toward providers is fundamental to ensure the desired level of privacy for applications hosted in the cloud. Legal and regulatory issues also need attention. When data are moved into the Cloud, providers may choose to locate them anywhere on the planet. The physical location of data centers determines the set of laws that can be applied to the management of data. For example, specific cryptography techniques could not be used because they are not allowed in some countries. Similarly, country laws can impose that sensitive data, such as patient health records, are to be stored within national borders.

Data Lock-In and Standardization: A major concern of cloud computing users is about having their data locked-in by a certain provider. Users may want to move data and applications out from a provider that does not meet their requirements. However, in their current form, cloud computing infrastructures and platforms do not employ standard methods of storing user data and applications. Consequently, they do not interoperate and user data are not portable.

The answer to this concern is standardization. In this direction, there are efforts to create open standards for cloud computing. The Cloud Computing Interoperability Forum (CCIF) was formed by organizations such as Intel, Sun, and Cisco in order to "enable a global cloud computing ecosystem whereby organizations are able to seamlessly work together for the purposes for wider industry adoption of cloud computing technology." The development of the Unified Cloud Interface (UCI) by CCIF aims at creating a standard programmatic point of access to an entire cloud infrastructure. In the hardware virtualization sphere, the Open Virtual Format (OVF) aims at facilitating packing and distribution of software to be run on VMs so that virtual appliances can be made portable— that is, seamlessly run on hypervisor of different vendors.

Availability, Fault-Tolerance, and Disaster Recovery: It is expected that users will have certain expectations about the service level to be provided once their applications are moved to the cloud. These expectations include availability of the service, its overall performance, and what measures are to be taken when something goes wrong in the system or its components. In summary, users seek for a warranty before they can comfortably move their business to the cloud. SLAs, which include QoS requirements, must be ideally set up between customers and cloud computing providers to act as warranty. An SLA specifies the details of the service to be provided, including availability and performance guarantees.

Additionally, metrics must be agreed upon by all parties, and penalties for violatingthe expectations must also be approved.

Resource Management and Energy-Efficiency: One important challenge faced byproviders of cloud computing services is the efficient management of virtualized resource pools. Physical resources such as CPU cores, disk space, and network bandwidth must be sliced and shared among virtual machines running potentially heterogeneous workloads. The multi-dimensional nature of virtual machines complicates the activity of finding a good mapping of VMs onto available physical hosts while maximizing user utility. Dimensionsto be considered include: number of CPUs, amount of memory, size of virtual disks, and network bandwidth. Dynamic VM mapping policies may leverage the ability to suspend, migrate, and resume VMs as an easy way of preempting low-priority allocations in favor of higher-priority ones. Migration of VMs also brings additional challenges such as detecting when to initiate a migration, which VM to migrate, and where to migrate. In addition, policies may take advantage of live migration of virtual machines to relocate data centerload without significantly disrupting running services. In this case, an additional concern is the trade-off between the negative impact of a live migration on the performance and stability of a service and the benefits to be achieved with that migration. Another challenge concerns the outstanding amount of data to be managed in various VM management activities. Such data amount is a result of particular abilities of virtual machines, including the ability of traveling through space (i.e., migration) and time (i.e., check pointing and rewinding), operations that may be required in load balancing, backup, and recovery scenarios. In addition, dynamic provisioning of new VMs and replicating existing VMsrequire efficient mechanisms to make VM block storage devices (e.g., image files) quickly available at selected hosts. Data centers consumer large amounts of electricity. Accordingto a data published by HP, 100 server racks can consume 1.3MWof power and another 1.3 MW are required by the cooling system, thus costing USD 2.6 million per year. Besides the monetary cost, data centers significantly impact the environment in terms of $CO2$ emissions from the cooling systems.

COMPUTING PARADIGMS:

High-Performance Computing:

For many years, HPC systems emphasize the raw speed performance. The speed of HPC systems has increased from Gflops in the early 1990s to now Pflops in 2010. This improvement was driven mainly by the demands from scientific, engineering, and manufacturing communities.

Top 500 most powerful computer systems in the world are measured by floating- point speed in Linpack benchmark results. However, the number of supercomputer users is limited to less than 10% of all computer users.

Today, the majority of computer users are using desktop computers or large servers when they conduct Internet searches and market-driven computing tasks.

Three New Computing Paradigms

- With the introduction of SOA, **Web 2.0 services** become available.

- Advances in virtualization make it possible to see the growth of **Internet clouds** as a new computing paradigm.
- The maturity of radio-frequency identification (RFID), Global Positioning System (GPS), and sensor technologies has triggered the development of the **Internet of Things (IoT).**

Computing Paradigm Distinctions

- Centralized computing
- Parallel computing
- Distributed computing
- Cluster omputing
- Cloud computing
- Grid computing

Centralized computing:

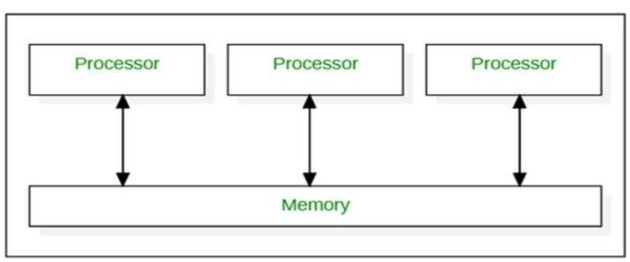
This is a computing paradigm by which all computer **resources are centralized in one physical system.**

All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS.Many data centers and supercomputers are centralized systems, but they are used inparallel, distributed, and cloud computing applications

Parallel Computing:

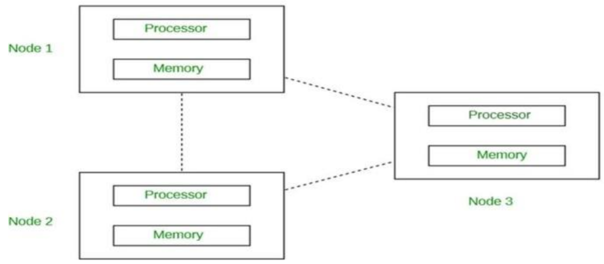
Parallel computing is defined as a type of computing where multiple computer systems are used simultaneously. Here a problem is broken into sub-problems and then further broken down into instructions. These instructions from each sub-problem are executed concurrently on different processors.

Here in the below diagram, you can see how the parallel computing system consists of multiple processors that communicate with each other and perform multiple tasks over a shared memory simultaneously.



Distributed Computing:

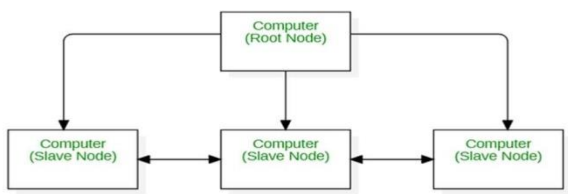
Distributed computing is defined as a type of computing where multiple computer systems work on a single problem. Here all the computer systems are linked together, andthe problem is divided into sub-problems where each part is solved by different computer systems. The goal of distributed computing is to increase the performance and efficiency of the system and ensure fault tolerance. In the below diagram, each processor has its own local memory, and all the processors communicate with each other over a network.



Cluster Computing:

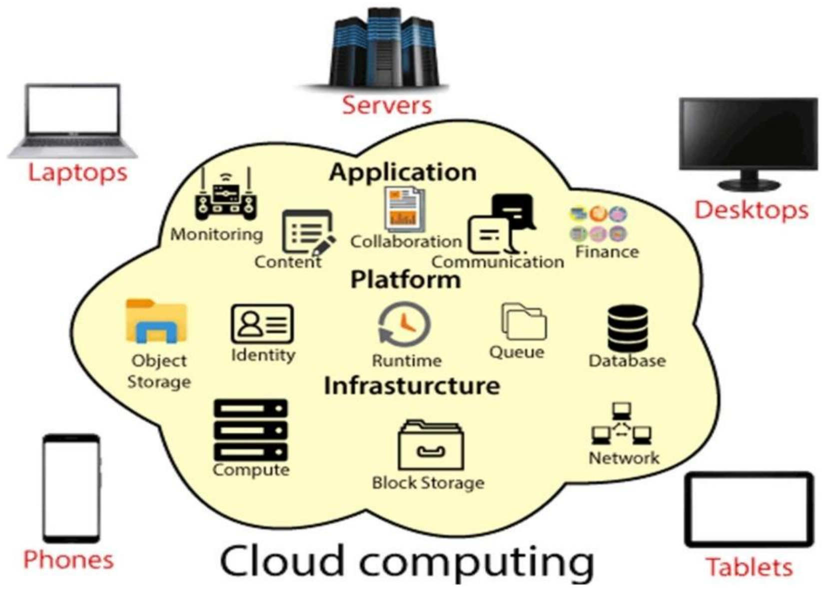
A cluster is a group of independent computers that work together to perform thetasks given. Cluster computing is defined as a type of computing that consists of two or more independent computers, referred to as nodes, that work together to execute tasks as a single machine.

The goal of cluster computing is to increase the performance, scalability and simplicity ofthe system. As you can see in the below diagram, all the nodes, (irrespective of whether they are a parent node or child node), act as a single entity to perform the tasks.



Cloud computing :

An Internet cloud of resources can be either a centralized or a distributed computing system. The cloud applies parallel or distributed computing, or both. Clouds can be built with physical or virtualized resources over large data centers thatare centralized or distributed



Grid Computing:

Grid computing is defined as a type of computing where it is constitutes a networkof computers that work together to perform tasks that may be difficult for a single machine to handle. All the computers on that network work under the same umbrella and are termedas a virtual supercomputer.

The tasks they work on is of either high computing power and consist of large data sets. All communication between the computer systems in grid computing is done on the "data grid".The goal of grid computing is to solve more high computational problems in less time and improve productivity.