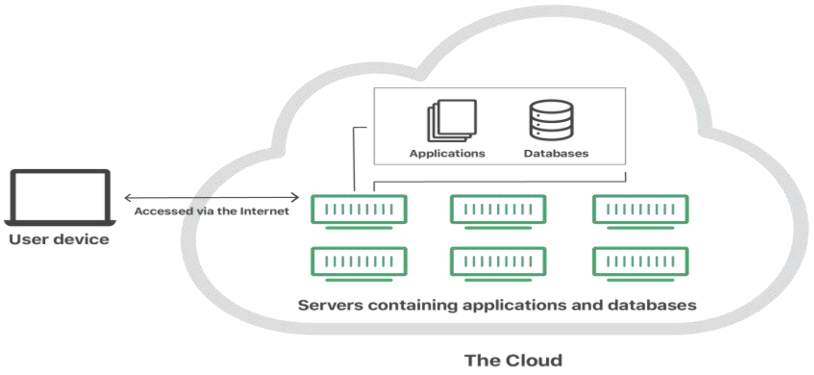
INDEX

UNIT -1

Definition of 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.


Applications
Databases
User device
Accessed via the Internet
Servers containing applications and databases
The Cloud

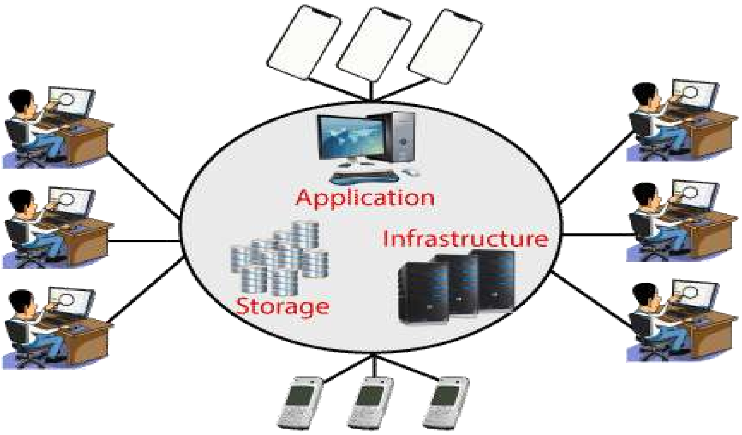
The cloud enables users to access the same files and applications from almost any device, because the computing and storage takes place on servers in a data centre, instead of locally on the user device. Therefore, a user can log into their Instagram account on a new phone after their old phone breaks 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 like Gmail 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:

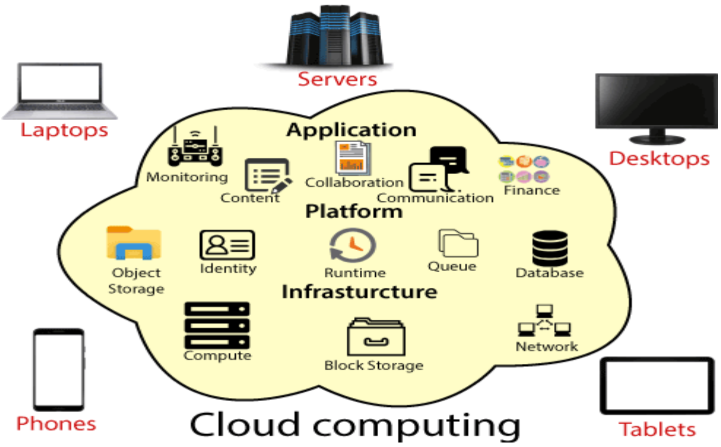
The term “Cloud Computing” refers to services provided by the cloud which is responsible for delivering of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud (Internet).

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 centre. With an on-premises data centre, 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 doing all the set-up, we become responsible for maintaining it through its entire lifecycle.

But 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 will be 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:

- **Cost:** It reduces the huge capital costs of buying hardware and software.
- **Speed:** Resources can be accessed in minutes, typically within a few clicks.
- **Scalability:** We can increase or decrease the requirement of resources according to the business requirements.
- **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.
- **Reliability:** Backup and recovery of data are less expensive and extremely fast for business continuity.
- **Security:** Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Cloud computing shares characteristics with:

- **Client–server model**—*Client–server computing* refers broadly to any distributed application that distinguishes between service providers (servers) and service requestors (clients).
- **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.
- **Fog computing—**Distributed computing paradigm that provides data, compute, storage and application services closer to the client or near-user edge devices, such as network 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.
- **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.
- **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.

- **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).
- **Green computing—**Study and practice of environmentally sustainable computing or IT.
- **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

- Agility for organizations
- Cost reductions
- Device and location independence
- No Maintenance required

Multitenancy enables sharing of resources and costs across a large pool of users thus allowing for:

- Centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
- Peak-load capacity increases (users need not engineer and pay for the resources and equipment to meet their highest possible load-levels)
- Utilization and efficiency improvements for systems that are often only 10–20% utilized.
- Performance is monitored by IT experts from the service provider, and consistent and loosely coupled architectures are constructed using web services as the system interface.
- Productivity may be increased when multiple users can work on the same data simultaneously, rather than waiting for it to be saved and emailed. Time may be saved as information does not need to be re-entered when fields are matched, nor do users need to install application software upgrades to their computer.
- Availability improves with the use of multiple redundant sites, which makes well-designed cloud computing suitable for business continuity and disaster recovery.
- Scalability and elasticity via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis in near real-time (Note, the VM startup time varies by VM type, location, OS and cloud providers), without users having to engineer for peak loads.
- Security can improve due to centralization of data, increased security-focused resources

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

- On-demand self-service.
- Broad network access.
- Resource pooling.
- Rapid elasticity.
- Measured service.

ROOTS OF CLOUD COMPUTING

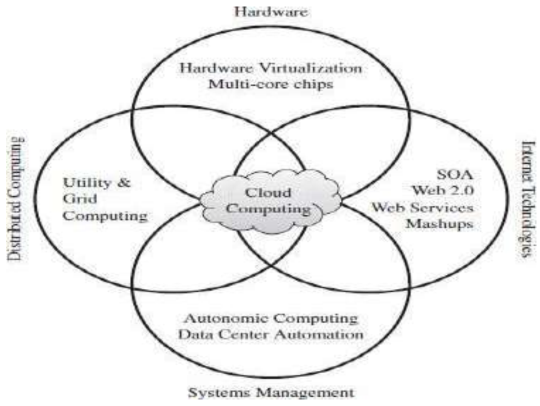
We can track the roots of clouds computing by observing the advancement of several technologies, especially in hardware (virtualization, multi-core chips), Internet technologies (Web services, service-oriented architectures, Web 2.0), distributed computing (clusters, grids), and systems management (autonomic 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.

❖ 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”



❖ 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 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 in a 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.

❖ 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.

❖ 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

❖ 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

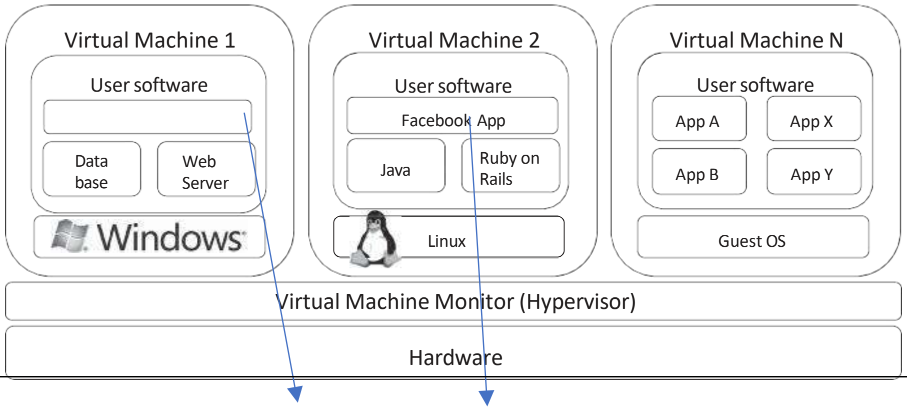


FIGURE 1.2. A hardware virtualized server hosting three virtual machines, each one running distinct operating system and user level software stack.

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.

❖ 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.

❖ 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

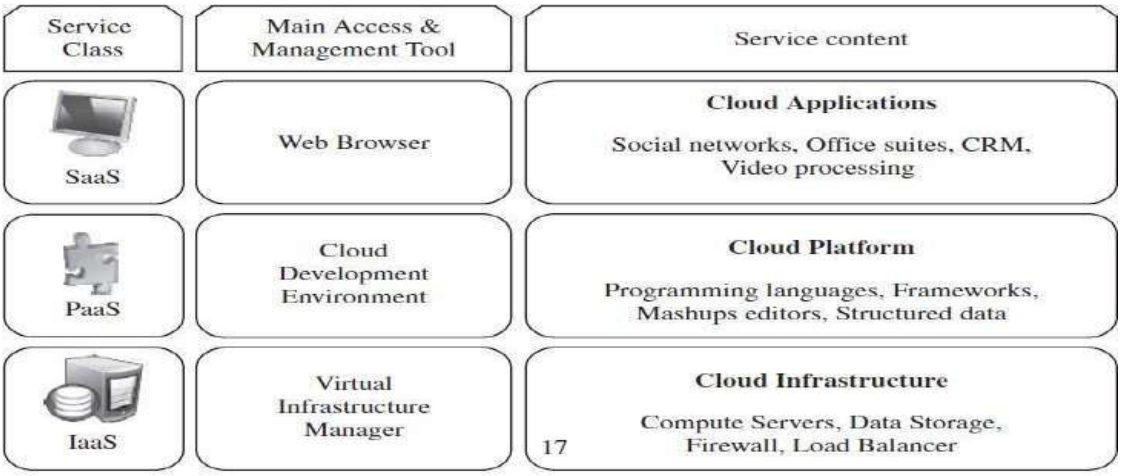
Cloud computing services are divided into three classes, according to the abstraction level of the capability provided and the service model of providers, namely:

- Infrastructure as a Service,
- 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 in the 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.



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.

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 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.

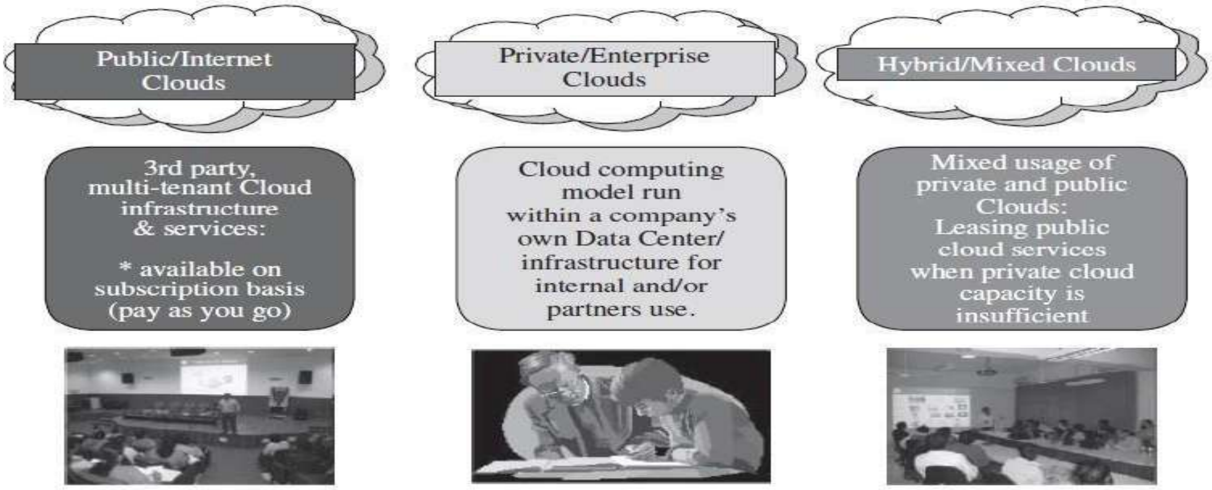

Public/Internet Clouds
Private/Enterprise Clouds
Hybrid/Mixed Clouds
3rd party, multi-tenant Cloud infrastructure & services:
* available on subscription basis (pay as you go)
Cloud computing model run within a company's own Data Center/ infrastructure for internal and/or partners use.
Mixed usage of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

- Self-service
- Per-usage metered and billed
- Elastic,
- Customizable.

❖ 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.

❖ 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.

❖ 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).

- **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

- Cloud toolkits—includes those that “expose a remote and secure interface for creating, controlling and monitoring virtualize resources,” but do not specialize in VI management.

- 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 to managing 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 to commercial 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 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.

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 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 [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.

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:

.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., load- balancers, firewalls).

.iv Choice of virtualization platform and operating systems and

.v 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 at anytime 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 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. 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 equal to 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.

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

Or chestrating 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 architecture is 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 violating the expectations must also be approved.

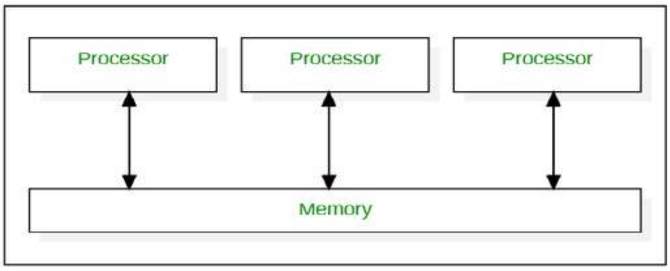
Resource Management and Energy-Efficiency: One important challenge faced by providers of cloud computing services is the efficient management of virtualized resource pools. Physical resources such as CPU cores, disk space, and network bandwidth must be sliced and shared among virtual machines running 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. Dimensions to 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 center load 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 VMs require efficient mechanisms to make VM block storage devices (e.g., image files) quickly available at selected hosts. Data centers consumer large amounts of electricity. According to 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

Parallel Computing:

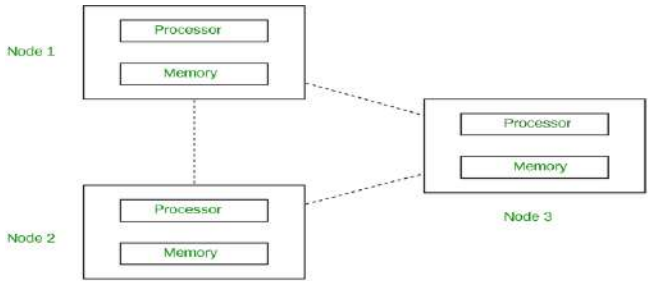
Parallel computing is defined as a type of computing where multiple computer systems are used simultaneously. Here a problem is broken into sub-problems and then further broken down into instructions. These instructions from each sub-problem are executed concurrently on different processors.

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:

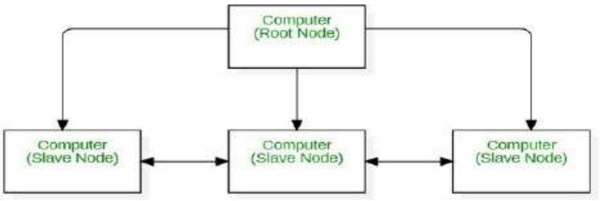
Distributed computing is defined as a type of computing where multiple computer systems work on a single problem. Here all the computer systems are linked together, and the problem is divided into sub-problems where each part is solved by different computer systems. The goal of distributed computing 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 the tasks 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.

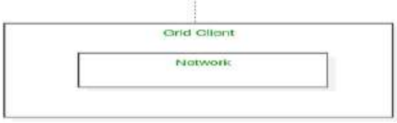
The goal of cluster computing is to increase the performance, scalability and simplicity of the system. As you can see in the below diagram, all the nodes, (irrespective of whether they are a parent node or child node), act as a single entity to perform the tasks.



Grid Computing:

Grid computing is defined as a type of computing where it is constitutes a network of 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 termed as 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.



UNIT - 2

MIGRATING INTO A CLOUD: INTRODUCTION

Cloud computing has been a hotly debated and discussed topic amongst IT professionals and researchers both in the industry and in academia. There are intense discussions on several blogs, in Web sites, and in several research efforts. This also resulted in several entrepreneurial efforts to help leverage and migrate into the cloud given the myriad issues, challenges, benefits, and limitations and lack of comprehensive understanding of what cloud computing can do.

On the one hand, there were these large cloud computing IT vendors like Google, Amazon, and Microsoft, who had started offering cloud computing services on what seemed like a demonstration and trial basis though not explicitly mentioned. They were charging users fees that in certain contexts demonstrated very attractive pricing models.

Most enterprises today are powered by captive data centers. In most large or small enterprises today, IT is the backbone of their operations. Invariably for these large enterprises, their data centers are distributed across various geographies.

They comprise systems and software that span several generations of products sold by a variety of IT vendors. In order to meet varying loads, most of these data centers are provisioned with capacity beyond the peak loads experienced.

Many data center management teams have been continuously innovating their management practices and technologies.

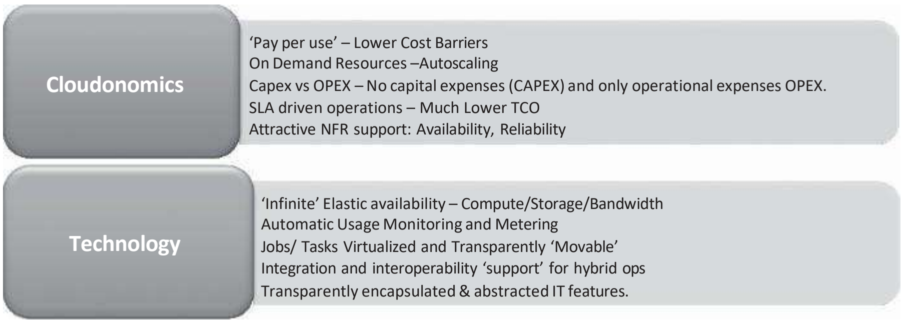
Cloud computing turned attractive to them because they could pass on the additional demand from their IT setups onto the cloud while paying only for the usage and being unencumbered by the load of operations and management.

The Promise of the Cloud

- The promise of the cloud both on the business front (the attractive cloudonomics) and the technology front widely aided the CxOs to spawn out several non-mission critical IT needs from the ambit of their captive traditional data centers to the appropriate cloud service.
- Invariably, these IT needs had some common features: They were typically Web-oriented; they represented seasonal IT demands; they were amenable to parallel batch processing; they were non-mission critical and therefore did not have high security demands. They included scientific applications too [7]. Several small and medium business enterprises, however, leveraged the cloud much beyond the cautious user.
- Many startups opened their IT departments exclusively using cloud services—

very successfully and with high ROI. Having observed these successes, several large enterprises have started successfully running pilots for leveraging the cloud.

- Many large enterprises run SAP to manage their operations. SAP itself is experimenting with running its suite of products: SAP Business One as well as SAP NetWeaver on Amazon cloud offerings.
- Gartner, Forrester, and other industry research analysts predict that a substantially significant percentage of the top enterprises in the world would have migrated most of their IT needs to the cloud offerings by 2012, thereby demonstrating the widespread impact and benefits from cloud computing. Indeed, the promise of the cloud has been significant in its impact.



Cloudonomics

'Pay per use' – Lower Cost Barriers
On Demand Resources –Autoscaling
Capex vs OPEX – No capital expenses (CAPEX) and only operational expenses OPEX.
SLA driven operations – Much Lower TCO
Attractive NFR support: Availability, Reliability

Technology

'Infinite' Elastic availability – Compute/Storage/Bandwidth
Automatic Usage Monitoring and Metering
Jobs/ Tasks Virtualized and Transparently 'Movable'
Integration and interoperability 'support' for hybrid ops
Transparently encapsulated & abstracted IT features.

The promise of the cloud computing services.

BROAD APPROACHES TO MIGRATING INTO THE CLOUD

Why Migrate:

There are economic and business reasons why an enterprise application can be migrated into the cloud, and there are also several number of technological reasons.

Many of these efforts come up as initiatives in adoption of cloud technologies in the enterprise, resulting in integration of enterprise applications running off the captive data centers with the new ones that have been developed on the cloud.

At the core, migration of an application into the cloud can happen in one of several ways:

- Either the application is clean and independent.
- Perhaps some degree of code needs to be modified and adapted or the design (and therefore the code) needs to be first migrated into the cloud computing service environment
- Perhaps the migration results in the core architecture being migrated for a cloud computing service setting, this resulting in a new architecture being developed, along with the accompanying design and code implementation.
- Perhaps while the application is migrated as is, it is the usage of the application

that needs to be migrated and therefore adapted and modified.

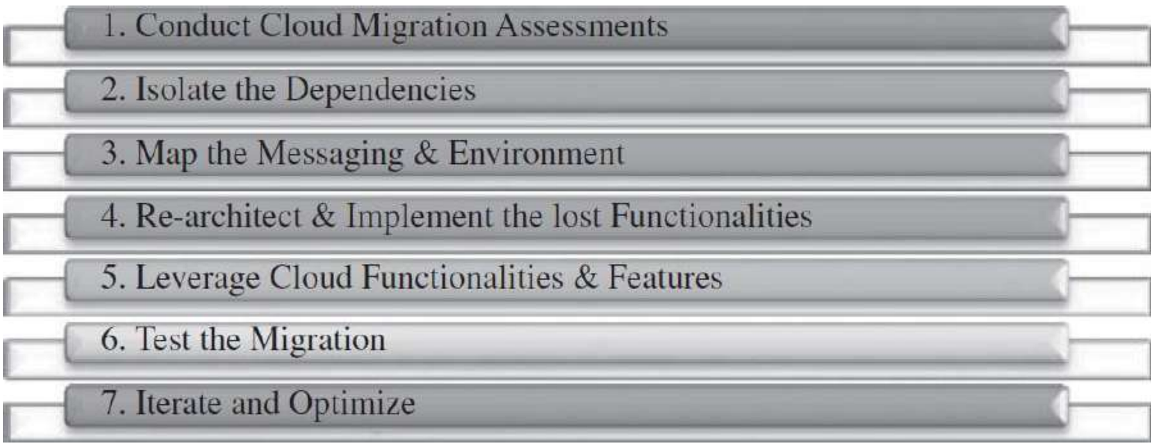
- Migration can happen at five levels i.e.,
    - Application
    - Code
    - Design
    - Architecture
    - Usage

With due simplification, the migration of an enterprise application is best captured by the following:

$$P \text{—} P^{l}_{C}\ 1\ P^{l}_{l} \text{—} P^{l}_{\mathrm{OFC}}\ 1\ \mathrm{P}^{l}_{l}$$

- Where *P* is the application before migration running in captive data center.
- P^{0}_{C} is the application part after migration into a (hybrid) cloud.
- P^{0}_{l} is the part of application being run in the captive local data center.
- P^{0}_{OFC} is the application part optimized for cloud.

Seven-Step Model of Migration into a Cloud



1. Conduct Cloud Migration Assessments
2. Isolate the Dependencies
3. Map the Messaging & Environment
4. Re-architect & Implement the lost Functionalities
5. Leverage Cloud Functionalities & Features
6. Test the Migration
7. Iterate and Optimize

Step-1: Cloud migration assessments comprise assessments to understand the issues involved in the specific case of migration at the application level or the code, the design, the architecture, or usage levels. These assessments are about the cost of migration as well as about the ROI that can be achieved in the case of production version.

Step-2: Isolating all systemic and environmental dependencies of the enterprise application components within the captive data center.

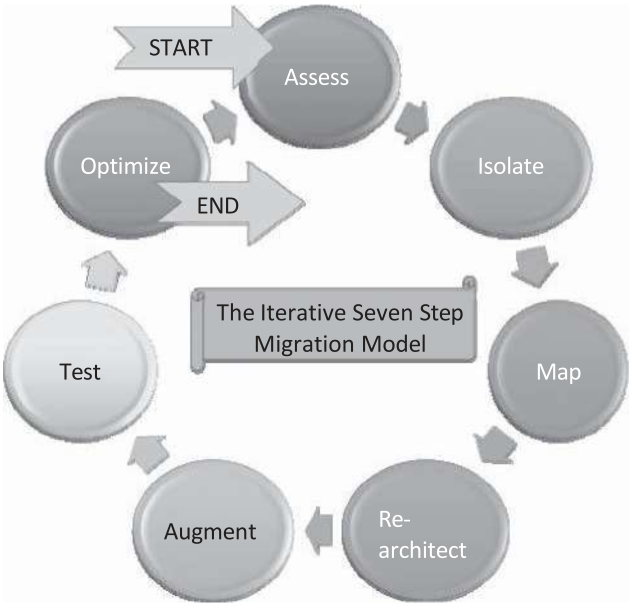
Step-3: Generating the mapping constructs between what shall possibly remain in the local captive data center and what goes onto the cloud.

Step-4: substantial part of the enterprise application needs to be rearchitected, redesigned, and reimplemented on the cloud

Step-5: We leverage the intrinsic features of the cloud computing service to augment our enterprise application in its own small ways.

Step-6: We validate and test the new form of the enterprise application with an extensive test suite that comprises testing the components of the enterprise application on the cloud as well

Step-7: Test results could be positive or mixed. In the latter case, we iterate and optimize as appropriate. After several such optimizing iterations, the migration is deemed successful

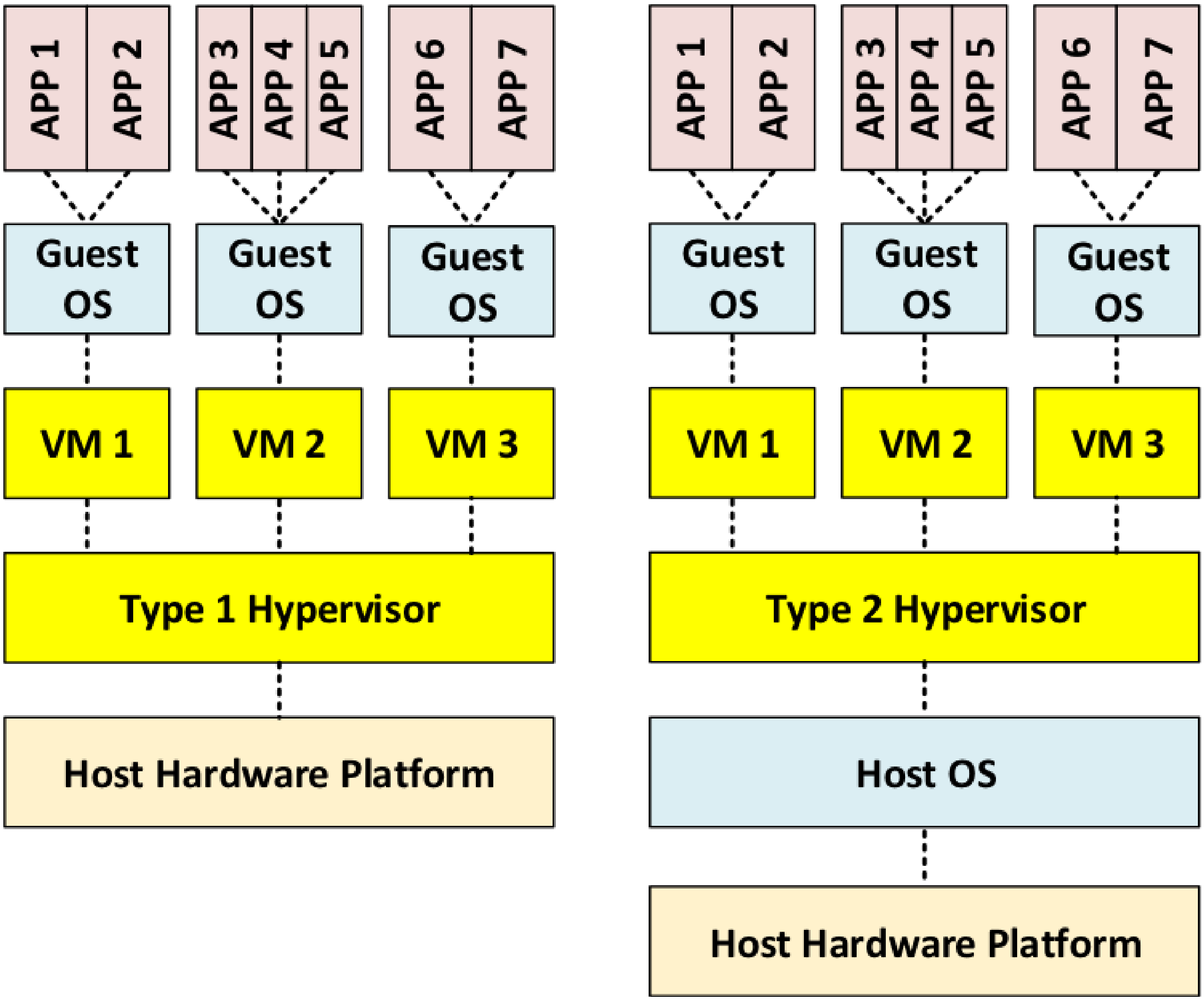


What is Virtualization?

In layman words, Virtualization enables users to disjoint operating systems from the underlying hardware, i.e., users can run multiple operating systems such as Windows, Linux, on a single physical machine at the same time. Such operating systems are known as guest Oses (operating systems).Virtualization deploys software that makes an abstraction layer across computer hardware, letting the hardware components such as processors, memory, storage etc of a particular computer to be segmented into several virtual elements (also known as virtual machines).

Moreover, in today’s’ time, virtualization is globally adopted in enterprise IT architecture and drives cloud computing economics. Essentially, Virtualization allows cloud providers to deliver users along with existing physical computer hardware.

As a simple process, it enables cloud users to purchase only necessary computing resources when they actually need it, and to sustain those resources cost-effectively when the workload expands.



Some terminologies associated with Virtualization

1. **Hypervisor:** It is an operating system, performing on the actual hardware, the virtual counterpart is a subpart of this operating system in the form of a running process. Hypervisors are observed as Domain 0 or Dom0.
2. **Virtual Machine (VM):**It is a virtual computer, executing underneath a hypervisor.
3. **Container:** Some light-weighted VMs that are subpart of the same operating system instance as its hypervisor are known as containers. They are a group of processes that runs along with their corresponding namespace for process identifiers.

4. **Virtualization Software:** Either be a piece of a software application package or an operating system or a specific version of that operating system, this is the software that assists in deploying the virtualization on any computer device.
5. **Virtual Network:** It is a logically separated network inside the servers that could be expanded across multiple servers.

1. Virtualization as a Concept of Cloud Computing

In the context of cloud computing, virtualization is a technique that makes a virtual ecosystem of storage devices and the server OS.

- In that case, virtualization enables users to use various machines that share one particular physical instance of any resource.
- Cloud virtualization transforms the traditional computing methods such that the workload management is more efficient, economic and scalable.
- Virtualization concerning Cloud Computing is being unified swiftly, and advancing the conventional course of computing such as virtualization is helping in the sharing of appli- cations across a network thread of several enterprises and active users.

Since cloud computing is being considered as a service or an application, assisting a virtualized ecosystem that could either be private or public, so with virtualization, resources could be escalated, reducing the necessity for a physical system.

Besides that, in order to understand how virtualization works in cloud computing, check the video below that explains how virtualization is still the elemental component of cloud strategy.

1. Characteristics of Virtualization

1. **Resource Distribution:** Either be a single computer or a network of connected servers, virtualization allows users to make a unique computer environment from one host machine that lets users to restrict the participants as active users, scale down power consumption and easy control.

2. **Isolation:** Virtualization software involves self-contained virtual machines, these VMs give guest users (not an individual but a number of instances as applications, operating systems, and devices) an isolated online, virtual environment. This online environment not only defends sensitive knowledge but also allows guest users to remain-connected.

3. **Availability:** Virtualization software provides various number of features that users won't obtain at physical servers, these features are beneficial in increasing uptime, availability, fault tolerance, and many more. These features help users to avoid downtime that subverts the users' efficiencies and productivities and also generates security threats and safety hazards.

4. **Aggregation:** Since virtualization allows several devices to split resources from a single machine, so it can be deployed to join multiple devices into a single potent host. In addition to that, aggregation also demands for cluster management software in

order to connect a homogeneous group of computers or servers collectively for making a unified resource center.

5. **Authenticity and security:** At ease, virtualization platforms assure the continuous uptime by balancing load automatically that runs an excessive number of servers across multiple host machines in order to prevent interruption services.

4. **Benefits of Virtualization**

There are following benefits of virtualization in cloud computing;

Security:

Security has been the advantageous concern for adopting virtualization .The security is served through firewalls that prevent from any unreliable access and preserve the data safe and confidential.

In addition to that,

The firewalls provide extra security from any sort of cyber threats and virus attacks,

- The protocols consist of end to end encryption, saving data automatically from other risky threads, and
- Users can virtualize their data and make backups of the same data on another server when needed.

2. Flexible Operations:

With the deployment of virtualization, users can work efficiently as the working process is very streamlined and agile. Presently, the employed network switch is easy to use, flexible and saves time. Virtualization is also helpful in troubleshooting technical errors, occurring in any of the connected devices. It eradicates the issues of retaining or recovering lost data due to corrupted or crashed devices, and therefore promotes ROI and saves time.

3. Economical:

This is the most prime reason to choose virtualization rapidly as with this technique companies can manage additional expenditure on physical devices and servers.

Being active with a virtual environment, data can be gathered on virtual servers. It also reduces the rigorous use of electricity (that has been a concern if several physical devices and services are being used at the same time), lowering bills while executing the numerous components of an operating system and applications over the users and companies network.

4. Flexible data transfer:

The data can be transferred to virtual servers anytime and also be retrieved due to this users or cloud providers need not to waste time in finding out hard drives to discover data.

With the implementation of virtualization, it has become easy to allocate the required data and transfer them to the appropriate authorities. Moreover, there is no limitation of data transfer and can be transferred to a far distance with minimal charges.

5. Remove system failure risks:

While performing any function, it often happens that the system might malfunction in critical timing such that this system failure could be adverse for a company’s resources and also deteriorate its reputation. This system failure can be protected with virtualization as users could perform the same task simultaneously over multiple devices, and the accumulated data can also be retrieved anytime with any device.

Implementation Levels of Virtualization

- Virtualization is a computer architecture technology by which mul- tiple virtual machines(VMs) are multiplexed in the same hardware machine.
- After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware

independent of the host OS

- done by adding additional software, called a virtualization layer
- This virtualization layer is known as hypervisor or virtual machine monitor (VMM)

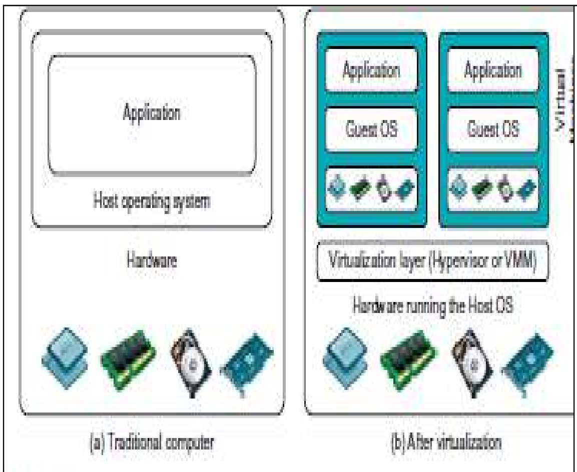
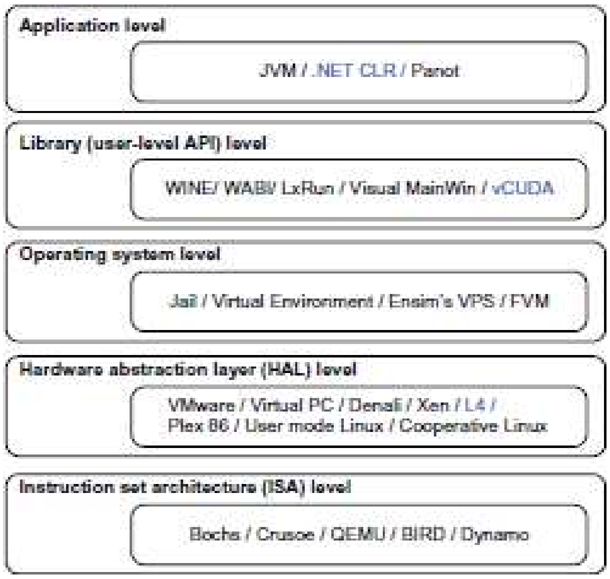


FIGURE 3.1

The architecture of a computer system before and after virtualization, where VMM stands for virtual mac monitor.


Application level
JVM / .NET CLR / Panot
Library (user-level API) level
WINE/ WABI/ LxRun / Visual MainWin / vCUDA
Operating system level
Jail / Virtual Environment / Ensim's VPS / FVM
Hardware abstraction layer (HAL) level
VMware / Virtual PC / Denali / Xen / L4 / Plex 86 / User mode Linux / Cooperative Linux
Instruction set architecture (ISA) level
Bochs / Crusoe / QEMU / BIRD / Dynamo

- The main function of the software layer for virtualization is to virtu- alize the physical hardware of a o s machine into virtual resources to be used by the VMs.
- Common virtualization layers include the instruction set architecture (ISA) level, hardware level, operating system level, library support level, and application level.

Instruction Set Architecture Level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the hostmachine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation. With this approach, it is possible to run a large amount of legacy binary code writ- ten for various processors on any given new hardware host machine.
- Instruction set emulation leads to virtual ISAs created on any hardware machine. The basic emulation method is through code

interpreter program interprets the sourceinstructions to target instructions one by one. One source instruction may require tens or hundreds of native target instructions to perform its function. Obviously, this process is relatively slow. For better performance, dynamic binary translation is desired.

- This approach translates basic blocks of dynamic source instructions to target instructions. The basic blocks can also be extended to program traces or super blocks to increase translation efficiency.
- Instruction set emulation requires binary translation and optimization. A virtual instruction set architecture (V-ISA) thus requires adding a pro- cessor-specific software translation layer to the compiler.

Hardware Abstraction Level

- Hardware-level virtualization is performed right on top of the bare hardware.
- This approach generates a virtual hardware environment for a VM.
- The process manages the underlying hardware through virtualization. The idea is to virtualize a computer’s resources, such as its processors, memory, and I/O devices.
- The intention is to upgrade the hardware utilization rate by multiple users concurrently. Theidea was implemented in the IBM VM/370 in the 1960s.
- More recently, the Xen hypervisor has been applied to virtualize x86- based machines to runLinux or other guest OS applications.

Operating System Level

- This refers to an abstraction layer between traditional OS and user applications.
- OS-level virtualization creates isolated containers on a single phys- ical server and the OS instances to utilize the hardware and soft- ware in data centers.
- The containers behave like real servers.

- OS-level virtualization is commonly used in creating virtual hosting en- vironments to allocatehardware resources among a large number of mu- tually distrusting users.
- It is also used, to a lesser extent, in consolidating server hardware by moving services onseparate hosts into containers or VMs on one server.

Library Support Level

- Most applications use APIs exported by user-level libraries rather than using lengthy systemcalls by the OS.
- Since most systems provide well-documented APIs, such an interface becomesanoth
- Virtualization with library interfaces is possible by controlling the communication linkbetween applications and the rest of a system through API hooks.

User-Application Level

- Virtualization at the application level virtualizes an application as a VM.
- On a traditional OS, an application often runs as a process. Therefore, application-level virtualization is also known as process-level virtualiza- tion.
- The most popular approach is to deploy high level language (HLL)VMs. In this scenario, the virtualization layer sits as an application program on top of the operating system,
- The layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.
- Any program written in the HLL and compiled for this VM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are two good examples of this class of VM.

Virtualization Structures/Tools and Mechanisms

- The layer between real hardware and traditional operating sys- tems. This layer is commonlycalled the Virtual Machine Monitor

(VMM)

- three requirements for a VMM
- a VMM should provide an environment for programs which is es- sentially identical to theoriginal machine
- programs run in this environment should show, at worst, only minor decreases in speed
- VMM should be in complete control of the system resources.
- VMM includes the following aspects:

(1) The VMM is responsible for allocating hardware resources for programs;

(2) it is not possible for a program to access any resource not explicitly allocated to it;

(3) it is possible under certain circumstances for a VMM to regain control of resourcesalready allocated.

Virtualization Support at the OS Level

- Why OS-Level Virtualization? :
  - it is slow to initialize a hardware-level VM because each VM creates its ownimage from scratch.
- OS virtualization inserts a virtualization layer inside an operating system to partition amachine's physical resources.
- It enables multiple isolated VMs within a single operating system kernel.
- This kind of VM is often called a virtual execution environment (VE), Virtual PrivateSystem (VPS), or simply container
- The benefits of OS extensions are twofold:

  (1) VMs at the operating system level have minimal startup/shutdown costs, lowresource requirements, and high scalability;

  (2) for an OS-level VM, it is possible for a VM and its host environment tosynchronize state changes when

necessary**.**

Middleware Support for Virtualization

- Library-level virtualization is also known as user-level Application Binary Interface(ABI) or API emulation.
- This type of virtualization can create execution environments for running alien programson a platform

Hypervisor and Xen Architecture

- The hypervisor software sits directly between the physical hardware and its OS.
- This virtualization layer is referred to as either the VMM or the hypervisor

Xen Architecture

- Xen is an open source hypervisor program developed by Cambridge University.
- Xen is a microkernel hypervisor

- The core components of a Xen system are the hypervisor, kernel, and applications
- The **guest OS**, which has control ability, is called **Domain 0**, and the others are called

  Domain U
- Domain 0 is designed to access hardware directly and manage devices

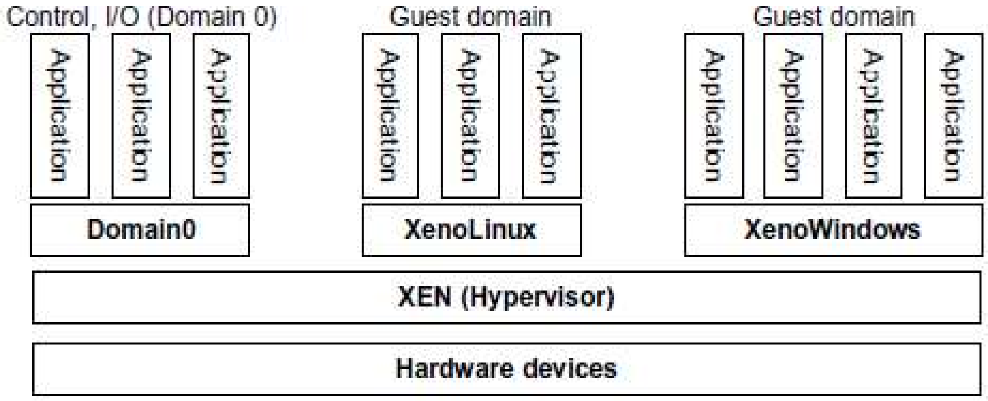


FIGURE 3.5

The Xen architecture's special domain 0 for control and I/O, and several guest domains for user applications.

(Courtesy of P. Barham, et al. [7])

- VM state is akin to a tree: the current state of the machine is a point that progresses monotonically as the software executes.
- VMs are allowed to roll back to previous states in their execution (e.g., to fix configuration errors) or rerun from the same point many times

Full virtualization

- Full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software
- **VMware** puts the **VMM at Ring 0** and the **guest OS at Ring 1**.
- The VMM scans the instruction stream and identifies the privileged, control- and behavior-sensitive instructions.
- When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions.
- The method used in this emulation is called binary translation.

- Therefore, full virtualization combines binary translation and direct execution.

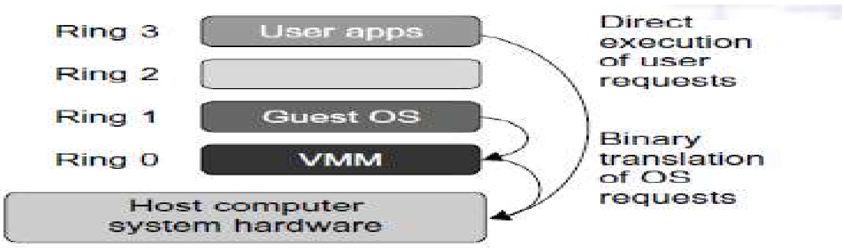

Ring 3
Ring 2
Ring 1
Ring 0
User apps
Guest OS
VMM
Host computer system hardware
Direct execution of user requests
Binary translation of OS requests

FIGURE 3.6

Indirect execution of complex instructions via binary translation of guest OS requests using the VMM plus direct execution of simple instructions on the same host.

(Courtesy of VM Ware [71])

Para-Virtualization

- Para-virtualization needs to modify the guest operating systems
- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications

Virtualization of CPU

- A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode.
- Hardware-Assisted CPU Virtualization: This technique attempts to simplify virtualization because full or paravirtualization is complicated

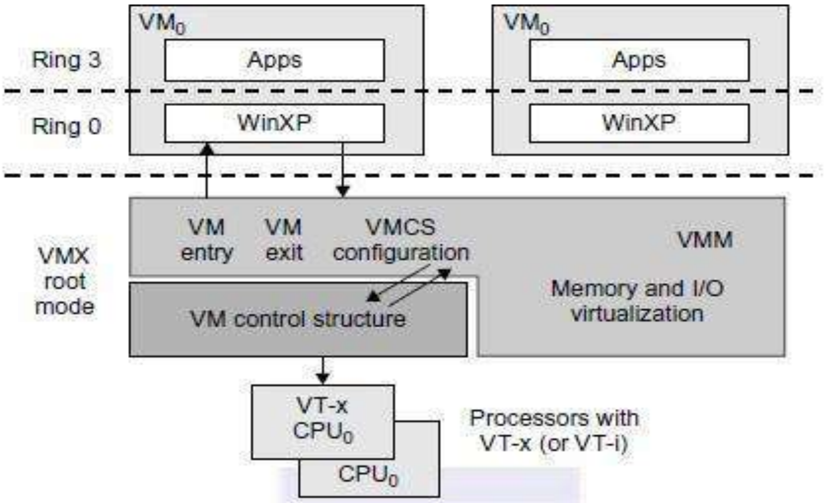


FIGURE 3.11

Intel hardware-assisted CPU virtualization.

(Modified from [68], Courtesy of Lizhong Chen, USC)

Memory Virtualization

- **Memory Virtualization** :the operating system maintains mappings of virtual memory to machine memory using page table
- All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance
- Two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory.

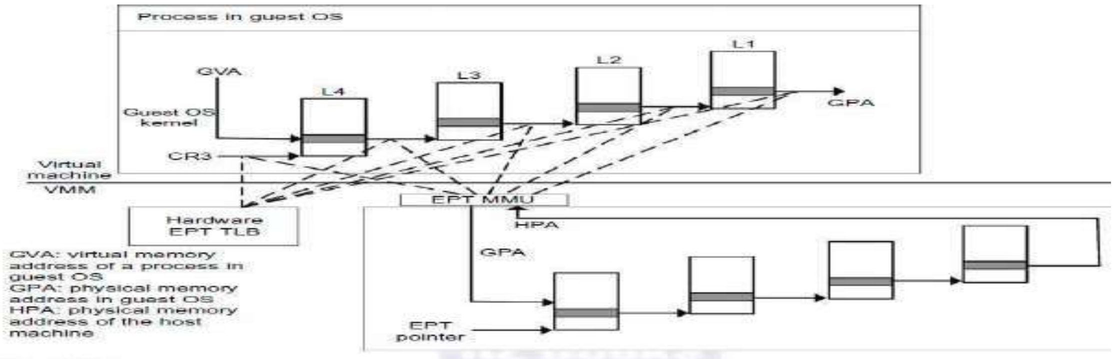

Process in guest OS
GVA
Guest OS
kernel
CR3
L4
L3
L2
L1
GPA
Virtual
machine
VMM
EPT MMU
Hardware
EPT TLB
HPA
GPA
EPT
pointer
GVA: virtual memory address of a process in guest OS
GPA: physical memory address in guest OS
HPA: physical memory address of the host machine

FIGURE 3.13

Memory virtualization using EPT by Intel (the EPT is also known as the shadow page table [68]).

I/O Virtualization

- I/O Virtualization managing the routing of I/O requests between virtual devices and the shared physical hardware
- managing the routing of I/O requests between virtual devices and the shared physical hardware
- Full device emulation emulates well-known, real-world devices All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device
- Two-stage mapping process should be maintained by the guest OS and the VMM,

  respectively: virtual memory to physical memory and physical memory to machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory.

Virtualization in Multi-Core Processors

- **Muti-core virtualization** has raised some new **challenges**
- **Two difficulties**: Application programs must be parallelized to use all cores fully, andsoftware must explicitly
- Assign tasks to the cores, which is a very complex problem
- The **first challenge**, new programming models, languages, and libraries are needed tomake parallel programming easier.
- The **second challenge** has spawned research involving scheduling algorithms andresource management policies
- **Dynamic heterogeneity** is emerging to mix the fat CPU core and thin GPU cores on thesame chip

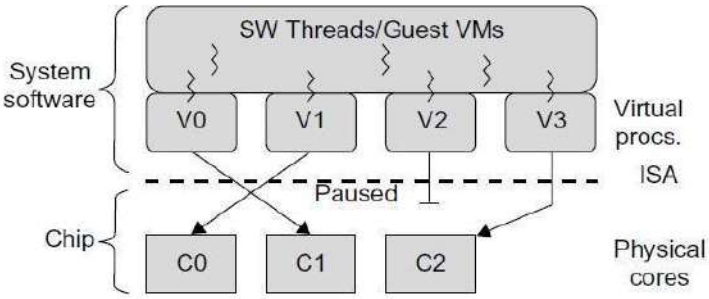


FIGURE 3.16

Multicore virtualization method that exposes four VCPUs to the software, when only three cores are actually present.

(*Courtesy of Wells, et al. [74]*)

- In many-core chip multiprocessors (CMPs)
- Instead of supporting **time-sharing jobs** on one or a few cores, use the abundant cores

  space-sharing, where single-threaded or multithreaded jobs are simultaneouslyassigned to separate groups of cores

Physical versus Virtual Clusters

- Virtual clusters are built with VMs installed at distributed servers from one or morephysical clusters.
- Assign tasks to the cores, which is a very complex problem
- Fast deployment
- High-Performance Virtual Storage
- reduce duplicated blocks

Virtual Clusters and Resource Management

- **Four ways to manage a virtual cluster**.
- First, you can use a **guest-based manager**, by which the cluster manager resides on a guest system.
- The **host-based manager** supervises the guest systems and can restart the guest system on another physical machine
- Third way to manage a virtual cluster is to use an **independent cluster**

manager on both the host and guest systems.

- Finally, use an **integrated cluster** on the guest and host systems.
- This means the manager must be designed to distinguish between virtualized resources and physical resources

Virtualization for data-center automation

- **Data-center automation** means that huge volumes of hardware, software, and database resources in these data centers can be allocated dynamically to millions of Internet users simultaneously, with guaranteed QoS and cost-effectiveness.
- This **automation** process is triggered by the growth of virtualization products and cloud computing services.
- The latest virtualization development highlights high availability (HA), backup services, workload balancing, and further increases in client bases.

Server Consolidation in Data Centers

- heterogeneous workloads -chatty workloads and non interactive workloads
- **Server consolidation** is an approach to improve the low utility ratio of hardware resources by reducing the number of physical servers

Virtual Storage Management

- storage virtualization has a different meaning in a system virtualization environment
- system virtualization, virtual storage includes the storage managed by VMMs

  and guest OS es data stored in this environment can be classified into two categories: VM images and application data.

Cloud OS for Virtualized Data Centers

- Data centers must be virtualized to serve as cloud providers
- Eucalyptus for Virtual Networking of Private Cloud :
- Eucalyptus is an **open source software system** intended mainly for supporting

  Infrastructure as a Service (IaaS) clouds
- The system primarily supports **virtual networking** and the management of **VMs**;
- virtual storage is not supported.
- Its purpose is to build **private clouds**

Three resource managers

- Instance Manager
- Group Manager
- Cloud Manager

UNIT – 3

INFRASTRUCTURE AS A SERVICE (IAAS) & PLATFORM (PAAS)

- Cloud computing is an emerging research infrastructure that builds on the achievements of different research areas, such as service-oriented architecture (SOA), grid computing, and virtualization technology.

- It offers infrastructure as a service that is based on pay-as-you-use and on-demand computing models to the end users (exactly the same as a public utility service like electricity, water, gas, etc.). This service is referred to as Infrastructure as a Service (IaaS).

- To provide this cloud computing service, the provisioning of the cloud infrastructure in data centers is a prerequisite. However, the provisioning for systems and applications on a large number of physical machines is traditionally a time-consuming process with low assurance on deployment's time and cost.

- There are two core services that enable the users to get the best out of the IaaS model in public and private cloud setups. These services are named virtual machine provisioning and migration services.

- Now, with the emergence of virtualization technology and the cloud computing IaaS model, it is just a matter of minutes to achieve the same task.

- All you need is to provision a virtual server through a self-service interface with small steps to get what you desire with the required specifications—whether you are provisioning this machine in a public cloud like Amazon Elastic Compute Cloud (EC2) or using a virtualization management software package or a private cloud management solution installed at your data center in order to provision the virtual machine inside the organization and within the private cloud setup.

- There was a need for performing a server's upgrade or performing maintenance tasks, you would exert a lot of time and effort, because it is an expensive operation to maintain or upgrade a main server that has lots of applications and users.

- Now, with the advance of the revolutionized virtualization technology and migration services associated with hypervisors' capabilities, these tasks (maintenance, upgrades, patches, etc.) are very easy and need no time to accomplish.

- Provisioning a new virtual machine is a matter of minutes, saving lots of time and effort. Migrations of a virtual machine is a matter of milliseconds: saving time, effort, making the service alive for customers, and achieving the SLA/ SLO agreements and quality-of-service (QoS) specifications required.

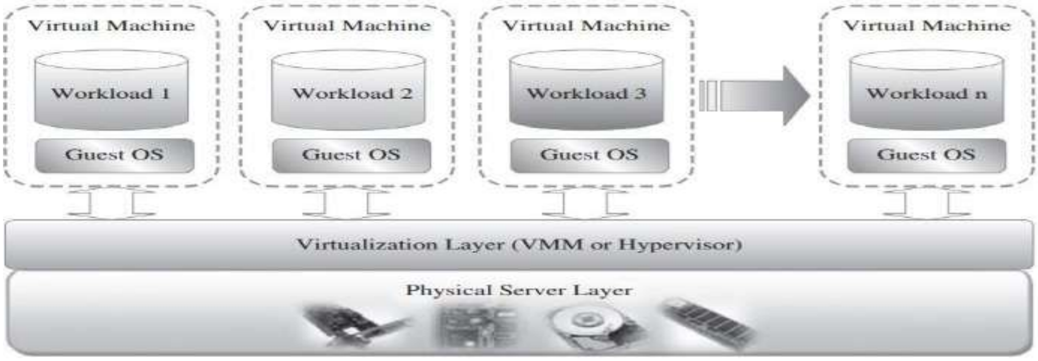


FIGURE 5.2. A layered virtualization technology architecture.

INFRASTRUCTURE AS A SERVICE (IAAS) & PLATFORM (PAAS)

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:

.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., load- balancers, firewalls).

.iv Choice of virtualization platform and operating systems and

.v 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 at anytime 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 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. 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 equal to 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 recovery of 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 specialized software- 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, such as 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; automatic scaling(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 architecture is 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.

Public Cloud and Infrastructure Services

- Public cloud or external cloud describes cloud computing in a traditional mainstream sense, whereby resources are dynamically provisioned via publicly accessible Web applications/Web services (SOAP or RESTful interfaces) from an off-site third-party provider, who shares resources and bills on a fine-grained utility computing basis, the user pays only for the capacity of the provisioned resources at a particular time.

- Amazon Elastic Compute Cloud (EC2) is an IaaS service that provides elastic compute capacity in the cloud. These services can be leveraged via Web services (SOAP or REST), a Web-based AWS (Amazon Web Service) management console, or the EC2 command line tools. The Amazon service provides hundreds of pre-made AMIs (Amazon Machine Images) with a variety of operating systems (i.e., Linux, OpenSolaris, or Windows) and pre-loaded software. The market now bristles with lots of competition like GoGrid, Joyent Accelerator, Rackspace, AppNexus, FlexiScale, and Manjrasoft Aneka.

Private Cloud and Infrastructure Services

A private cloud aims at providing public cloud functionality, but on private resources, while maintaining control over an organization's data and resources to meet security and governance's requirements in an organization. Private cloud exhibits a highly virtualized cloud data center located inside your organization's firewall. It may also be a private space dedicated for your company within a cloud vendor's data center designed to handle the organization's workloads.

Private clouds exhibit the following characteristics:

- Allow service provisioning and compute capability for an organization's users in a self-service manner.

- Automate and provide well-managed virtualized environments.
- Optimize computing resources, and servers' utilization.
- Support specific workloads.

Distributed Management of Virtualization

Virtualization's benefits bring their own challenges and complexities presented in the need for a powerful management capabilities. That is why many commercial, open source products and research projects such as OpenNebula, IBM Virtualization Manager, Joyent, and VMware DRS are being developed to dynamically provision virtual machines, utilizing the physical infrastrcture. There are also some commercial and scientific infrastructure cloud computing initiatives, such as Globus VWS, Eucalyptus and Amazon, which provide remote interfaces for controling and monitoring virtual resources.

One more effort in this context is the RESERVOIR initiative, in which grid interfaces and protocols enable the required interoperability between the clouds or infrastructure's providers.

High Availability

High availability is a system design protocol and an associated implementation that ensures a certain absolute degree of operational continuity during a given measurement period. Availability refers to the ability of a user's community to access the system—whether for submitting new work, updating or altering existing work, or collecting the results of the previous work.

Cloud and Virtualization Standardization Efforts

Standardization is important to ensure interoperability between virtualization mangement vendors, the virtual machines produced by each one of them, and cloud computing. In the past few years, virtualization standardization efforts led by the Distributed Management Task Force (DMTF) have produced standards for almost all the aspects of virtualization technology.

DMTF initiated the VMAN (Virtualization Management Initiative), which delivers broadly supported interoperability and portability standards for managing the virtual computing lifecycle. VMAN's OVF (Open Virtualization Format) in a collaboration between industry key players: Dell, HP, IBM, Microsoft, XenSource, and Vmware.

OCCI and OGF

Open Grid Forum (OGF) organizing an official new working group to deliver a standard API for cloud IaaS, the Open Cloud Computing Interface Working Group (OCCIWG). This group is dedicated for delivering an API specification for the remote management of cloud computing's infrastructure and for allowing the development of interoperable tools for common tasks including deployment, autonomic scaling, and monitoring. The scope of the specification will be covering a high-level functionality required for managing the life-cycle virtual machines (or workloads), running on virtualization technologies (or containers), and supporting service elasticity. The new API for interfacing "IaaS" cloud computing facilities will allow

- **Consumers** to interact with cloud computing infrastructure on an ad hoc basis.
- **Integrators** to offer advanced management services.
- **Aggregators** to offer a single common interface to multiple providers. Providers to offer a standard interface that is compatible with the available tools.
- **Vendors** of grids/clouds to offer standard interfaces for dynamically scalable service's delivery in their products.

Virtual machines provisioning and Migration services

Virtual machines Provisioning Process

Typical life cycle of VM and its major possible states of operation, which make the management and automation of VMs in virtual and cloud environments easier Process & Steps to Provision VM. Here, we describe the common and normal steps of provisioning a virtual server:

- Firstly, you need to select a server from a pool of available servers (physical servers with enough capacity) along with the appropriate OS template you need to provision the virtual machine.
- Secondly, you need to load the appropriate software (operating system you selected in the previous step, device drivers, middleware, and the needed applications for the service required).
- Thirdly, you need to customize and configure the machine (e.g., IP address, Gateway) to configure an associated network and storage resources.
- Finally, the virtual server is ready to start with its newly loaded software. Typically, these are the tasks required or being performed by an IT or a data center's specialist to provision a particular virtual machine.

virtual machines can be provisioned by manually installing an operating system, by using a preconfigured VM template, by cloning an existing VM, or by importing a physical server or a virtual server from another hosting platform. Physical servers can also be virtualized and provisioned using P2V (physical to virtual) tools and techniques (e.g., virt-p2v).

After creating a virtual machine by virtualizing a physical server, or by building a new virtual server in the virtual environment, a template can be created out of it. Most virtualization management vendors (VMware, XenServer, etc.) provide the data center's administration with the ability to do such tasks in an easy way.

Provisioning from a template is an invaluable feature, because it reduces the time required to create a new virtual machine. Administrators can create different templates for different purposes. For example, you can create a Windows 2003 Server template for the finance department, or a Red Hat Linux template for the engineering department.

This enables the administrator to quickly provision a correctly configured virtual server on demand. This ease and flexibility bring with them the problem of virtual machine's sprawl, where virtual machines are provisioned so rapidly that documenting and managing the virtual machine's life cycle become a challenge

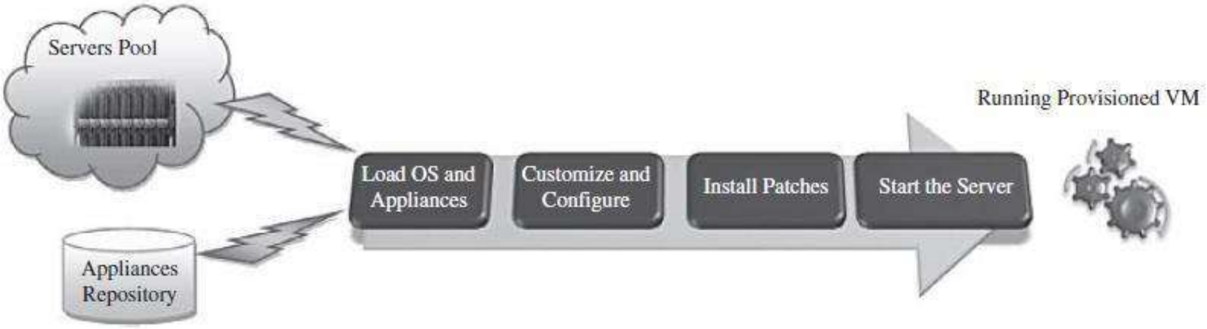


FIGURE 5.4. Virtual machine provision process.

VIRTUAL MACHINE MIGRATION SERVICES

Migration service, in the context of virtual machines, is the process of moving a virtual machine from one host server or storage location to another; there aredifferent techniques of VM migration, hot/life migration, cold/regular migration, and live storage migration of a virtual machine. In this process, all key machine components, such as CPU, storage disks, networking, and memory, are completely virtualized, thereby facilitating the entire state of a virtual machine to be captured by a set of easily moved data files. Here are some of the migration's techniques that most virtualization tools provide as a feature.

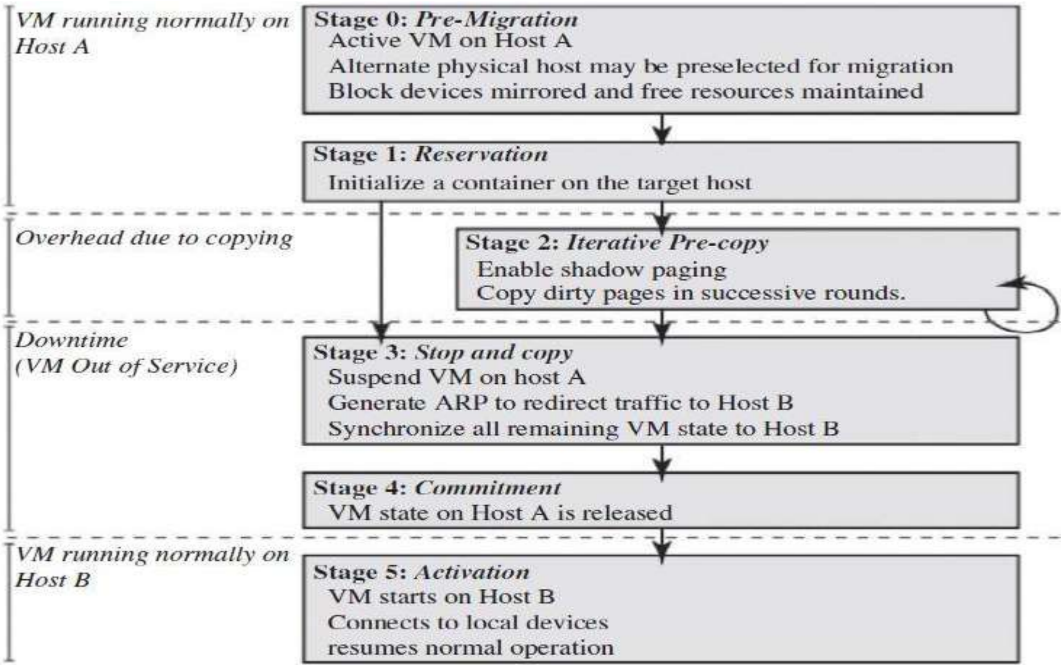
Migrations Techniques

Live Migration and High Availability.

Live migration (which is also called hot or real-time migration) can be defined as the movement of a virtual machine from one physical host to another while being powered on. When it is properly carried out, this process takes place without any noticeable effect from the end user's point of view (a matter of milliseconds). One of the most significant advantages of live migration is the fact that it facilitates proactive maintenance in case of failure, because the potential problem can be resolved before the disruption of service occurs. Live migration can also be used for load balancing in which work is shared among computers in order to optimize the utilization of available CPU resources.

Live Migration Anatomy, Xen Hypervisor Algorithm.

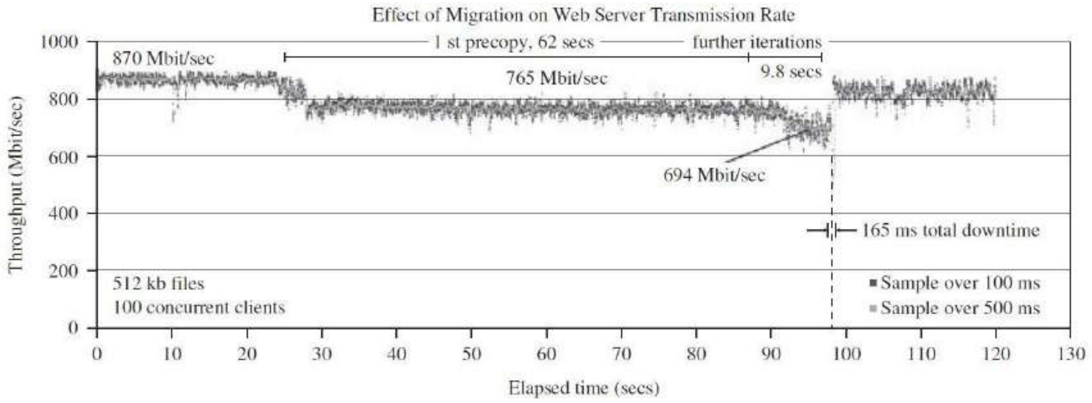
In this section we will explain live migration's mechanism and how memory and virtual machine states are being transferred, through the network, from one host A to another host B, the Xen hypervisor is an example for this mechanism. The logical steps that are executed when migrating an OS are summarized in the diagram below In this research, the migration process has been viewed as a transactional interaction between the two hosts involved:


VM running normally on Host A
Stage 0: Pre-Migration
Active VM on Host A
Alternate physical host may be preselected for migration
Block devices mirrored and free resources maintained
Stage 1: Reservation
Initialize a container on the target host
Overhead due to copying
Stage 2: Iterative Pre-copy
Enable shadow paging
Copy dirty pages in successive rounds.
Downtime (VM Out of Service)
Stage 3: Stop and copy
Suspend VM on host A
Generate ARP to redirect traffic to Host B
Synchronize all remaining VM state to Host B
Stage 4: Commitment
VM state on Host A is released
VM running normally on Host B
Stage 5: Activation
VM starts on Host B
Connects to local devices
resumes normal operation

Live Migration Effect on a Running Web Server.

Clark et al. Did evaluate the above migration on an Apache 1.3 Web server; this served static content at a high rate as shown in diagram below. The throughput is achieved when continuously serving a single 512-kB file to a set of one hundred concurrent clients. The Web server virtual machine has a memory allocation of 800 MB. At the start of the trace, the server achieves a consistent throughput of approximately 870 Mbit/sec.

Migration starts 27 sec into the trace, but is initially rate-limited to 100 Mbit/sec (12% CPU), resulting in server's throughput drop to 765 Mbit/sec. This initial low-rate pass transfers 776 MB and lasts for 62 sec. At this point, the migration's algorithm, described in Section 5.4.1, increases its rate over several iterations and finally suspends the VM after a further 9.8 sec. The final stop-and-copy phase then transfers the remaining pages, and the Web server resumes at full rate after a 165-msec outage. This simple example demonstrates that a highly loaded server can be migrated with both controlled impact on live services and a short downtime. However, the working set of the server, in this case, is rather small. So, this should be expected as a relatively easy case of live migration.



Live Migration Vendor Implementations Examples.

There are lots of VM management and provisioning tools that provide the live migration of VM facility, two of which are VMware VMotion and Citrix XenServer “XenMotion.”

VMware Vmotion.

This allows users to automatically optimize and allocate an entire pool of resources for maximum hardware utilization, flexibility, and availability and perform hardware’s maintenance without scheduled downtime along with migrating virtual machines away from failing or underperforming servers.

Citrix XenServer XenMotion.

This is a nice feature of the Citrix XenServer product, inherited from the Xen live migrate utility, which provides the IT administrator with the facility to move a running VM from one XenServer to another in the same pool without interrupting the service (hypothetically for zero-downtime server maintenance, which actually takes minutes), making it a highly available service. This also can be a good feature to balance the workloads on the virtualized environment.

Regular/Cold Migration.

Cold migration is the migration of a powered-off virtual machine. With cold migration, you have the option of moving the associated disks from one data store to another. The virtual machines are not required to be on a shared storage. It’s important to highlight that the two main differences between live migration and cold migration are that live migration needs a shared storage for virtual machines in the server’s pool, but cold migration does not and also in live migration for a virtual machine between two hosts, there would be certain CPU compatibility checks to be applied; while in cold migration this checks do not apply.

The cold migration process is simple to implement and it can be summarized as follows:

- The configuration files, including the NVRAM file (BIOS settings), log files, as well as the disks of the virtual machine, are moved from the source host to the destination host’s associated storage area.
- The virtual machine is registered with the new host.

- After the migration is completed, the old version of the virtual machine is deleted from the source host.

Live Storage Migration of Virtual Machine.

This kind of migration constitutes moving the virtual disks or configuration file of a running virtual machine to a new data store without any interruption in the availability of the virtual machine's service.

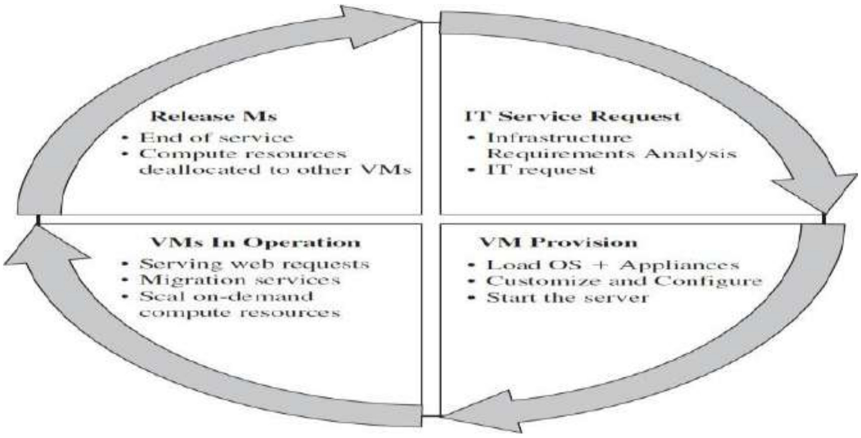
Migration of Virtual Machines to Alternate Platforms

One of the nicest advantages of having facility in data center's technologies is to have the ability to migrate virtual machines from one platform to another. There are a number of ways for achieving this, such as depending on the source and target virtualization's platforms and on the vendor's tools that manage this facility—for example, the VMware converter that handles migrations between ESX hosts; the VMware server; and the VMware workstation. The VMware converter can also import from other virtualization platforms, such as Microsoft virtual server machines.

VIRTUAL MACHINES PROVISIONING AND MANAGEABILITY

The typical life cycle of VM and its major possible states of operation, which make the management and automation of VMs in virtual and cloud environments easier than in traditional computing environments.

As shown in the diagram below the cycle starts by a request delivered to the IT department, stating the requirement for creating a new server for a particular service. This request is being processed by the IT administration to start seeing the servers' resource pool, matching these resources with the requirements, and starting the provision of the needed virtual machine. Once it is provisioned and started, it is ready to provide the required service according to an SLA, or a time period after which the virtual is being released; and free resources, in this case, won't be needed.



VM PROVISIONING AND MIGRATION IN ACTION

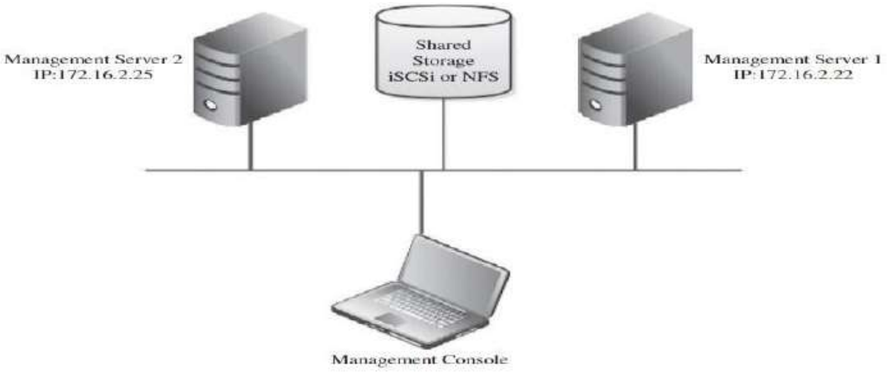
Now, it is time to get into business with a real example of how we can manage the life cycle, provision, and migrate a virtual machine by the help of one of the open source frameworks used to manage virtualized infrastructure.

Here, we will use ConVirt (open source framework for the management of open source virtualization like Xen and KVM known previously as XenMan). Deployment Scenario. ConVirt deployment consists of at least one ConVirt workstation, where ConVirt is installed and ran, which provides the main console for managing the VM life cycle, managing images, provisioning new VMs, monitoring machine resources, and so on.

There are two essential deployment scenarios for ConVirt:

- Basic configuration in which the Xen or KVM virtualization platform is on the local machine, where ConVirt is already installed.
- An advanced configuration in which the Xen or KVM is on one or more remote servers. The scenario in use here is the advanced one. In data centers, it is very common to install centralized management software (ConVirt here) on a dedicated machine for use in managing remote servers in the data center.

In our example, we will use this dedicated machine where ConVirt is installed and used to manage a pool of remote servers (two machines). In order to use advanced features of ConVirt (e.g., live migration), you should set up a shared storage for the server pool in use on which the disks of the provisioned virtual machines are stored.



Installation

The installation process involves the following:

- Installing ConVirt on at least one computer. See reference 28 for installation details.

- Preparing each managed server to be managed by ConVirt. See reference 28 for managed servers' installation details. We have two managing servers with the following Ips (managed server 1, IP:172.16.2.22; and managed server 2, IP:172.16.2.25) as shown in the deployment diagram (Figure 5.7).
- Starting ConVirt and discovering the managed servers you have prepared.

Notes

1 Try to follow the installation steps existing in reference 28 according to the distribution of the operating system in use. In our experiment, we use Ubuntu 8.10 in our setup.

2 Make sure that the managed servers include Xen or KVM hypervisors installed.

3 Make sure that you can access managed servers from your ConVirt management console through SSH.

Environment, Software, and Hardware: ConVirt 1.1, Linux Ubuntu 8.10, three machines, Dell core 2 due processor, 4G RAM.

Adding Managed Servers and Provisioning VM: Once the installation is done and you are ready to manage your virtual infrastructure, then you can start the ConVirt management console.

Select any of servers' pools existing (QA Lab in our scenario) and on its context menu, select "Add Server".

1 You will be faced with a message asking about the virtualization platform you want to manage (Xen or KVM).

2 Choose KVM, and then enter the managed server information and credentials (IP, username, and password).

3 Once the server is synchronized and authenticated with the management console, it will appear in the left pane/of the ConVirt.

4 Select this server, and start provisioning your virtual machine.

5 Fill in the virtual machine's information (name, storage, OS template, etc) then you will find it created on the managed server tree powered-off. Note: While provisioning your virtual machine, make sure that you create disks on the shared storage (NFS or iSCSi). You can do so by selecting the "provisioning" tab, and changing the VM_DISKS_DIR to point to the location of your shared NFS.

6 Start your VM (Figures 5.14 and 5.15), and make sure the installation media of the operating system you need is placed in drive, in order to use it for booting the new VM and proceed in the installation process; then start the installation process as shown in Figure 5.16.

7 Once the installation finishes, you can access your provisioned virtual machine from the console icon on the top of your ConVirt management console.

8 Reaching this step, you have created your first managed server and provisioned virtual machine. You can repeat the same procedure to add the second managed server in your pool to be ready for the next step of migrating one virtual machine from one server to the other.

9 To start the migration of a virtual machine from one host to the other, select it and choose a migrating virtual machine.

10 You will have a window containing all the managed servers in your data center. Choose one as a destination and start

11 Once the virtual machine has been successfully placed and migrated to the destination host, you can see it still living and working,

ON THE MANAGEMENT OF VIRTUAL MACHINES FOR CLOUD INFRASTRUCTURES

In 2006, Amazon started offering virtual machines (VMs) to anyone with a credit card for just $0.10/hour through its Elastic Compute Cloud (EC2) service. Although not the first company to lease VMs, the programmer-friendly EC2 Web services API and their pay-as-you-go pricing popularized the “Infrastructure as a Service” (IaaS) paradigm, which is now closely related to the notion of a “cloud.”

Following the success of Amazon EC2, several other IaaS cloud providers, or public clouds, have emerged—such as Elastic- Hosts, GoGrid, and FlexiScale—that provide a publicly accessible interface for purchasing and managing computing infrastructure that is instantiated as VMs running on the provider’s data center.

There is also a growing ecosystem of technologies and tools to build private clouds—where inhouse resources are virtualized, and internal users can request and manage these resources using interfaces similar or equal to those of public clouds—and hybrid clouds—where an organization’s private cloud can supplement its capacity using a public cloud.

THE ANATOMY OF CLOUD INFRASTRUCTURES

There are many commercial IaaS cloud providers in the market, such as those cited earlier, and all of them share five characteristics:

(i) They provide on-demand provisioning of computational resources.

(ii) they use virtualization technologies to lease these resources.

(iii) they provide public and simple remote interfaces to manage those resources

(iv) they use a pay-as-you-go cost model, typically charging by the hour

(v) they operate data centers large enough to provide a seemingly unlimited amount of resources to their clients (usually touted as “infinite capacity” or “unlimited elasticity”).

- Private and hybrid clouds share these same characteristics but, instead of selling capacity over publicly accessible interfaces, focus on providing capacity to an organization’s internal users.
- Virtualization technologies have been the key enabler of many of these salient characteristics of IaaS clouds by giving providers a more flexible and generic way of managing their resources. Thus, virtual infrastructure (VI) management—the management of virtual machines distributed across a pool of physical resources—becomes a key concern when building an IaaS cloud and poses a number of challenges.
- Virtual infrastructure management in private clouds has to deal with an additional problem: Unlike large IaaS cloud providers, such as Amazon, private clouds typically do not have enough resources to provide the illusion of “infinite capacity.” The immediate provisioning scheme used in public clouds, where resources are provisioned at the moment they are requested, is ineffective in private clouds.
- Several VI management solutions have emerged over time, such as platform ISF and VMware vSphere, along with open-source initiatives such as Enomaly Computing Platform and Ovirt.
- However, managing virtual infrastructures in a private/hybrid cloud is a different, albeit similar, problem than managing a virtualized data center, and existing tools lack several features that are required for building IaaS clouds.

Distributed Management of Virtual Machines

The first problem is how to manage the virtual infrastructures themselves. Although resource management has been extensively studied, particularly for job management in high-performance computing, managing VMs poses additional problems that do not arise when managing jobs, such as the need to set up custom software environments forVMs, setting up and managing networking for interrelated VMs, and reducing the various overheads involved in using VMs.

- Thus, VI managers must be able to efficiently orchestrate all these different tasks. The problem of efficiently selecting or scheduling computational resources is well known.
- However, the state of the art in VM-based resource scheduling follows a static approach, where resources are initially selected using a greedy allocation strategy, with minimal or no support for other placement policies.

- To efficiently schedule resources, VI managers must be able to support flexible and complex scheduling policies and must leverage the ability of VMs to suspend, resume, and migrate. This complex task is one of the core problems that the RESERVOIR(Resources and Services Virtualization without Barriers) project tries to solve.

Reservation-Based Provisioning of Virtualized Resources

A particularly interesting problem when provisioning virtual infrastructures is how to deal with situations where the demand for resources is known beforehand— for example, when an experiment depending on some complex piece of equipment is going to run from 2 pm to 4 pm, and computational resources must be available at exactly that time to process the data produced by the equipment. Commercial cloud providers, such as Amazon, have enough resources to provide the illusion of infinite capacity, which means that this situation is simply resolved by requesting the resources exactly when needed; if capacity is "infinite," then there will be resources available at 2 pm. On the other hand, when dealing with finite capacity, a different approach is needed. However, the intuitively simple solution of reserving the resources beforehand turns out to not be so simple, because it is known to cause resources to be underutilized, due to the difficulty of scheduling other requests around an inflexible reservation. VMs allow us to overcome the utilization problems typically associated with advance reservations and we describe Haizea, a VM-based lease manager supporting advance reservation along with other provisioning models not supported in existing IaaS clouds, such as best-effort provisioning.

Provisioning to Meet SLA Commitments

IaaS clouds can be used to deploy services that will be consumed by users other than the one that deployed the services. For example, a company might depend on an IaaS cloud provider to deploy three-tier applications (Web front-end, application server, and database server) for its customers. In this case, there is a distinction between the cloud consumer (i.e., the service owner) and the end users of the resources provisioned on the cloud (the service user).

Furthermore, service owners will enter into service-level agreements (SLAs) with their end users, covering guarantees such as the timeliness with which these services will respond. However, cloud providers are typically not directly exposed to the service semantics or the SLAs that service owners may contract with their end users. The capacity requirements are less predictable and more elastic.

The cloud provider's task is, therefore, to make sure that resource allocation requests are satisfied with specific probability and timeliness. These requirements are formalized in infrastructure SLAs between the service owner and cloud provider, separate from the high-level SLAs between the service owner and its end users.

RESERVOIR proposes a flexible framework where service owners may register service-specific elasticity rules and monitoring probes, and these rules are being executed to match environment conditions.

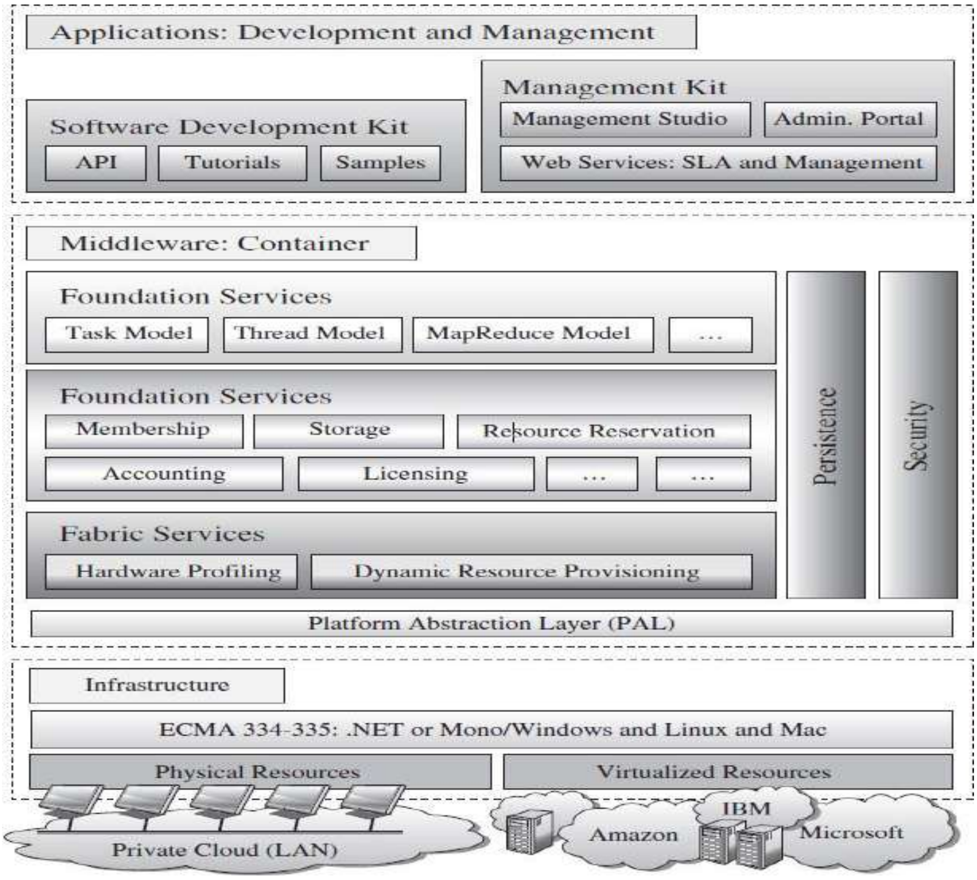
Elasticity of the application should be contracted and formalized as part of capacity availability SLA between the cloud provider and service owner. This poses interesting research issues on the IaaS side, which can be grouped around two main topics:

- SLA-oriented capacity planning that guarantees that there is enough capacity to guarantee service elasticity with minimal over-provisioning.
- Continuous resource placement and scheduling optimization that lowers operational costs and takes advantage of available capacity transparently to the service while keeping the service SLAs.

ANEKA—INTEGRATION OF PRIVATE AND PUBLIC CLOUDS

- Aneka is a software platform and a framework for developing distributed applications on the cloud. It harnesses the computing resources of a heterogeneous network of workstations and servers or data centers on demand.
- Aneka provides developers with a rich set of APIs for transparently exploiting these resources by expressing the application logic with a variety of programming abstractions. System administrators can leverage a collection of tools to monitor and control the deployed infrastructure.
- This can be a public cloud available to anyone through the Internet, a private cloud constituted by a set of nodes with restricted access within an enterprise, or a hybrid cloud where external resources are integrated on demand, thus allowing applications to scale. Diagram below provides a layered view of the framework.
- Aneka is essentially an implementation of the PaaS model, and it provides a runtime environment for executing applications by leveraging the underlying infrastructure of the cloud. Developers can express distributed applications by using the API contained in the Software Development Kit (SDK) or by porting existing legacy applications to the cloud.
- Such applications are executed on the Aneka cloud, represented by a collection of nodes connected through the network hosting the Aneka container.
- The container is the building block of the middleware and represents the runtime environment for executing applications; it contains the core functionalities of the system and is built up from an extensible collection of services that allow administrators to customize the Aneka cloud. There are three classes of services that characterize the container:
- Execution Services. They are responsible for scheduling and executing applications. Each of the programming models supported by Aneka defines specialized implementations of these services for managing the execution of a unit of work defined in the model.

- Foundation Services. These are the core management services of the Aneka container. They are in charge of metering applications, allocating resources for execution, managing the collection of available nodes, and keeping the services registry updated.
- Fabric Services: They constitute the lowest level of the services stack of Aneka and provide access to the resources managed by the cloud. An important service in this layer is the Resource Provisioning Service, which enables horizontal scaling3 in the cloud. Resource provisioning makes Aneka elastic and allows it to grow or to shrink dynamically to meet the QoS requirements of applications.
- Aneka also provides a tool for managing the cloud, allowing administrators to easily start, stop, and deploy instances of the Aneka container on new resources and then reconfigure them dynamically to alter the behavior of the cloud.



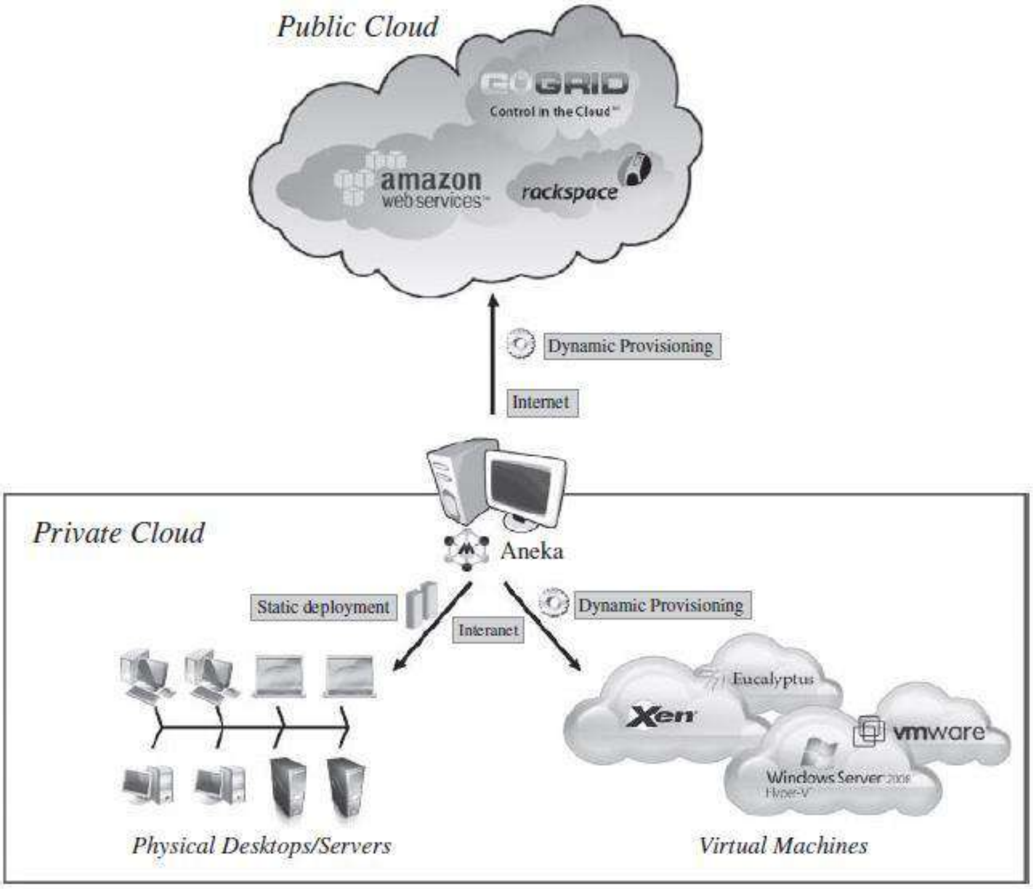
Public Cloud: Computing in which service provider makes all resources public over the internet. It is connected to the public Internet.

- Service provider serves resources such as virtual machines, applications, storage, etc to the general public over the internet.

- It may be free of cost or with minimal pay-per-usage. It is available for public display, Google uses the cloud to run some of its applications like google docs, google drive or YouTube, etc. It is the most common way of implementing cloud computing.
- External cloud service provider owns, operates and delivers it over the public network.
- It is best for the companies which need an infrastructure to accommodate large number of customers and working on projects which has diverse organisation i.e. research institution and NGO etc.

Private Cloud: Computing in which service provider makes all resources public over the internet. It only supports connectivity over the private network.

- It has only authentic users and single-occupant architecture. Google back-end data of the applications like Google Drive, Google docs or YouTube, etc is not available to the public, these types of data and applications run on a private cloud.
- The infrastructure and services are maintained and deployed over a private network; hardware and software are dedicated only to a private company i.e. members of the special entity.
- It is best for the companies which need an infrastructure which has high performance, high security and privacy due to it's best adaptability and flexibility.


Public Cloud
GOGRID
Control in the Cloud
amazon web services
rackspace
Dynamic Provisioning
Internet
Private Cloud
Aneka
Static deployment
Interanet
Dynamic Provisioning
Eucalyptus
Xen
vmware
Windows Server 2008 Hyper-V
Physical Desktops/Servers
Virtual Machines

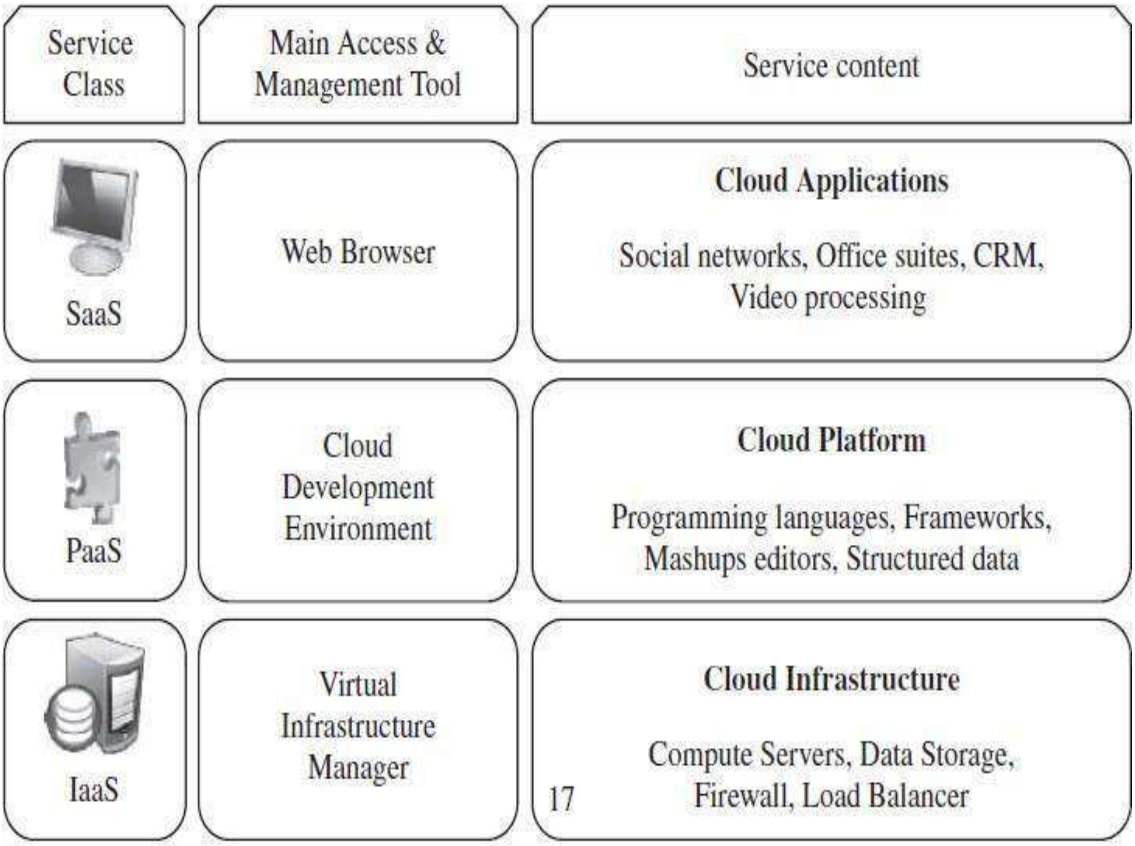
UNIT - 4

Software as a Service (SAAS) &Data Security in the Cloud

Software as a Service(SAAS)

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.



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

Deployment Models

Although cloud computing has emerged mainly from the appearance of public 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.

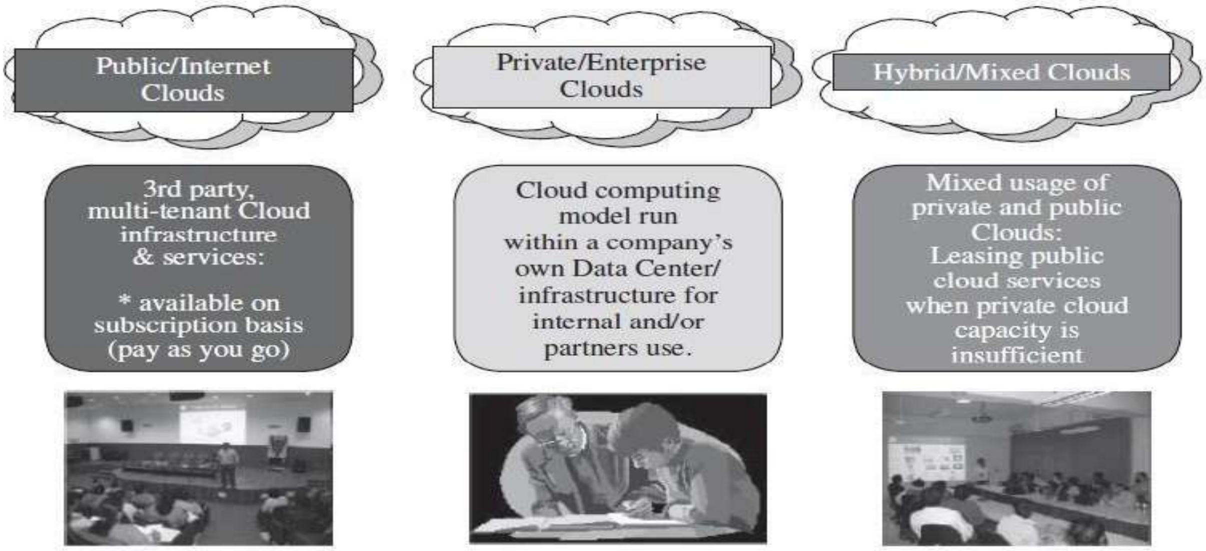

Public/Internet Clouds
Private/Enterprise Clouds
Hybrid/Mixed Clouds
3rd party, multi-tenant Cloud infrastructure & services:
* available on subscription basis (pay as you go)
Cloud computing model run within a company's own Data Center/ infrastructure for internal and/or partners use.
Mixed usage of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

- Self-service

- Per-usage metered and billed
- Elastic,
- Customizable.

- **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.

- **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.

- **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).

- **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.

Google APP Engine

The app engine is a Cloud-based platform, is quite comprehensive and combines infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).The app engine supports the delivery, testing and development of software on demand in a Cloud computing environment that supports millions of users and is highly scalable.

The company extends its platform and infrastructure to the Cloud through its app engine. It presents the platform to those who want to develop SaaS solutions at competitive costs.

Google is a leader in web-based applications,

Google is a leader in web-based applications and leading searching engine in the world. so it’s not surprising that the company also offers cloud development services.

- These services come in the form of the Google App Engine, which enables developers to build their own web applications utilizing the same infrastructure that powers Google’s powerful applications.
- The Google App Engine provides a fully integrated application environment. Using Google’s development tools and computing cloud, App Engine applications are easy to build, easy to maintain, and easy to scale. All you have to do

Features of App Engine

1. These are covered by the depreciation policy and the service-level agreement of the app engine. Any changes made to such a feature are backward-compatible and implementation of such a feature is usually stable. These include data storage, retrieval, and search; communications; process management; computation; app configuration and management.
2. Data storage, retrieval, and search include features such as HRD migration tool, Google Cloud SQL, logs, datastore, dedicated Memcache, blobstore, Memcache and search.
3. Communications include features such as XMPP. channel, URL fetch, mail, and Google Cloud Endpoints.
4. Process management includes features like scheduled tasks and task queue. Computation includes images.
5. App management and configuration cover app identity, users, capabilities, traffic splitting, modules, SSL for custom domains, modules, remote access, and multitenancy

Centralizing email Communications

- The key here is to enable anywhere/anytime access to email. Precloud computing, your email access was via a single computer, which also stored all your email messages.

- For this purpose, you probably used a program like Microsoft Outlook or Outlook Express, installed on your home computer.

- To check your home email from work, it took a bit of juggling and perhaps the use of your ISP’s email access web page. That web page was never in sync with the messages on your home PC, of course, which is just the start of the problemswith trying to communicate in this fashion.

- A better approach is to use a web-based email service, such as Google's Gmail (mail.google.com), Microsoft's Windows Live Hotmail (mail.live.com), or Yahoo! Mail (mail.yahoo.com).
- These services place your email inbox in the cloud and you can access it from any computer connected to the Internet.

Collaborating via Web-Based Communication Tools-GMAIL

1. Gmail offers a few unique features that set it apart from the web-based email crowd.
2. First, Gmail doesn't use folders. With Gmail you can't organize your mail into folders, as you can with the other services.
3. Instead, Gmail pushes the search paradigm as the way to find the messages you want— not a surprise, given Google's search-centric business model.
4. Gmail does, however, let you "tag" each message with one or more labels. This has the effect of creating virtual folders, as you can search and sort your messages by any of their labels.
5. In addition, Gmail groups together related email messages in what Google calls conversationsYahoo! Mail Yahoo! Mail (mail.yahoo.com)
6. There is another web mail service, provided by the popular Yahoo! search site.
7. The basic Yahoo! Mail is free and can be accessed from any PC, using any web browser.
8. Yahoo! also offers a paid service called Yahoo! Mail Plus that lets you send larger messages and offers offline access to your messages via POP email clients.

Web Mail Services

- AOL Mail (mail.aol.com)
- BigString (www.bigstring.com)
- Excite Mail (mail.excite.com)
- FlashMail (www.flashmail.com)
- GMX Mail (www.gmx.com)
- Inbox.com (www.inbox.com)
- Lycos Mail (mail.lycos.com)
- Mail.com (www.mail.com)
- Zoho Mail (zoho.mail.com)

Data Security

- Data Security defines as Information in a cloud environment has much more dynamism and fluidity than information that is static on a desktop or in a network folder
- Nature of cloud computing dictates that data are fluid objects, accessible froma multitude of nodes and geographic locations and, as such, must have a datasecurity methodology that takes this into account while ensuring that this fluidity is not compromised.
- The idea of content-centric or information-centric protection, being an inherent part of a data object is a development out of the idea of the “de-perimerization” of the enterprise.
- This idea was put forward by a group of Chief Information Officers (CIOs) who formed an organization called the Jericho Forum

The Current State of data Security in the Cloud:

When it comes to data, the cloud poses a variety of risks that the enterprise must address as part of its security strategy. The biggest risks—as organizations increasingly rely on the cloud for collecting, storing, and processing critical data—are cyberattacks and data breaches.

A SailPoint survey, for example, found that 45% of companies that have implemented IaaS have experienced cyberattacks and 25% have experienced a data breach. Other research found that IT security professionals cite the proliferation of cloud services as the second-biggest barrier to their ability to respond to a data breach, and this challenge has grown in recent years.

Some of the common cloud-related risks that organizations face include:

- **Regulatory noncompliance**—whether it’s the General Protection Data Regulation (GDPR) or the Healthcare Insurance Portability and Accountability Act (HIPAA), cloud computing adds complexity to satisfying compliance requirements.
- **Data loss and data leaks**—data loss and data leaks can result from poor security practices such as mis configurations of cloud systems or threats such as insiders.
- **Loss of customer trust and brand reputation**—customers trust organizations to safeguard their personally identifiable information (PII) and when a security incident leads to data compromise, companies lose customer goodwill.
- **Business interruption**—risk professionals around the globe identified business disruption caused by failure of cloud technology / platforms or supply chains as one of their top five cyber exposure concerns.[2]

- **Financial losses**—the costs of incident mitigation, data breaches, business disruption, and other consequences of cloud security incidents can add up to hundreds of millions of dollars.

Cloud Computing and Data Security Risk

- Cloud computing is a development that is meant to allow more open accessibility and easier and improved data sharing.
- Data are uploaded into a cloud and stored in a data center, for access by users from that data center; or in a more fully cloud-based model, the data themselves are created in the cloud and stored and accessed from the cloud (again via a data center).
- The most obvious risk in this scenario is that associated with the storage of that data. A user uploading or creating cloud-based data include those data that are stored and maintained by a third-party cloud provider such as Google, Amazon, Microsoft, and so on.

This action has several risks associated with it:

- Firstly, it is necessary to protect the data during upload into the data center to ensure that the data do not get hijacked on the way into the database.
- Secondly, it is necessary to the stores the data in the data center to ensure that they are encrypted at all times.
- Thirdly, and perhaps less obvious, the access to those data need to be controlled; this control should also be applied to the hosting company, including the administrators of thedata center.
- In addition, an area often forgotten in the application of security to a data resource is the protection of that resource during its use

Data security risks are compounded by the open nature of cloud computing.

- Access control becomes a much more fundamental issue in cloud-based systems because of the accessibility of the data

- Information-centric access control (as opposed to access control lists) can help to balance improved accessibility with risk, by associating access rules with different data objects within an open and accessible platform, without losing the Inherent usability of that platform

- A further area of risk associated not only with cloud computing, but also with traditional network computing, is the use of content after access.

- The risk is potentially higher in a cloud network, for the simple reason that the information is outside of your corporate walls.

CLOUD COMPUTING AND IDENTITY

Digital identity

- Digital identity holds the key to flexible data security within a cloud Environment
- A digital identity represents who we are and how we interact with others on-line.
- **Access, identity, and risk** are three variables that can become inherently connected when applied to the security of data, because access and risk are directly proportional: As access increases, so then risk to the security of the data increases.
- Access controlled by identifying the actor attempting the access is the most logical manner of performing this operation.
- Ultimately, digital identity holds the key to securing data, if that digital identity can be programmatically linked to security policies controlling the post-access usage ofdata.

Identity, Reputation, and Trust

1. Reputation is a real-world commodity; that is a basic requirement of human-to-human relationships
2. Our basic societal communication structure is built upon the idea of reputation and trust.
3. Reputation and its counter value, trust, is easily transferable to a digital realm:
4. EBay, for example, having partly built a successful business model on the strength of a ratings system, builds up the reputation of its buyers and sellers through successful (or unsuccessful) transactions.
5. These types of reputation systems can be extremely useful when used with a digital identity.
6. They can be used to associate varying levels of trust with that identity, which in turn can be used to define the level (granular variations) of security policy applied to data resources that the individual wishes to access

User-Centric Identity:

I. Digital identities are a mechanism for identifying an individual, particularly within a cloud environment and identity ownership being placed upon the individual is known as user- centric identity

II. It allows users to consent and control how their identity (and the individual identifiers making up the identity, the claims) is used.

III. This reversal of ownership away from centrally managed identity platforms(enterprise- centric) has many advantages.

IV. This includes the potential to improve the privacy aspects of a digital identity, by giving an individual the ability to apply permission policies based on their identity and to control which aspects of that identity are divulged

V. An identity may be controllable by the end user, to the extent that the user can then decide what information is given to the party relying on the identity

Information Card:

1. Information cards permit a user to present to a Web site or other service (relying party) one or more claims, in the form of a software token, which may be used to uniquely identify that user.

2. They can be used in place of user name/ passwords, digital certificates, and other identification systems, when user identity needs to be established to control access to a Web site or other resource, or to permit digital signing

3. Information cards are part of an identity meta-system consisting of:

   I. **Identity providers (IdP**), who provision and manage information cards with specific claims, to users.

   II. **Users** who own and utilize the cards to gain access to Web sites and other resources that support information cards.

   III. **An identity selector/service**, which is a piece of software on the user's desktop or in the cloud that allows a user to select and manage their cards.

   IV. **Relying parties.** These are the applications, services & so on, that can use an information card to authenticate a person and to then authorize an action such as logging onto a Web site, accessing a document, signing content, and so on.
   5. Each information card is associated with a set of claims which can be used to identify the user. These claims include identifiers such as name, email address post code.

Using Information Cards to Protect Data

i. Information cards are built around a set of open standards devised by a consortium that includes Microsoft, IBM, Novell, and so on.

ii. The original remit of the cards was to create a type of single sign on system for the Internet, to help users to move away from the need to remember multiple passwords.

iii. However, the information card system can be used in many more ways.

iv. Because an information card is a type of digital identity, it can be used in the same way that other digital identities can be used.

v. For example, an information card can be used to digitally sign data and content and to control access to data and content. One of the more sophisticated uses of an information card is the advantage given to the cards by way of the claims system.

Cloud Computing and Data Security Risk

1. Cloud computing is a development that is meant to allow more open accessibility and easier and improved data sharing.
2. Data are uploaded into a cloud and stored in a data center, for access by users from that data center; or in a more fully cloud-based model, the data themselves are created in the cloud and stored and accessed from the cloud (again via a data center).
3. The most obvious risk in this scenario is that associated with the storage of that data. A user uploading or creating cloud-based data include those data that are stored and maintained by a third-party cloud provider such as Google, Amazon, Microsoft, and so on.

This action has several risks associated with it:

- Firstly, it is necessary to protect the data during upload into the data center to ensure that the data do not get hijacked on the way into the database.
- Secondly, it is necessary to the stores the data in the data center to ensure that they are encrypted at all times.
- Thirdly, and perhaps less obvious, the access to those data need to be controlled; this control should also be applied to the hosting company, including the administrators of the data center.
- In addition, an area often forgotten in the application of security to a data resource is the protection of that resource during its use

Data security risks are compounded by the open nature of cloud computing.

- Access control becomes a much more fundamental issue in cloud-based systems because of the accessibility of thedata
- Information-centric access control (as opposed to access control lists) can help to balance improved accessibility with risk, by associating access rules with different data objects within an open and accessible platform, without losing the Inherent usability of that platform
- A further area of risk associated not only with cloud computing, but also with traditional network computing, is the use of content after access.

- The risk is potentially higher in a cloud network, for the simple reason that the information is outside of your corporate walls

Data-centric mashups

- that are used to perform business processes around data creation and dissemination—by their very nature, can be used to hijack data, leaking sensitive information and/or affecting integrity of that data
- Cloud computing, more than any other form of digital communication technology, has created a need to ensure that protection is applied at the inception of the information, in a content centric manner, ensuring that a security policy becomes an integral part of that data throughout its life cycle.

Encryption

- It is a vital component of the protection policy, but further controls over the access of that data and on the use of the data must be met.
- In the case of mashups, the controlling of access to data resources, can help toalleviate the security concerns by ensuring that mashup access is authenticated.
- Linking security policies, as applied to the use of content, to the access control method offer a way of continuing protection of data, post access and throughout the life cycle; this type of data security philosophy must be incorporated into the use of cloud computing to alleviate security risks.

UNIT- V

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

SLA MANAGEMENT IN CLOUD COMPUTING

In the early days of web-application deployment, performance of the application at peak load was a single important criterion for provisioning server resources. Provisioning in those days involved deciding hardware configuration, determining the number of physical machines, and acquiring them upfront so that the overall business objectives could be achieved. The web applications were hosted on these dedicated individual servers within enterprises' own server rooms. These web applications were used to provide different kinds of e-services to various clients. Typically, the service-level objectives (SLOs) for these applications were response time and throughput of the application end-user requests. The capacity buildup was to cater to the estimated peak load experienced by the application. The activity of determining the number of servers and their capacity that could satisfactorily serve the application end-user requests at peak loads is called capacity planning. An example scenario where two web applications, application A and application B, are hosted on a separate set of dedicated servers within the enterprise-owned server rooms is shown in Figure 16.1. The planned capacity for each of the applications to run successfully is three servers. As the number of web applications grew, the server rooms in the organization became large and such server rooms were known as data centers. These data centers were owned and managed by the enterprises themselves.

414 SLA MANAGEMENT IN CLOUD COMPUTING: A SERVICE PROVIDER'S PERSPECTIVE

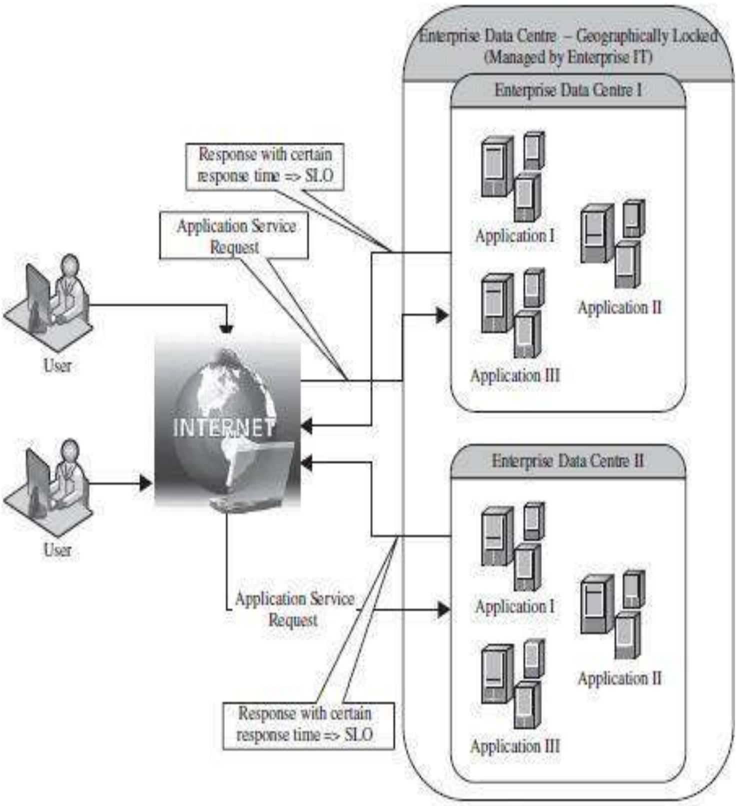

Enterprise Data Centre – Geographically Locked
(Managed by Enterprise IT)
Enterprise Data Centre I
Response with certain response time => SLO
Application Service Request
Application I
Application II
Application III
User
INTERNET
User
Enterprise Data Centre II
Application I
Application II
Application III
Application Service Request
Response with certain response time => SLO

FIGURE 16.1. Hosting of applications on servers within enterprise's data centers.

TRADITIONAL APPROACHES TO SLO MANAGEMENT

Traditionally, load balancing techniques and admission control mechanisms have been used to provide guaranteed quality of service (QoS) for hosted web applications. These mechanisms can be viewed as the first attempt towards managing the SLOs. In the following subsections we discuss the existing approaches for load balancing and admission control for ensuring QoS.

16.2.1 Load Balancing The objective of a load balancing is to distribute the incoming requests onto a set of physical machines, each hosting a replica of an application, so that the load on the machines is equally distributed [4]. The load balancing algorithm executes on a physical machine that interfaces with the clients. This physical machine, also called the front- end node, receives the incoming requests and distributes these requests to different physical machines for further execution.

This set of physical machines is responsible for serving the incoming requests and are known as the back-end nodes.

TYPES OF SLA:

Service-level agreement provides a framework within which both seller and buyer of a service can pursue a profitable service business relationship. It outlines the broad understanding between the service provider and the service consumer for conducting business and forms the basis for maintaining a mutually beneficial relationship. From a legal perspective, the necessary terms and conditions that bind the service provider to provide services continually to the service consumer are formally defined in SLA.

SLA can be modeled using web service-level agreement (WSLA) language specification [7]. Although WSLA is intended for web-service-based applications, it is equally applicable for hosting of applications. Service-level parameter, metric, function, measurement directive, service-level objective, and penalty are some of the important components of WSLA and are described in Table 16.1.

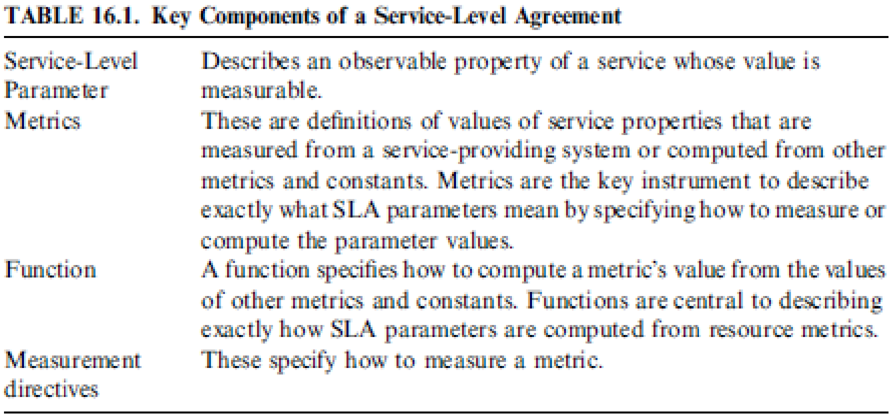


TABLE 16.1. Key Components of a Service-Level Agreement

Service-Level Parameter	Describes an observable property of a service whose value is measurable.
Metrics	These are definitions of values of service properties that are measured from a service-providing system or computed from other metrics and constants. Metrics are the key instrument to describe exactly what SLA parameters mean by specifying how to measure or compute the parameter values.
Function	A function specifies how to compute a metric's value from the values of other metrics and constants. Functions are central to describing exactly how SLA parameters are computed from resource metrics.
Measurement directives	These specify how to measure a metric.

There are two types of SLAs from the perspective of application hosting. These are described in detail here.

Infrastructure SLA. The infrastructure provider manages and offers guarantees on availability of the infrastructure, namely, server machine, power, network connectivity, and so on. Enterprises manage themselves, their applications that are deployed on these server machines. The machines are leased to the customers and are isolated from machines of other customers. In such dedicated hosting environments, a practical example of service-level guarantees offered by infrastructure providers is shown in Table 16.2.

Application SLA. In the application co-location hosting model, the server capacity is available to the applications based solely on their resource demands. Hence, the service providers are flexible in allocating and de-allocating computing resources among the co-located applications.

Therefore, the service providers are also responsible for ensuring to meet their customer's application SLOs. For example, an enterprise can have the following application SLA with a service provider for one of its application.

LIFE CYCLE OF SLA:

Each SLA goes through a sequence of steps starting from identification of terms and conditions, activation and monitoring of the stated terms and conditions, and eventual termination of contract once the hosting relationship ceases to exist. Such a sequence of steps is called SLA life cycle and consists of the following five phases:

1. Contract definition
2. Publishing and discovery
3. Negotiation
4. Operationalization
5. De-commissioning

Here, we explain in detail each of these phases of SLA life cycle.

Contract Definition. Generally, service providers define a set of service offerings and corresponding SLAs using standard templates. These service offerings form a catalog. Individual SLAs for enterprises can be derived by customizing these base SLA templates.

Publication and Discovery. Service provider advertises these base service offerings through standard publication media, and the customers should be able to locate the service provider by searching the catalog. The customers can search different competitive offerings and shortlist a few that fulfill their requirements for further negotiation.

Negotiation. Once the customer has discovered a service provider who can meet their application hosting need, the SLA terms and conditions needs to be mutually agreed upon before signing the agreement for hosting the application. For a standard packaged application which is offered as service, this phase could be automated. For customized applications that are hosted on cloud platforms, this phase is manual. The service provider needs to analyze the application's behavior with respect to scalability and performance before agreeing on the specification of SLA. At the end of this phase, the SLA is mutually agreed by both customer and provider and is eventually signed off. SLA negotiation can utilize the WS-negotiation specification [8].

Operationalization. SLA operation consists of SLA monitoring, SLA accounting, and SLA enforcement. SLA monitoring involves measuring parameter values and calculating the metrics defined as a part of SLA and determining the deviations. On identifying the deviations, the concerned parties are notified. SLA accounting involves capturing and archiving the SLA adherence for compliance.

As part of accounting, the application's actual performance and the performance guaranteed as a part of SLA is reported. Apart from the frequency and the duration of the SLA breach, it should also provide the penalties paid for each SLA violation. SLA enforcement involves taking appropriate action when the runtime monitoring detects a SLA violation. Such actions could be notifying the concerned parties, charging the penalties besides other things. The different policies can be expressed using a subset of the Common Information Model (CIM) [9]. The CIM model is an open standard that allows expressing managed elements of data center via relationships and common objects.

De-commissioning. SLA decommissioning involves termination of all activities performed under a particular SLA when the hosting relationship between the service provider and the service consumer has ended. SLA specifies the terms and conditions of contract termination and specifies situations under which the relationship between a service provider and a service consumer can be considered to be legally ended.

SLA MANAGEMENT IN CLOUD: SLA management of applications hosted on cloud platforms involves five phases.

1. Feasibilitys
2. On-boarding
3. Pre-production
4. Production
5. Termination

Different activities performed under each of these phases are shown in Figure 16.7. These activities are explained in detail in the following subsections.

Feasibility Analysis

MSP conducts the feasibility study of hosting an application on their cloud platforms. This study involves three kinds of feasibility: (1) technical feasibility, (2) infrastructure feasibility, and (3) financial feasibility. The technical feasibility of an application implies determining the following:

1. Ability of an application to scale out.
2. Compatibility of the application with the cloud platform being used within the MSP’s data center.
3. The need and availability of a specific hardware and software required for hosting and running of the application.
4. Preliminary information about the application performance and whether they can be met by the MSP.

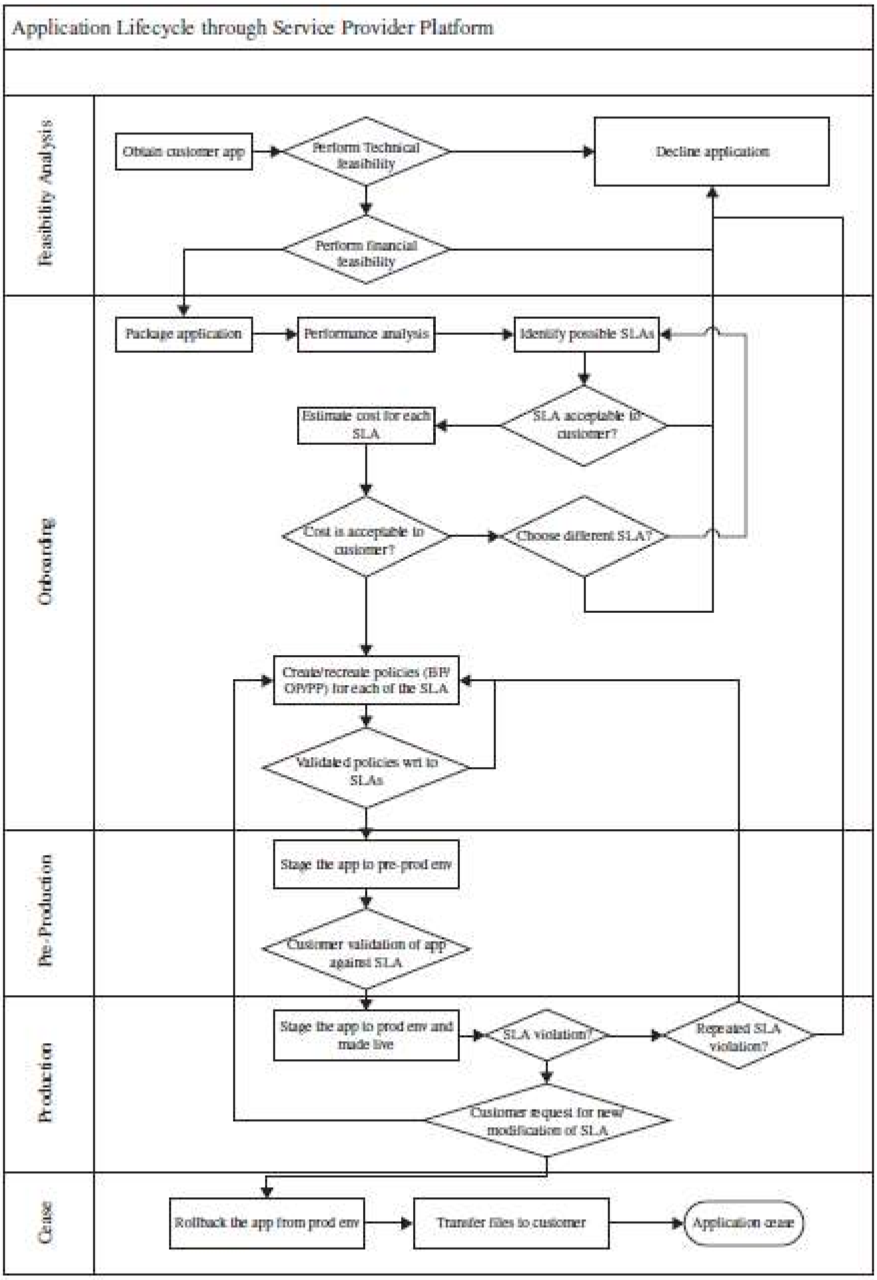

Application Lifecycle through Service Provider Platform
Feasibility Analysis
Obtain customer app
Perform Technical feasibility
Decline application
Perform financial feasibility
Onboarding
Package application
Performance analysis
Identify possible SLAs
Estimate cost for each SLA
SLA acceptable to customer?
Cost is acceptable to customer?
Choose different SLA?
Create/recreate policies (BP/OP/PP) for each of the SLA
Validated policies wrt to SLAs
Pre-Production
Stage the app to pre-prod env
Customer validation of app against SLA
Production
Stage the app to prod env and made live
SLA violation?
Repeated SLA violation?
Customer request for new/modification of SLA
Cease
Rollback the app from prod env
Transfer files to customer
Application cease

FIGURE 16.7. Flowchart of the SLA management in cloud.

On-Boarding of Application

Once the customer and the MSP agree in principle to host the application based on the findings of the feasibility study, the application is moved from the customer servers to the hosting platform. Moving an application to the MSP's hosting platform is called on-boarding [10]. As part of the on-boarding activity, the MSP understands the application runtime characteristics using runtime profilers. This helps the MSP to identify the possible SLAs that can be offered to the customer for that application. This also helps in creation of the necessary policies (also called rule sets) required to guarantee the SLOs mentioned in the application SLA. The application is accessible to its end users only after the on-boarding activity is completed.

Preproduction

Once the determination of policies is completed as discussed in previous phase, the application is hosted in a simulated production environment. It facilitates the customer to verify and validate the MSP's findings on application's runtime characteristics and agree on the defined SLA. Once both parties agree on the cost and the terms and conditions of the SLA, the customer sign-off is obtained. On successful completion of this phase the MSP allows the application to go on-live.

Production

In this phase, the application is made accessible to its end users under the agreed SLA. However, there could be situations when the managed application tends to behave differently in a production environment compared to the preproduction environment. This in turn may cause sustained breach of the terms and conditions mentioned in the SLA. Additionally, customer may request the MSP for inclusion of new terms and conditions in the SLA. If the application SLA is breached frequently or if the customer requests for a new non-agreed SLA, the on-boarding process is performed again. In the case of the former, on-boarding activity is repeated to analyze the application and its policies with respect to SLA fulfillment. In case of the latter, a new set of policies are formulated to meet the fresh terms and conditions of the SLA.

Termination

When the customer wishes to withdraw the hosted application and does not wish to continue to avail the services of the MSP for managing the hosting of its application, the termination activity is initiated. On initiation of termination, all data related to the application are transferred to the customer and only the essential information is retained for legal compliance. This ends the hosting relationship between the two parties for that application, and the customer sign-off is obtained.