DIGITAL NOTES
ON

CLOUD COMPUTING
(R20A0521)

B.TECH IV YEAR – ISEM

(2023-24)

DEPARTMENT
OF
COMPUTER SCIENCE AND ENGINEERING

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015
Certified)Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, INDIA.

CLOUD COMPUTING

Objectives

- To understand the various distributed system models and evolving computingparadigms
- To gain knowledge in virtualization of computer resources
- To realize the reasons for migrating into cloud
- To introduce the various levels of services that can be achieved by a cloud.
- To describe the security aspects in cloud and the services offered by a cloud.

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.

UNIT- II

Migrating into a Cloud: Introduction, Broad Approaches to Migrating into the Cloud, the Seven-Step Model of Migration into a Cloud.

Virtualization: Virtual Machines and Virtualization of Clusters and data centers-Implementation Levels of Virtualization -Virtualization Structures/Tools and Mechanisms-Virtualization of CPU, Memory, and I/O Devices-Virtual Clusters and Data Centers

UNIT- III

Infrastructure as a Service (IAAS) & Platform (PAAS): Virtual machines provisioning and Migration services, Virtual Machines Provisioning and Manageability, Virtual Machine Migration Services, VM Provisioning and Migration in Action. On the Management of Virtual machines for Cloud Infrastructures- Aneka—Integration of Private and Public Clouds.

UNIT- IV

Software as a Service (SAAS) &Data Security in the Cloud: Software as a Service SAAS), Google App Engine – Centralizing Email Communications- Collaborating via Web-Based Communication Tools-An Introduction to the idea of Data Security.

UNIT- V

SLA Management in cloud computing: Traditional Approaches to SLO Management, Typesof SLA, Life Cycle of SLA, SLA Management in Cloud.

COURSE OUTCOMES:

1. Ability to analyze various service delivery models of cloud computing
2. Ability to interpret the ways in which the cloud can be programmed and deployed.
3. Ability to comprehend the virtualization and cloud computing concepts
4. Assess the comparative advantages and disadvantages of Virtualization technology
5. Analyze authentication, confidentiality and privacy issues in cloud computing

INDEX

S. No	Name of the Topic	Page No
1	**Cloud Computing Fundamentals**	1
2	Definition of Cloud computing	1
3	Roots of Cloud Computing	2
4	Layers and Types of Clouds	5
5	Desired Features of a Cloud	11
6	Cloud Infrastructure Management	11
7	**Computing Paradigms**	23
8	High-Performance Computing	24
9	Parallel Computing	24
10	Distributed Computing	25
11	Cluster Computing	26
12	Grid Computing	26
13	**Migrating into a Cloud**	28
14	Broad Approaches to Migrating into the Cloud	29
15	Seven-Step Model of Migration into a Cloud	30
16	**Virtualization:** Implementation Levels of Virtualization	31
17	Virtualization Structures/Tools and Mechanisms	35
18	Virtualization of CPU, Memory, and I/O Devices	39
19	Virtual Clusters and Data Centers	44
20	**Infrastructure as a Service (IAAS) & Platform (PAAS)**	47
21	Virtual machines provisioning and Migration services	50
22	Virtual Machines Provisioning and Manageability	56
23	Virtual Machine Migration Services	58
24	VM Provisioning and Migration in Action	58
25	On the Management of Virtual machines for CloudInfrastructures	60
26	Aneka—Integration of Private and Public Clouds.	61
27	**Software as a Service (SAAS) &Data Security in the Cloud**	66
28	Software as a Service SAAS)	66
29	Google App Engine – Centralizing Email Communications	69

30	Collaborating via Web-Based Communication Tools	73
31	**SLA Management in cloud computing**	79
32	Traditional Approaches to SLO Management	79
33	Types of SLA	80
34	Life Cycle of SLA	81
35	SLA Management in Cloud	82

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

"Cloud is a parallel and distributed computing system consisting of a collection ofinter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service- level agreements (SLA) established through negotiation between the service provider and consumers."

"Clouds are a large pool of easily usable and accessible virtualized resources (suchas hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization"

"This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service Level Agreements."

"Clouds are hardware based services offering compute, network, and storage capacity where Hardware management is highly abstracted from the buyer, buyers incur infrastructure costs as variable OPEX, and infrastructure capacityis highly elastic."

Key characteristics of cloud computing

(1) the illusion of infinite computing resources;
(2) the elimination of an up-front commitment by cloud users;
(3) the ability to pay for use…as needed

The National Institute of Standards and Technology (NIST) characterizes **cloud computing** as ". . . a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

Most common characteristics which a cloud should have:
(i) pay-per-use (no ongoing commitment, utility prices); (ii) elastic capacity and the illusion of infinite resources; (iii) self-service interface; and (iv) resourcesthat is abstracted or virtualized.

Roots of Cloud Computing

The roots of clouds computing can be tracked 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.

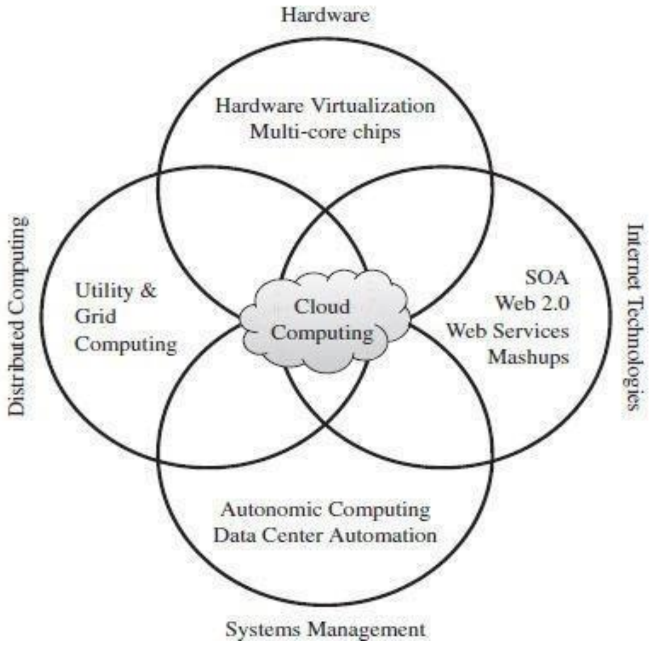


FIGURE 1.1. Convergence of various advances leading to the advent of cloud computing.

The IT world is currently experiencing a switch from in-house generated computing power into utility-supplied computing resources delivered over the Internet as Web services.

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".

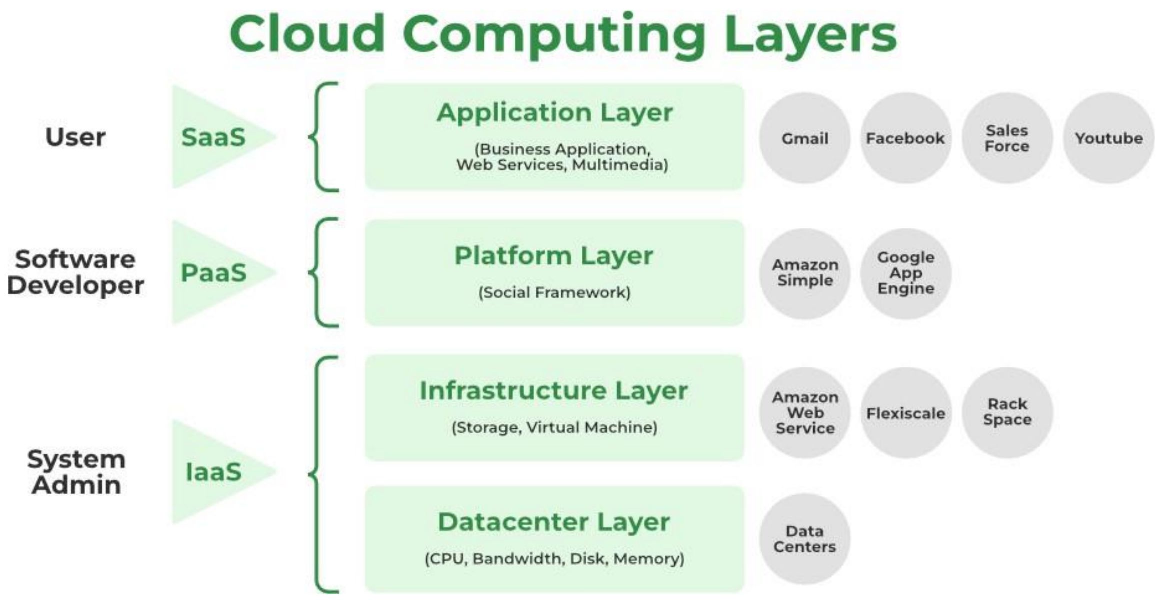
This model brings benefits to both consumers and providers of IT services. Consumers can attain reduction on IT-related costs by choosing to obtain cheaper services from external providers as opposed to heavily investing on IT infrastructure and personnel hiring. The "on-demand" component of this model allows consumers to adapt their IT usage to rapidly increasing or unpredictable computing needs.

Providers of IT services achieve better operational costs; hardware and software infrastructures are built to provide multiple solutions and serve many users, thus increasing efficiency and ultimately leading to faster return on investment (ROI) as well as lower total cost of ownership (TCO).

In the 1970s, companies who offered common data processing tasks, such as payroll automation, operated time-shared main frames as utilities, which could serve dozens of applications and often operated close to 100% of their capacity.

The mainframe era collapsed with the advent of fast and inexpensive microprocessors and IT data centers moved to collections of commodity servers. Apart from its clear advantages, this new model inevitably led to isolation of workload into dedicated servers, mainly due to incompatibilities between softwarestacks and operating systems.

In addition, the unavailability of efficient computer networks meant that IT infrastructure should be hosted in proximity to where it would be consumed. Altogether, these facts have prevented the utility computing reality of taking place on modern computer systems. These facts reveal the potential of delivering computing services with the speed and reliability that businesses enjoy with their local machines. The benefits of economies of scale and high utilization allow providers to offer computing services for a fraction of what it costs for a typical company that generates its own computing power.



SOA, Web Services, Web 2.0, and Mashups

The emergence of Web services (WS) open standards has significantly contributed to advances in the domain of software integration. Web services can combine together applications running on different messaging product platforms, 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 arepackaged 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 in a standard definition language and have a published interface. The maturity of WS has enabled the creation of powerful services that can be accessed on-demand, in a uniform way. An enterprise application that follows the SOA paradigm is a collection of services that together perform complex business logic.

In the consumer Web, information and services may be programmatically aggregated, acting as building blocks of complex compositions, called service mashups. Many service providers, such as Amazon, del.icio.us, 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.

In the Software as a Service (SaaS) domain, cloud applications can be built as compositions of other services from the same or different providers. Services such as user authentication, e-mail, payroll management, and calendars are examples of building blocks that can be reused and combined in a business solution in case a single, ready-made system does not provide all those features.

Grid Computing

Grid computing enables aggregation of distributed resources and transparently access to them. Most production grids such as Tera Grid and EGEE seek to share compute and storage resources distributed across different administrative domains, with their main 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.

Globus Toolkit is a middleware that implements several standard Grid services andover the years has aided the deployment of several service-oriented Grid infrastructures and applications.

The development of standardized protocols for several grid computing activities has contributed—theoretically—to allow delivery of on-demand computing services over the Internet. However, ensuring QoS in grids has been perceived as a difficult endeavor. Lack of performance isolation has prevented grids adoption in a variety of scenarios, especially on environments where resources are oversubscribed or users are uncooperative.

Another issue that has lead to frustration when using grids is the availability of resources with diverse software configurations, including disparate operating systems, libraries, compilers, runtime environments, and so forth. At the same time, user applications would often run only on specially customized environments. Consequently, a portability barrier has often been present on most grid infrastructures, inhibiting users of adopting grids as utility computing environments

Utility Computing

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 providersthen 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.

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.
The idea of virtualizing a computer system's 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 physicalplatform.

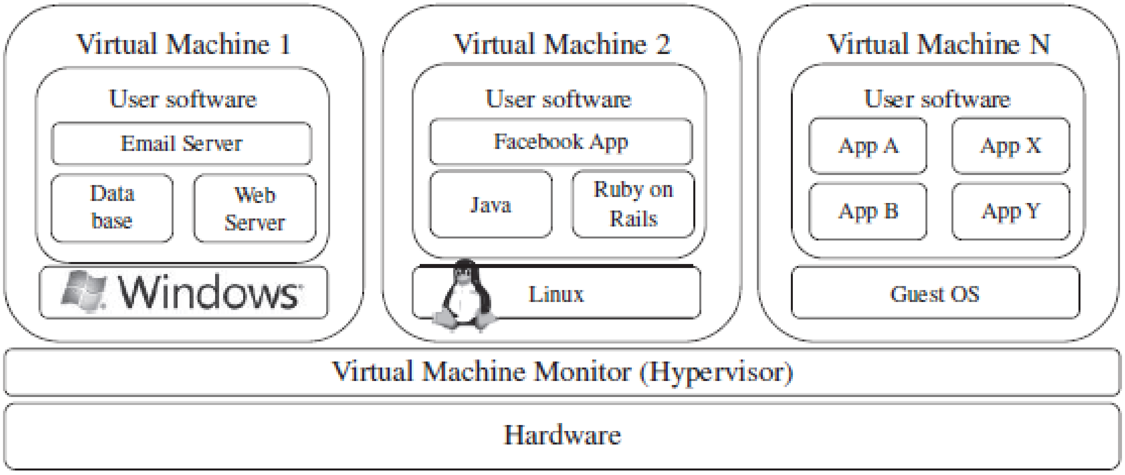


FIGURE 1.2. A hardware virtualized server hosting three virtual machines, each one running distinct operating system and user level software stack.

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

The advent of several innovative technologies—multi-core chips, para- virtualization, hardware-assisted virtualization, and live migration of VMs—has contributed to an increasing adoption of virtualization on server systems.

Perceived benefits were improvements on sharing and utilization, better manageability, and higher reliability.

There are three basic capabilities regarding management of workload in a virtualized system, namely isolation, consolidation, and migration

Workload isolation is achieved since all program instructions are fully confined inside a VM, which leads to improvements in security. Better reliability is also achieved because software failures inside one VM do not affect others

The consolidation of several individual and heterogeneous workloads onto a single physical platform leads to better system utilization. This practice is also employed for overcoming potential software and hardware incompatibilities incase of upgrades, given that it is possible to run legacy and new operation systems concurrently

Workload migration, also referred to as application mobility, targets at facilitating hardware maintenance, load balancing, and disaster recovery. It is done by encapsulating a guest OS state within a VM and allowing it to be suspended, fully serialized, migrated to a different platform, and resumed immediately or preserved to be restored at a later date. A VM's state includes a full disk or partition image, configuration files, and an image of its RAM.

A number of VMM platforms exist that are the basis of many utility or cloud computing environments. The most notable ones are VMWare, Xen, andKVM.
VMWARE ESXi: is a VMM from VMWare. It is a bare-metal hypervisor,meaning that it installs directly on the physical server, whereas others may requirea host operating system. It provides advanced virtualization techniques ofprocessor, memory, and I/O. Especially, through page sharing, it can over commitmemory, 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.

KVM: The kernel-based virtual machine (KVM) is a Linux virtualization subsystem. It 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

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 toas 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 asa VM disk image associated with hardware requirements, and it can be readily deployed in a hypervisor. The VMWare virtual appliance marketplace allows users to deploy appliances onVMWare hypervisors or on partners public clouds, and Amazon allows developersto share specialized Amazon Machine Images (AMI) and monetize their usage on Amazon EC2.

In a multitude of hypervisors, where each one supports a different VM image format and the formats are incompatible with one another, a great deal of interoperability issues arises. In order to facilitate packing and distribution of software to be run on VMs several vendors, including VMware, IBM, Citrix, Cisco, Microsoft, Dell, and HP, have devised the Open Virtualization Format

(OVF). It aims at being “open, secure, portable, efficient and extensible” [32]. An OVF package consists of a file, or set of files, describing the VM hardware characteristics (e.g., memory, network cards, and disks), operating system details, startup, and shutdown actions, the virtual disks themselves, and other metadata containing product and licensing information. OVF also supports complex packages composed of multiple VMs.

Autonomic Computing

The increasing complexity of computing systems has motivated research on autonomic computing, which seeks to improve systems by decreasing human involvement in their operation. In other words, systems should manage themselves,with high-level guidance from humans

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)

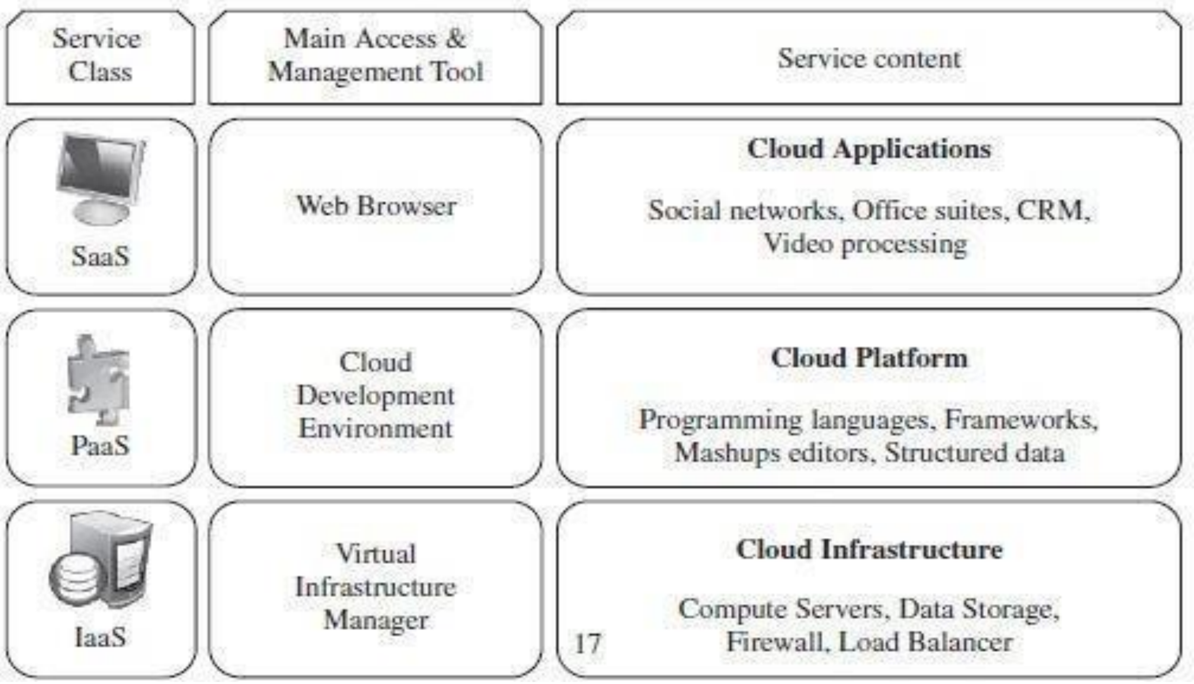
The large data centers of cloud computing providers must be managed in anefficient way. In this sense, the concepts of autonomic computing inspiresoftware technologies for data center automation, which may perform taskssuch as: management of service levels of running applications; management ofdata center capacity; proactive disaster recovery; and automation of VMprovisioning

LAYERS AND TYPES OFCLOUDS

Cloud computing services are divided into three classes
(1) Infrastructure as a Service, (2) Platform as a Service, and (3) Softwareas aService

Figure 1.3 depicts the layered organization of the cloud stackfrom physical infrastructure to applications.

These abstraction levels can also be viewed as a layered architecture whereservices of a higher layer can be composed from services of the underlying layerA core middleware manages physical resources andthe VMs deployed on top of them; in addition, it provides the required features(e.g., accounting and billing) to offer multi-tenant pay- as-you-goservices.



Service Class	Main Access & Management Tool	Service content
SaaS	Web Browser	**Cloud Applications** Social networks, Office suites, CRM, Video processing
PaaS	Cloud Development Environment	**Cloud Platform** Programming languages, Frameworks, Mashups editors, Structured data
IaaS	Virtual Infrastructure Manager	**Cloud Infrastructure** Compute Servers, Data Storage, Firewall, Load Balancer

FIGURE 1.3 The cloud computing stack Infrastructure as a Service Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS). A cloud infrastructure enables on-demand provisioning of servers running several choices of operating systems and a customized software stack. Infrastructure services are considered to be the bottom layer of cloud computing systems.

Amazon Web Services mainly offers IaaS, which in the case of its EC2service means offering VMs with a software stack that can be customized similar to how an ordinary physical server would be customized. Users are given privileges to perform numerous activities to the server, such as: starting and stopping it, customizing it by installing software packages, attaching virtual disks to it, and configuring access permissions and firewalls rules.

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 programming models and specialized services (e.g., data access, authentication, and payments) are offered as building blocks to new applications.

Google App Engine, 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.

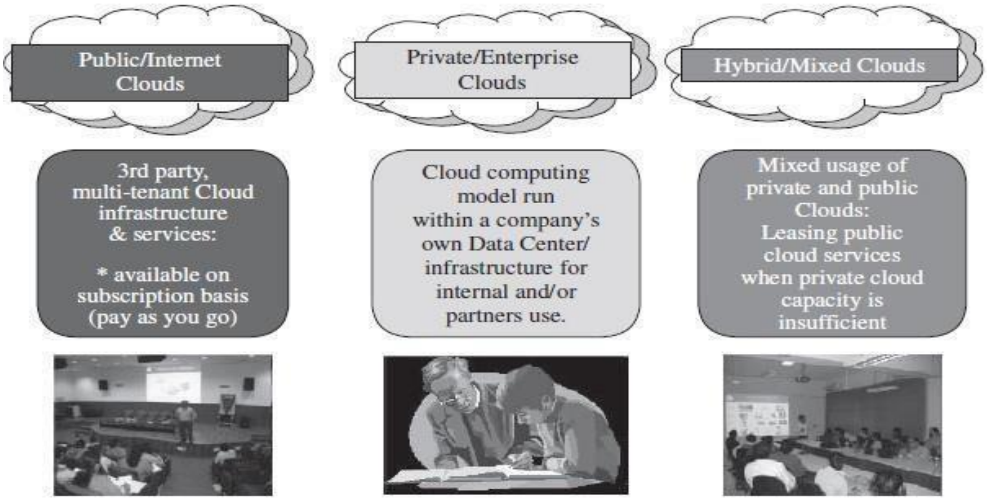
Software as a Service

Applications reside on the top of the cloud stack. Services provided by this layer can be accessed by end users through Web portals. Therefore, consumers are increasingly shifting from locally installed computer programs to on-line software services that offer the same functionally. 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

A cloud can be classified as public, private, community, or hybrid based on model of deployment as shown in Figure 1.4.



Public/Internet Clouds
Private/Enterprise Clouds
Hybrid/Mixed Clouds
3rd party, multi-tenant Cloud infrastructure & services:
* available on subscription basis (pay as you go)
Cloud computing model run within a company's own Data Center/ infrastructure for internal and/or partners use.
Mixed usage of private and public Clouds: Leasing public cloud services when private cloud capacity is insufficient

FIGURE 1.4. Types of clouds based on deployment models.

Public cloud: "cloud made available in a pay-as-you-go manner to the general public"
Private cloud: "internal data center of a business or other organization, not made available to the general public."
Community cloud:"shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations) **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"**

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 offeringsmust be

(i) self-service, (ii) per-usage metered and billed, (iii) elastic, and (iv)Customizable

Self-Service: clouds must allow self-service access so that customers can request, customize, pay, and use services without intervention of human operators

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), allowingusers to release (and not pay for) resources as soon as they are not needed

Elasticity: Cloud computing gives the illusion of infinite computing resources available on demand. Therefore users expect clouds to rapidly provide resources inany 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)

Customization: resources rented from the cloud must be highly customizable. customization means allowing users to deploy specialized virtual appliances and to be given privileged (root) access to the virtual servers.

CLOUD INFRASTRUCTURE MANAGEMENT

A key challenge IaaS providers face when building a cloud infrastructure is managing physical and virtual resources, namely servers, storage, and networks, in a holistic fashion. The orchestration of resources must be performed in a way to rapidly and dynamically provision resources to applications.

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.

Features

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 “eliminatesthe need for more time-consuming, labor-intensive, human driven procurement processes familiar to many in IT”.

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.

Interface to Public Clouds: 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.

Dynamic Resource Allocation: Increased awareness of energy consumption in data centers has encouraged the practice of dynamic consolidating VMs in a fewer number of servers. 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 obtaining a 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. In the same fashion, overheating canbe avoided by moving load away from hotspots.

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 resources to available at a specific time, requests are termed advance reservations

(AR), in contrast to best- effort requests, when users request resources whenever available. 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. This problem has been addressed in OpenPEX, which incorporates a bilateral negotiation protocol that allows users and providers to come to an alternative agreement by exchanging offers and counter offers.

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. For mission critical applications, when a failover solution involving restarting VMs does not suffice, additional levels of fault tolerance that rely on redundancy of VMs are implemented. In this style, redundant and synchronized VMs (running or in standby) are kept in a secondary physical server. The HA solution monitors failures of system components such as servers, VMs, disks, and network and ensures that a duplicate VM serves the application in case of failures. Data backup in clouds should take into account 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

Apache VCL: The Virtual Computing Lab 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. Apache VCL provides the following features:
(i) multi-platform controller, based on Apache/PHP;(ii)WebportalandXML-RPCinterfaces;(iii)supportforVMwarehypervisors
(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 Operating System.
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. It is essentially a hypervisor-agnostic solution, currently supporting Citrix XenServer and Microsoft Hyper-V. Citrix Essentials provides the following features: Windowsbased controller; GUI, CLI, Web portal, and XML-RPC interfaces; support for XenServer and Hyper-V hypervisors; Citrix Storage Link storage virtualization;virtual networks; dynamic resource allocation; three-level high availability (i.e., recovery by VM restart, recovery by activating paused duplicate VM, and running duplicate VM continuously); data protection with Citrix ConsolidatedBackup.

Enomaly ECP: The Enomaly Elastic Computing Platform, in its most complete edition, offers most features a service provider needs to build an IaaS cloud.
Enomaly ECP provides the following features: Linux-based controller; Web portal and Web services (REST) interfaces; Xen back-end; interface to the Amazon EC2 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 ServicesAPIs.
Eucalyptus provides the following features: Linux-based controller 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; virtualnetworks.

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.

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 reservation of capacity.

OpenPEX: OpenPEX (Open Provisioning and EXecution Environment) was constructed around the notion of using advance reservations as the primary methodfor allocatingVMinstances.
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 oVirt provides the following features: Fedora Linux-based controller packaged as a virtual appliance; Web portal interface; KVMbackend.

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 ConsolidatedBackup).

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 facilitiesare often provided.

Features

IaaS offerings can be distinguished by the availability of specialized features that influence the cost_benefit ratio to be experienced by user applications when movedto the cloud. The most relevant features are: (i) geographic distribution of data centers; (ii) variety of user interfaces and APIs to access the system; (iii) specialized components and services that aid particular applications (e.g., loadbalancers, firewalls); (iv) choice of virtualization platform and operating

systems; and (v) different billing methods and period (e.g., prepaid vs. post-paid, 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.

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 is, users requests are server wheneverresources are available.

Automatic Scaling and Load Balancing: Elasticity is a key characteristic of the cloud computing model. Applications often need to scale up and down to meet varying load conditions. Automatic scaling is a highly desirable feature of IaaS clouds. It allow 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 a warranty. 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.

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.