CLOUD COMPUTING FUNDAMENTALS

DIGITAL NOTES

B. TECH
(III YEAR – II SEM)
(2022-23)

Department of Electronics and Communication Engineering

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956
Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)Maisammaguda, Dhulapally (Post Via. Kompally), Secunderabad – 500100, Telangana State, India

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

III Year B.Tech. ECE- II Sem **L/T/P/C**
3/-/-/3

OPEN ELECTIVE - III

(R20A0555) CLOUD COMPUTING FUNDAMENTALS

COURSE OBJECTIVES:

1) To learn various system models for Distributed and Cloud Computing.
2) To understand about Virtual machines, Its Structure and mechanisms.
3) To learn Cloud Computing Paradigm.
4) To introduce the various levels of services that can be achieved by cloud.
5) To describe the security aspects in cloud.

UNIT- I

Systems Modeling: System Models for Distributed and Cloud Computing- Cloud Computing in a Nutshell, Layers and Types of Clouds, Desired Features of a Cloud, Infrastructure as a Service Providers, Platform as a Service Providers, Challenges and Risks.

UNIT- II

Virtualization: Virtual machines, Implementation Levels of Virtualization -Virtualization Structures/Tools and Mechanisms-Virtualization of CPU, Memory, and I/O Devices

UNIT- III

Foundations: Introduction to Cloud Computing- Migrating into a Cloud-The Enterprise Cloud Computing Paradigm.

UNIT- IV

Infrastructure as a Service (IAAS) & Platform (PAAS): Virtual machines provisioning and Migration services-On the Management of Virtual machines for Cloud Infrastructures-Aneka— Integration of Private and Public Clouds.

UNIT- V

Software as a Service (SAAS) &Data Security in the Cloud: Google App Engine, An Introduction to the idea of Data Security- The Current State of Data Security in the Cloud-Cloud Computing and Data Security Risk- Cloud Computing and Identity.

TEXT BOOKS:

1) Distributed and Cloud Computing, Kaittwang Geoffrey C.Fox and Jack J Dongrra, Elsevier India 2012.
2) Mastering Cloud Computing- Raj Kumar Buyya, Christian Vecchiola and S.Tanurai Selvi,TMH, 2012.
3) Michael Miller, Cloud Computing: Web-Based Applications That Change the Way YouWork and Collaborate Online, Que Publishing, August 2008.

COURSE OUTCOMES:

1) Understanding various system models for Distributed and Cloud Computing.
2) Understanding about Virtual machines, Its Structure and mechanisms.
3) Learning Cloud Computing Paradigm.
4) Understanding the various levels of services that can be achieved by cloud.
5) Learning about security aspects in cloud.

UNIT-I
SYSTEMS MODELLING

INTRODUCTION TO CLOUD COMPUTING

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

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

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

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

Key characteristics of cloud computing

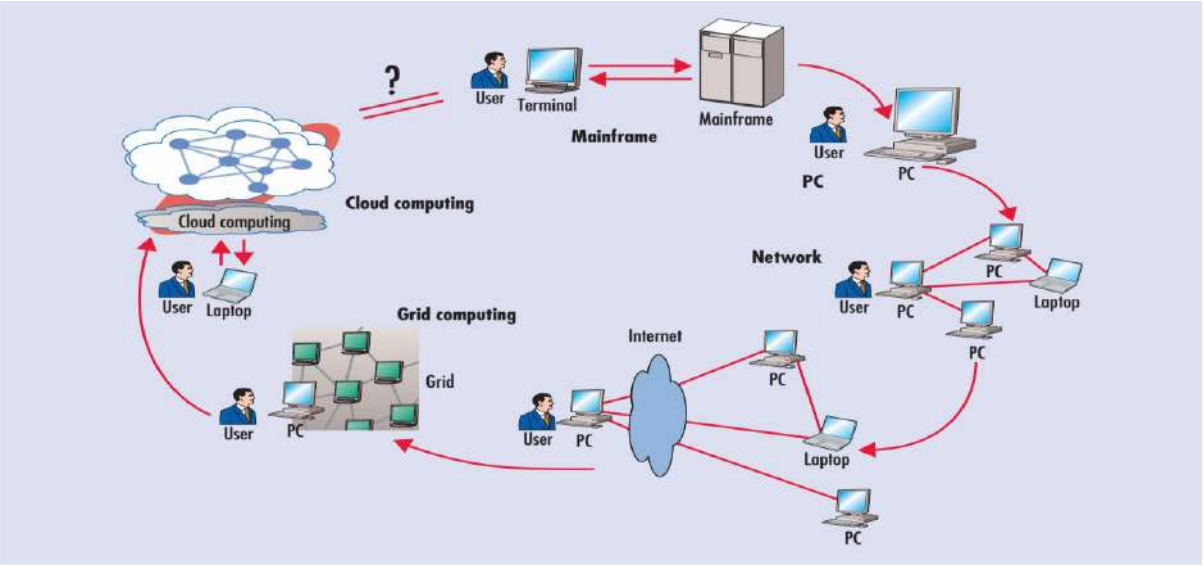
The illusion of infinite computing resources
The elimination of an up-front commitment by cloud users
The ability to pay for use…as needed

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

Most common characteristics which a cloud should have:

(i) Pay-per-use (no ongoing commitment, utility prices);
(ii) Elastic capacity and the illusion of infinite resources;
(iii) Self-service interface;
(iv) Resources that is abstracted or virtualized;

HISTORY OF CLOUD COMPUTING



- Six phases of computing paradigms, from dummy terminals / mainframes, to PCs, network computing, grid and cloud computing.
- Phase 1: Dummy terminals are used to connect to powerful hosts shared by many users. At that time, the terminals were basically keyboards and monitors.
- Phase 2: Independent personal computers (PCs) became powerful enough to satisfy users' daily work, which means that you didn't have to share a mainframe with anyone else.
- Phase 3: A computer networks that allowed multiple computers to connect to each other. You could work on a PC and connect to other computers through local networks (LAN) to share re-sources.
- Phase 4: Local networks were connected to other local networks to establish a more global network - users could now connect to the Internet to utilize remote applications and resources.
- Phase 5: Brought us the concept of an electronic grid to facilitate shared computing power and storage resources (distributed computing). People used PCs to access a grid of computers transparently.
- Phase 6: We can leverage all available resources on the Internet in an extremely scalable and simple way by cloud computing.

COMPUTING PARADIGM DISTINCTIONS

- Centralized computing: This is a computing paradigm by which all computer resources are centralized in one physical system. All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS.
- Parallel computing: In parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory.

- Distributed computing: (or distributed processing) is the technique of linking together multiple computer servers over a network into a cluster, to share data and to coordinate processing power. Such a cluster is referred to as a "distributed system."
    - Distributed computing offers advantages in scalability (through a "scale-out architecture"), performance (via parallelism), resilience (via redundancy), and cost-effectiveness (through the use of low-cost, commodity hardware).
- Cloud computing: An Internet cloud of resources can be either a centralized or a distributed computing system. The cloud applies parallel or distributed computing, or both.
- Clouds can be built with physical or virtualized resources over large data centers that are centralized or distributed. Some authors consider cloud computing to be a form of utility computing or service computing.

SYSTEM MODELS FOR DISTRIBUTED AND CLOUD COMPUTING

Distributed and cloud computing systems are built over a large number of autonomous computer nodes. These node machines are interconnected by SANs, LANs, or WANs in a hierarchical manner. One can build a massive system with millions of computers connected to edge networks. Massive systems are considered highly scalable, and can reach web-scale connectivity, either physically or logically.

Massive systems are classified into four groups:

- Clusters
- P2P networks
- Computing grids
- Internet clouds over huge data centers.

In terms of node number, these four system classes may involve hundreds, thousands, or even millions of computers as participating nodes. These machines work collectively, cooperatively, or collaboratively at various levels.

(1) CLUSTERS OF COOPERATIVE COMPUTERS/CLUSTER COMPUTING

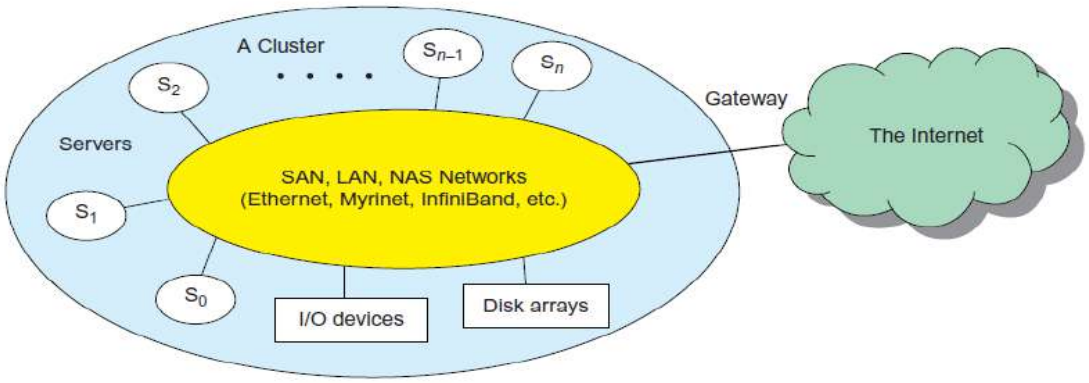
Cluster computing refers that many of the computers connected on a network and they perform like a single entity. Each computer that is connected to the network is called a node. Cluster computing offers solutions to solve complicated problems by providing faster computational speed, and enhanced data integrity. The connected computers execute operations all together thus creating the impression like a single system (virtual machine). This process is termed as transparency of the system. Based on the principle of distributed systems, this networking technology performs its operations. And here, LAN is the connection unit.

Cluster computing goes with the features of:

- All the connected computers are the same kind of machines
- They are tightly connected through dedicated network connections
- All the computers share a common home directory.

Cluster Architecture

- The architecture of a typical server cluster built around a low-latency, high bandwidth interconnection network is as shown below.
- This network can be as simple as a SAN (e.g., Myrinet) or a LAN (e.g., Ethernet). To build a larger cluster with more nodes, the interconnection network can be built with multiple levels of Gigabit Ethernet, Myrinet, or InfiniBand switches.
- Through hierarchical construction using a SAN, LAN, or WAN, one can build scalable clusters with an increasing number of nodes.
- The cluster is connected to the Internet via a virtual private network (VPN) gateway. The gateway IP address locates the cluster.
- The system image of a computer is decided by the way the OS manages the shared cluster resources.
- All resources of a server node are managed by their own OS. Thus, most clusters have multiple system images as a result of having many autonomous nodes under different OS control.



FIGURE

A cluster of servers interconnected by a high-bandwidth SAN or LAN with shared I/O devices and disk arrays; the cluster acts as a single computer attached to the Internet.

Cluster Computing: Single-System Image

- An ideal cluster should merge multiple system images into a single-system image (SSI).

- Cluster designers desire a cluster operating system or some middleware to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.
- An SSI is an illusion created by software or hardware that presents a collection of resources as one integrated, powerful resource.
- SSI makes the cluster appear like a single machine to the user.
- A cluster with multiple system images is nothing but a collection of independent computers

Cluster Computing: Hardware, Software, and Middleware Support

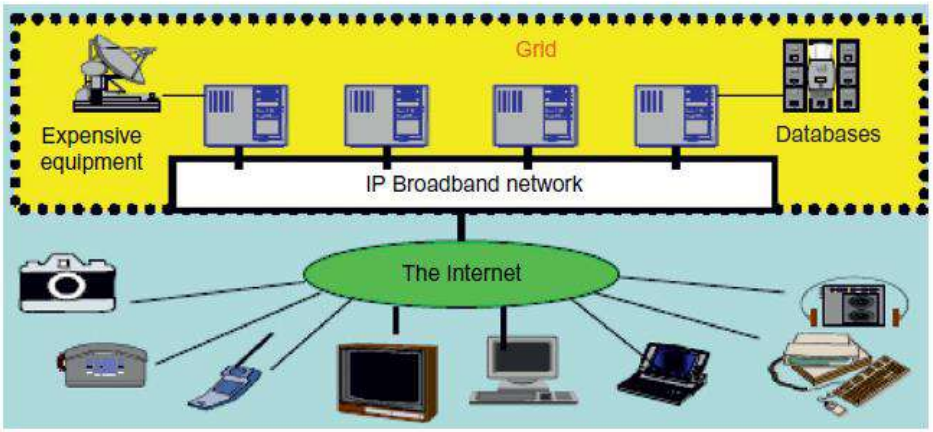
- Clusters exploring massive parallelism are commonly known as MPPs.
- The building blocks are computer nodes (PCs, workstations, servers, or SMP), special communication software such as PVM or MPI, and a network interface card in each computer node.
- Most clusters run under the Linux OS. The computer nodes are interconnected by a high-bandwidth network (such as Gigabit Ethernet, Myrinet, InfiniBand, etc.).
- Special cluster middleware supports are needed to create SSI or high availability (HA).
- Both sequential and parallel applications can run on the cluster, and special parallel environments are needed to facilitate use of the cluster resources.
- For example, distributed memory has multiple images. Users may want all distributed memory to be shared by all servers by forming distributed shared memory (DSM).
- Many SSI features are expensive or difficult to achieve at various cluster operational levels. Instead of achieving SSI, many clusters are loosely coupled machines.
- Using virtualization, one can build many virtual clusters dynamically, upon user demand.

Cluster Computing: Major Design Issues

- Middleware or OS extensions are developed at the user space to achieve SSI at selected functional levels.
- Without this middleware, cluster nodes cannot work together effectively to achieve cooperative computing.
- The software environments and applications must rely on the middleware to achieve high performance.
- The cluster benefits come from scalable performance, efficient message passing, high system availability, seamless fault tolerance, and cluster-wide job management.

(2) GRID COMPUTING

- Internet services such as the Telnet command enables a local computer to connect to a remote computer.
- A web service such as HTTP enables remote access of remote web pages.
- Grid computing is envisioned to allow close interaction among applications running on distant computers simultaneously.
- A computing grid offers an infrastructure that couples computers, software/middleware, special instruments, and people and sensors together.
- The grid is often constructed across LAN, WAN, or Internet backbone networks at a regional, national, or global scale.
- Enterprises or organizations present grids as integrated computing resources.
- They can also be viewed as virtual platforms to support virtual organizations.
- The computers used in a grid are primarily workstations, servers, clusters, and supercomputers.
- Personal computers, laptops, and PDAs can be used as access devices to a grid system.



FIGURE

Computational grid or data grid providing computing utility, data, and information services through resource sharing and cooperation among participating organizations.

Grid computing: Computational Grids

- Computational grid built over multiple resource sites owned by different organizations.
- The resource sites offer complementary computing resources, including workstations, large servers, a mesh of processors, and Linux clusters to satisfy a chain of computational needs.
- The grid is built across various IP broadband networks including LANs and WANs already used by enterprises or organizations over the Internet.

- The grid is presented to users as an integrated resource pool as shown in the upper half of the figure.
- At the server end, the grid is a network. At the client end, we see wired or wireless terminal devices.
- The grid integrates the computing, communication, contents, and transactions as rented services.
- Enterprises and consumers form the user base, which then defines the usage trends and service characteristics.

Grid computing: Grid Families

- Grid technology demands new distributed computing models, software/middleware support, network protocols, and hardware infrastructures.
- National grid projects are followed by industrial grid platform development by IBM, Microsoft, Sun, HP, Dell, Cisco, EMC, Platform Computing, and others.
- New grid service providers (GSPs) and new grid applications have emerged rapidly.
- Grid systems are classified in essentially two categories: Computational or data grids and P2P grids.

Table Two Grid Computing Infrastructures and Representative Systems

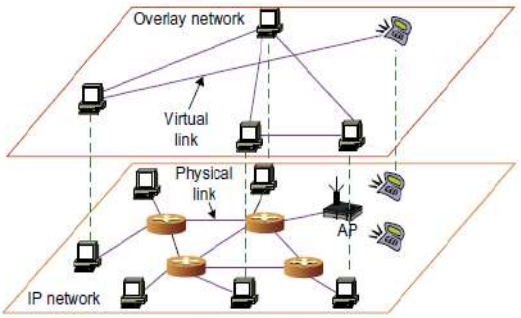
Design Issues	Computational and Data Grids	P2P Grids
Grid Applications Reported	Distributed supercomputing, National Grid initiatives, etc.	Open grid with P2P flexibility, all resources from client machines
Representative Systems	TeraGrid built in US, ChinaGrid in China, and the e-Science grid built in UK	JXTA, FightAid@home, SETI@home
Development Lessons Learned	Restricted user groups, middleware bugs, protocols to acquire resources	Unreliable user-contributed resources, limited to a few apps

(3) PEER-TO-PEER NETWORK FAMILIES

- Well-established distributed system is the client-server architecture.
- In this scenario client machines (PCs and workstations) are connected to a central server for compute, e-mail, file access, and database applications.
- The P2P architecture offers a distributed model of networked systems.
- First, a P2P network is client-oriented instead of server-oriented.

Peer-to-Peer Network Families: P2P Systems

- In a P2P system, every node acts as both a client and a server, providing part of the system resources.
- Peer machines are simply client computers connected to the Internet. All client machines act autonomously to join or leave the system freely.
- This implies that no master-slave relationship exists among the peers. No central coordination or central database is needed.
- In other words, no peer machine has a global view of the entire P2P system.
- The system is self-organizing with distributed control.



FIGURE

The structure of a P2P system by mapping a physical IP network to an overlay network built with virtual links.

- The architecture of a P2P network is at two abstraction levels. Initially, the peers are totally unrelated.
- Each peer machine joins or leaves the P2P network voluntarily. Only the participating peers form the physical network at any time.
- Unlike the cluster or grid, a P2P network does not use a dedicated interconnection network.
- The physical network is simply an ad hoc network formed at various Internet domains randomly using the TCP/IP and NAI protocols.
- Thus, the physical network varies in size and topology dynamically due to the free membership in the P2P network.

Peer-to-Peer Network Families: Overlay Networks

- Data items or files are distributed in the participating peers.
- Based on communication or file-sharing needs, the peer IDs form an overlay network at the logical level.
- This overlay is a virtual network formed by mapping each physical machine with its ID, logically, through a virtual mapping
- When a new peer joins the system, its peer ID is added as a node in the overlay network.

- When an existing peer leaves the system, its peer ID is removed from the overlay network automatically.
- Therefore, it is the P2P overlay network that characterizes the logical connectivity among the peers.
- There are two types of overlay networks: unstructured and structured.
- An unstructured overlay network is characterized by a random graph.
    - There is no fixed route to send messages or files among the nodes.
    - Often, flooding is applied to send a query to all nodes in an unstructured overlay, thus resulting in heavy network traffic and nondeterministic search results.
- Structured overlay networks follow certain connectivity topology and rules for inserting and removing nodes (peer IDs) from the overlay graph.
    - Routing mechanisms are developed to take advantage of the structured overlays.

Peer-to-Peer Network Families: P2P Application Families

- Based on application, P2P networks are classified into four groups
- The first family is for distributed file sharing of digital contents (music, videos, etc.) on the P2P network. This includes many popular P2P networks such as Gnutella, Napster, and Bit Torrent, among others.
- Collaboration P2P networks include MSN or Skype chatting, instant messaging, and collaborative design, among others.
- The third family is for distributed P2P computing in specific applications. For example, SETI@home provides 25 Tflops of distributed computing power, collectively, over 3 million Internet host machines.
- Other P2P platforms, such as JXTA, .NET, and FightingAID@home, support naming, discovery, communication, security, and resource aggregation in some P2P applications.

Peer-to-Peer Network Families: P2P Computing Challenges

- P2P computing faces three types of heterogeneity problems in hardware, software, and network requirements.
    - There are too many hardware models and architectures to select from;
    - Incompatibility exists between software and the OS;
    - Different network connections and protocols make it too complex to apply in real applications.
- System scalability is needed as the workload increases. System scaling is directly related to performance and bandwidth.

- P2P networks do have these properties. Data location is also important to affect collective performance.
- Data locality, network proximity, and interoperability are three design objectives in distributed P2P applications.

Disadvantages of P2P networks

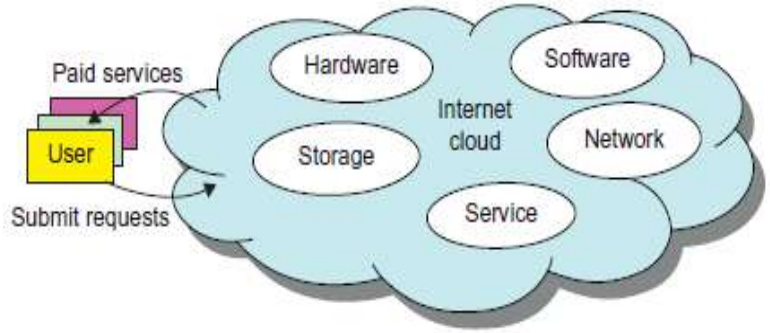
- As the system is not centralized, managing it is difficult.
- In addition, the system lacks security. Anyone can log on to the system and cause damage or abuse.
- Further, all client computers connected to a P2P network cannot be considered reliable or virus-free.
- In summary, P2P networks are reliable for a small number of peer nodes.
- They are only useful for applications that require a low level of security and have no concern for data sensitivity.

(4) CLOUD COMPUTING OVER THE INTERNET

- "Computational science is changing to be data-intensive. Supercomputers must be balanced systems, not just CPU farms but also petascale I/O and networking arrays."
- In the future, working with large data sets will typically mean sending the computations (programs) to the data, rather than copying the data to the workstations.
- This reflects the trend in IT of moving computing and data from desktops to large data centers, where there is on-demand provision of software, hardware, and data as a service.
- This data explosion has promoted the idea of cloud computing.
- IBM, a major player in cloud computing, has defined it as follows: "A cloud is a pool of virtualized computer resources. A cloud can host a variety of different workloads, including batch-style backend jobs and interactive and user-facing applications."
- Based on this definition, a cloud allows workloads to be deployed and scaled out quickly through rapid provisioning of virtual or physical machines.
- The cloud supports redundant, self-recovering, highly scalable programming models that allow workloads to recover from many unavoidable hardware/software failures.
- Finally, the cloud system should be able to monitor resource use in real time to enable rebalancing of allocations when needed.

Cloud Computing over the Internet: Internet Clouds

- Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically.
- The idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at data centers.
- Cloud computing leverages its low cost and simplicity to benefit both users and providers. Machine virtualization has enabled such cost-effectiveness.
- Cloud computing intends to satisfy many user applications simultaneously. The cloud ecosystem must be designed to be secure, trustworthy, and dependable.
- Some computer users think of the cloud as a centralized resource pool.
- Others consider the cloud to be a server cluster which practices distributed computing over all the servers used.

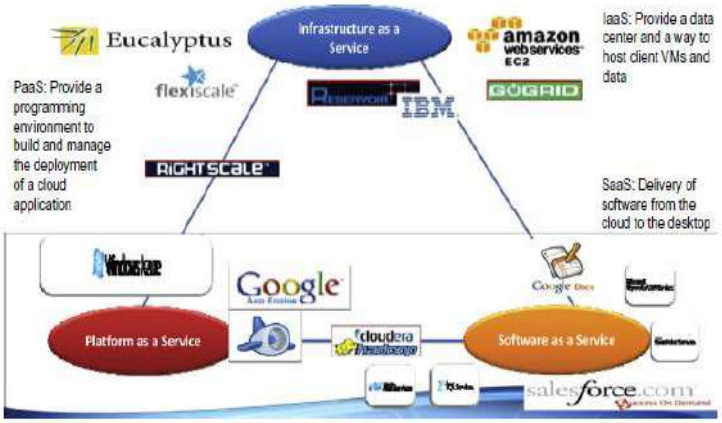


FIGURE

Virtualized resources from data centers to form an Internet cloud, provisioned with hardware, software, storage, network, and services for paid users to run their applications.

Cloud Computing over the Internet: The Cloud Landscape

- The cloud landscape and major cloud players, based on three cloud service models
    - Infrastructure as a Service (IaaS)
    - Platform as a Service (PaaS)
    - Software as a Service (SaaS)
- Internet clouds offer four deployment modes: private, public, managed, and hybrid



FIGURE

Three cloud service models in a cloud landscape of major providers.

(*Courtesy of Dennis Gannon, keynote address at Cloudcom2010 [19]*)

CLOUD COMPUTING IN A NUTSHELL

LAYERS AND TYPES OF CLOUDS

Cloud computing services are divided into three classes

(1) Infrastructure as a Service

(2) Platform as a Service

(3) Software as a Service

Figure depicts the layered organization of the cloud stack from physical infrastructure to applications.

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

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

The cloud computing stack

Infrastructure as a Service

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

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

Platform as a Service

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

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

Software as a Service

Applications reside on the top of the cloud stack. Services provided by this layer can be accessed by end users through Web portals. Therefore, consumers are increasingly shifting from locally installed computer programs to on-line software services that offer the same functionally. Traditional desktop applications such as word processing and spreadsheet can now be accessed as a service in the Web. This model of delivering applications, known as Software as a Service (SaaS), alleviates the burden of software maintenance for customers and simplifies development and testing for providers.

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

Deployment Models

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

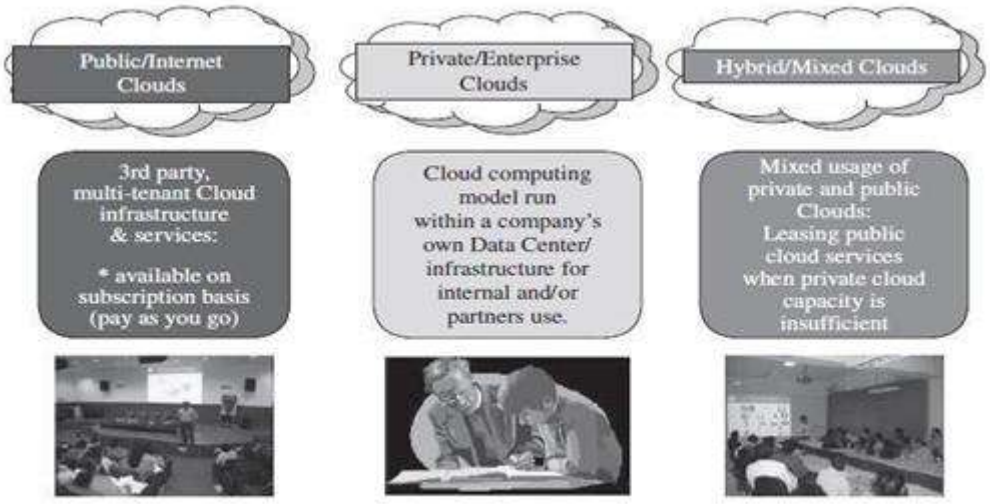


FIGURE Types of clouds based on deployment models.

Public cloud: Cloud made available in a pay-as-you-go manner to the general public.

Private cloud: Internal data center of a business or other organization, not made available to the general public.

Community cloud: Shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations).

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

Features of 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

(i) Self-service
(ii) Per-usage metered and billed
(iii) Elastic
(iv) Customizable

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

Per-Usage Metering and Billing: Cloud computing eliminates up-front commitment by users, allowing them to request and use only the necessary amount. Services must be priced on a short term basis (e.g., by the hour), allowing users to release (and not pay for) resources as soon as they are not needed.

Elasticity: Cloud computing gives the illusion of infinite computing resources available on demand. Therefore, users expect clouds to rapidly provide resources 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: resources rented from the cloud must be highly customizable. customization means allowing users to deploy specialized virtual appliances and to be given privileged (root) access to the virtual servers.

INFRASTRUCTURE AS A SERVICE PROVIDERS

Public Infrastructure as a Service provider 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 offerings can be distinguished by the availability of specialized features that 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 balancer firewalls)
(iv) Choice of virtualization platform and operating systems
(v) Different billing methods and period (e.g., prepaid vs. post-paid, hourly vs. monthly).

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

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

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

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

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.

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), Cloud front (content delivery), Cloud front Streaming (video streaming), SimpleDB (structured data store), RDS (Relational Database), SQS (reliable messaging), and Elastic Map Reduce (data processing). The Elastic Compute 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 in the 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 provider 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 petabytes 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 memcache 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 platform: a hosted Apex or Visualforce application. Apex is a proprietary Java-like language that can be used to create Salesforce applications. Visualforce 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 AppExchange 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.

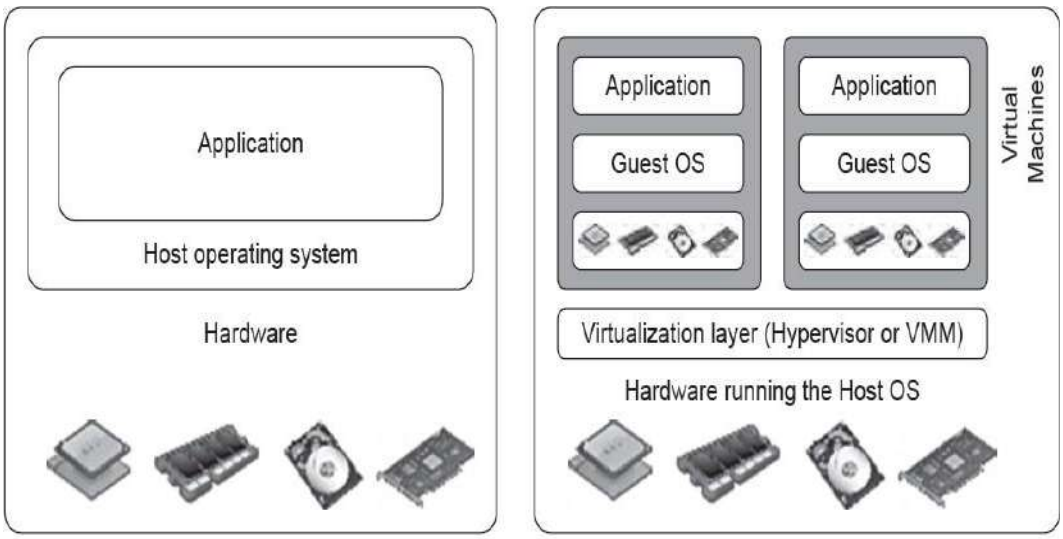
In addition, dynamic provisioning of new VMs and replicating existing VMs require efficient mechanisms to make VM block storage devices (e.g., image files) quickly available at selected hosts. Data centers consumer large amounts of electricity. According to a data published by HP, 100 server racks can consume 1.3MW of 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.

UNIT-II
VIRTUALIZATION

Implementation Levels of Virtualization

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine. The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility. The idea is to separate the hardware from the software to yield better system efficiency.

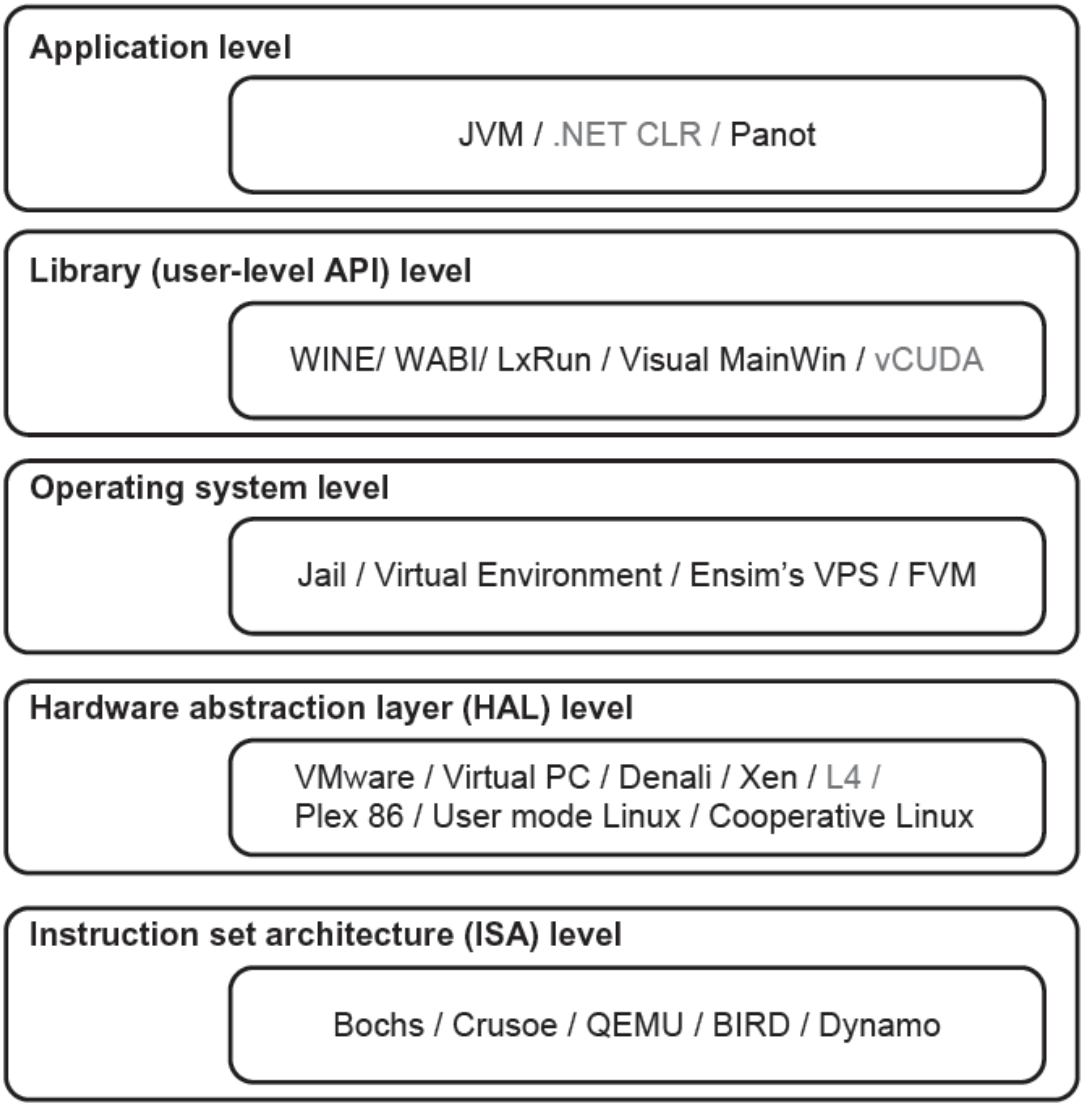
A traditional computer runs with a host operating system specially tailored for its hardware architecture, as shown in Figure. After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS. This is often done by adding additional software, called a virtualization layer as shown in Figure.



(a) Traditional computer (b) After virtualization

Virtualization can be implemented at various operational levels, as given below

- Instruction set architecture (ISA) level
- Hardware level
- Operating system level
- Library support level
- Application level



Instruction Set Architecture Level

At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. The basic emulation method is through code interpretation. An interpreter programinterprets the source instructions 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.A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

Hardware Abstraction Level

It is performed right on top of the bare hardware and generates a virtual hardware environment for a VM. On the other hand, the process manages the underlying hardware throughvirtualization. The idea is to virtualize a computer's resources, such as its processors, memory,and I/O devices so as hardware utilization rate by multiple users concurrently may be upgraded **Operating System Level.**

OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers. The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users.

Operating System Level

This refers to an abstraction layer between traditional OS and user applications. OS-level virtualization creates isolated *containers* on a single physical server and the OS instances to utilize the hardware and software in data centers. The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users. It is also used, to a lesser extent, in consolidating server hardware by moving services on separate hosts into containers or VMs on one server.

Library Support Level

Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks. The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts. Another example is the vCUDA which allows applications executing within VMs to leverage GPUhardware acceleration.

User-Application Level

On a traditional OS, an application often runs as a process. Therefore, application-level virtualization is also known as process-level virtualization. The most popular approach is to deploy high level language (HLL) VMs. In this scenario, the virtualization 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. Other forms of application-level virtualization are known as application isolation, application sandboxing, or application streaming. The process involves wrapping the application in a layer that is isolated from the host OS and other applications. The result is an applicationthat is much easier to distribute and remove from user workstations.

Virtualization Support at the OS Level

It is slow to initialize a hardware-level VM because each VM creates its own image from scratch and storage of such images are also slow. OS-level virtualization provides a feasible solution for these hardware-level virtualization issues. OS virtualization inserts a virtualization layer inside an operating system to partition a

machine'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). This VE has its own set of processes, file system, user accounts, network interfaces with IP addresses, routing tables, firewall rules, and other personal settings.

Advantages:

- VMs at the operating system level have minimal startup/shutdown costs, low resourcerequirements, and high Scalability
- It is possible for a VM and its host environment to synchronize state changes when necessary.

Virtualization Structures/Tools and Mechanisms

Before virtualization, the operating system manages the hardware. After virtualization, a virtualization layer is inserted between the hardware and the OS. In such a case, the virtualization layer is responsible for converting portions of the real hardware into virtual hardware. Depending on the position of the virtualization layer, there are several classes of VM architectures, namely

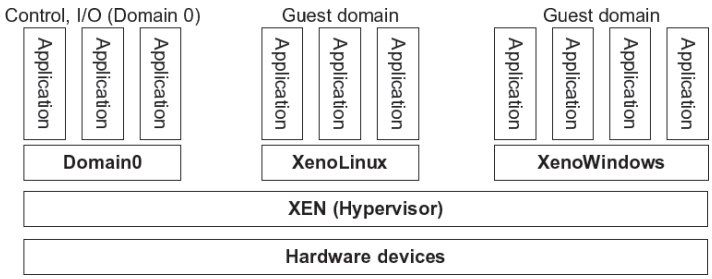
- Hypervisor architecture,
- Para virtualization
- host-based virtualization.

Hypervisor and Xen Architecture

Depending on the functionality, a hypervisor can assume a micro-kernel architecture or a monolithic hypervisor architecture. A micro-kernel hypervisor includes only the basic and unchanging functions (such as physical memory management and processor scheduling). The device drivers and other changeable components are outside the hypervisor. A monolithic hypervisor implements all the aforementioned functions, including those of the device drivers. Therefore, the size of the hypervisor code of a micro-kernel hypervisor is smaller than that of a monolithic hypervisor.

Xen Architecture

Xen is an open source hypervisor program developed by Cambridge University. Xen is a microkernel hypervisor, which separates the policy from the mechanism. It implements all the mechanisms, leaving the policy to be handled by Domain 0, as shown in Figure. Xen does not include any device drivers natively. It just provides a mechanism by which a guest OS can have direct access to the physical devices.



Like other virtualization systems, many guest OSes can run on top of the hypervisor. The guest OS (privileged guest OS), which has control ability, is called Domain 0, and the others are calledDomain U. It is first loaded when Xen boots without any file system drivers being available. Domain 0 is designed to access hardware directly and manage devices.

Binary Translation with Full Virtualization

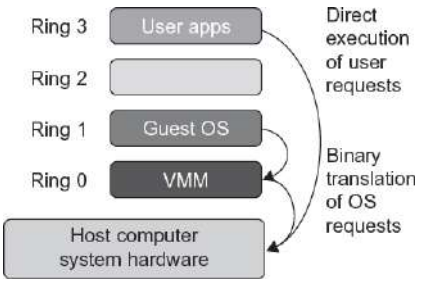
Depending on implementation technologies, hardware virtualization can be classified intotwo categories: full virtualization and host-based virtualization.

Full Virtualization

With full virtualization, noncritical instructions run on the hardware directly while criticalinstructions are discovered and replaced with traps into the VMM to be emulated by software.Both the hypervisor and VMM approaches are considered full virtualization. Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do. Therefore, running noncritical instructions on hardware not only can promote efficiency, but also can ensure system security.

Host-Based Virtualization

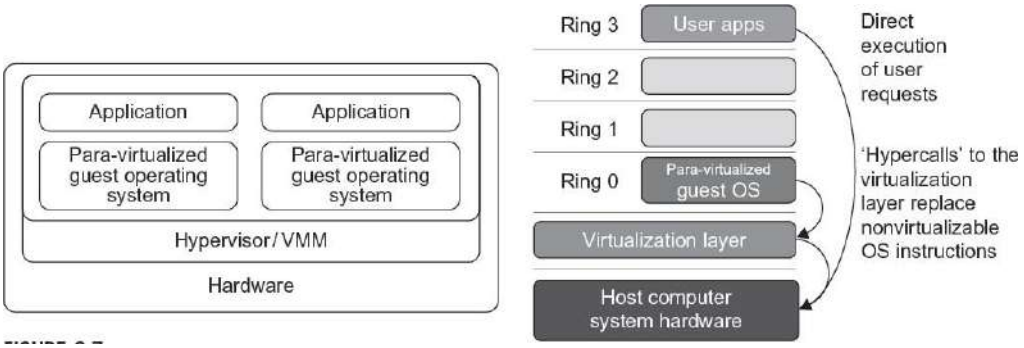
An alternative VM architecture is to install a virtualization layer on top of the host OS. This host OS is still responsible for managing the hardware. The guest OSes are installed and runon top of the virtualization layer. Dedicated applications may run on the VMs. Certainly, some other applications can also run with the host OS directly. This host based architecture has some distinct advantages, as enumerated next. First, the user can install this VM architecture without modifying the host OS. Second, the host-based approach appeals to many host machine configurations.



Para-Virtualization

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

degradation is a critical issue of a virtualized system. Figure illustrates the concept of a para-virtualized VM architecture. The guest OS are para-virtualized. They are assisted by an intelligent compiler to replace the non virtualizable OS instructions by hyper calls. The traditional x86 processor offers four instruction execution rings: Rings 0, 1, 2, and 3. The lower the ring number, the higher the privilege of instruction being executed. The OS is responsible for managing the hardware and the privileged instructions to execute at Ring 0, while user-level applications run at Ring 3.



Although para-virtualization reduces the overhead, it has incurred problems like compatibility and portability, because it must support the unmodified OS as well. Second, the cost is high, because they may require deep OS kernel modifications. Finally, the performance advantage of para-virtualization varies greatly due to workload variations.

Virtualization of CPU, Memory, And I/O Devices

To support virtualization, processors such as the x86 employ a special running mode and instructions, known as hardware-assisted virtualization. In this way, the VMM and guest OS run in different modes and all sensitive instructions of the guest OS and its applications are trapped in the VMM. To save processor states, mode switching is completed by hardware.

Hardware Support for Virtualization

Modern operating systems and processors permit multiple processes to run simultaneously. If there is no protection mechanism in a processor, all instructions from differentprocesses will access the hardware directly and cause a system crash. Therefore, all processors have at least two modes, user mode and supervisor mode, to ensure controlled access of critical hardware. Instructions running in supervisor mode are called privileged instructions. Otherinstructions are unprivileged instructions. In a virtualized environment, it is more difficult to make OSes and applications run correctly because there are more layers in the machine stack. Figure shows the hardware support by Intel.

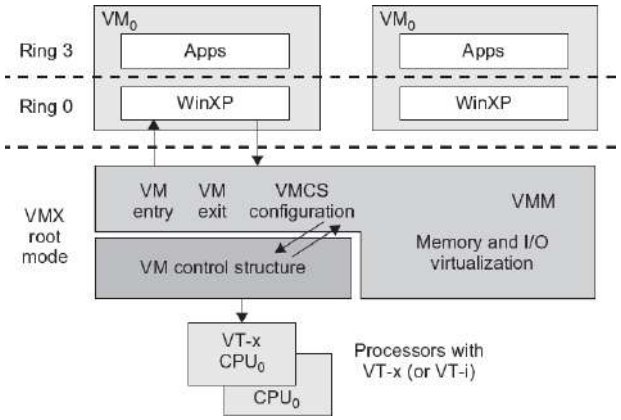
CPU Virtualization

Unprivileged instructions of VMs run directly on the host machine for higher efficiency. Other critical instructions should be handled carefully for correctness and stability. The critical instructions are divided into three categories: privileged instructions, controls sensitive instructions, and behavior-sensitive instructions. Privileged instructions execute in a privileged mode and will be trapped if executed outside this mode. Control-sensitive instructions attempt to change the configuration of resources used. Behavior-sensitive instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

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. When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM. RISC CPU architectures can be naturally virtualized because all control- and behavior-sensitive instructions are privileged instructions. On the contrary, x86 CPU architectures are not primarily designed to support virtualization.

Hardware-Assisted CPU Virtualization

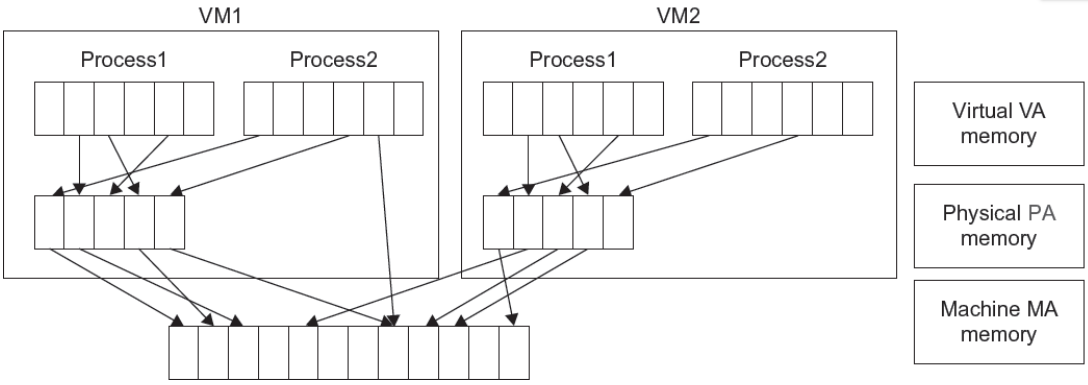
This technique attempts to simplify virtualization because full or para-virtualization is complicated. Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors. Therefore, operating systems can still run at Ring 0 and the hypervisor can run at Ring -1. All the privileged and sensitive instructions are trapped in the hypervisor automatically. This technique removes the difficulty of implementing binary translation of full virtualization. It also lets the operating system run in VMs without modification.



Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. In a traditional environment, the OS maintains page table for mappings of virtual memory to machine memory, which is a one-stage mapping. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance. However,

in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs. A 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 in guest OS.



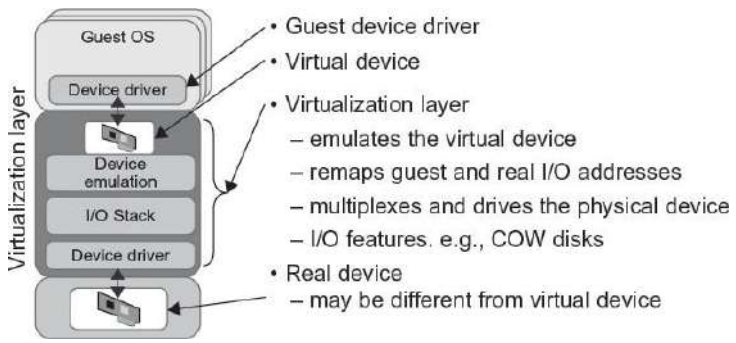
Since each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the shadow page table. VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation. Processors useTLB hardware to map the virtual memory directly to the machine memory to avoid the twolevels of translation on every access. When the guest OS changes the virtual memory to aphysical memory mapping, the VMM updates the shadow page tables to enable a direct lookup. **I/O virtualization.**

It involves managing the routing of I/O requests between virtual devices and the sharedphysical hardware. There are three ways to implement I/O virtualization:

- Full device emulation
- Para-virtualization
- Device I/O

Full device emulation

All the functions of a device like device enumeration, identification, interrupts, and DMA, are replicated in software and it is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.



Para-virtualization

It is a split driver model consisting of a frontend driver and a backend driver. Thefrontend driver is running in Domain U and the backend driver is running in Domain 0. They interact with each other via a block of shared memory. The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs. Although para-I/O-virtualization achieves better device performance than full device emulation, it comes with a higher CPU overhead.

Direct I/O virtualization

It lets the VM access devices directly. It can achieve close-to-native performance withouthigh CPU costs. However, current direct I/O virtualization implementations focus on networkingfor mainframes.

Another way to help I/O virtualization is via self-virtualized I/O (SV-IO). The key ideais to harness the rich resources of a multicore processor. All tasks associated with virtualizing an I/O device are encapsulated in SV-IO. SV-IO defines one virtual interface (VIF) for every kind of virtualized I/O device, such as virtual network interfaces, virtual block devices (disk), virtual camera devices, and others. The guest OS interacts with the VIFs via VIF device drivers. Each VIF consists of two message queues. One is for outgoing messages to the devices and the other isfor incoming messages from the devices. In addition, each VIF has a unique ID for identifying it in SV-IO.

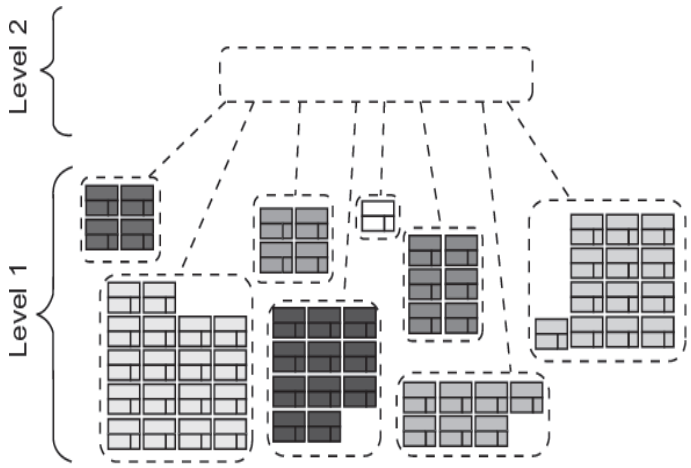
Virtualization in Multi-Core Processors

Multicore processors are claimed to have higher performance by integrating multiple processor cores in a single chip, multi-core virtualization has raised some new challenges to computer architects, compiler constructors, system designers, and application programmers.Application programs must be parallelized to use all cores fully, and software must explicitly assign tasks to the cores, which is a very complex problem. Concerning the first challenge, new programming models, languages, and libraries are needed to make parallel programming easier. The second challenge has spawned research involving scheduling algorithms and resource management policies

Virtual Hierarchy

A virtual hierarchy is a cache hierarchy that can adapt to fit the workload or mix of workloads. The hierarchy's first level locates data blocks close to the cores needing them for faster access, establishes a shared-cache domain, and establishes a point of coherence for faster communication. The first level can also provide isolation between independent workloads. A miss at the L1 cache can invoke the L2 access.

The following figure illustrates a logical view of such a virtual cluster hierarchy in two levels.



(b) Multiple virtual clusters assigned to various workloads

Each VM operates in an isolated fashion at the first level which minimize both miss access time and performance interference with other workloads or VMs. The second level maintains a globally shared memory facilitates dynamically repartitioning resources without costly cache flushes.

UNIT-III
FOUNDATIONS

INTRODUCTION TO CLOUD COMPUTING

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

Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service Level Agreements.

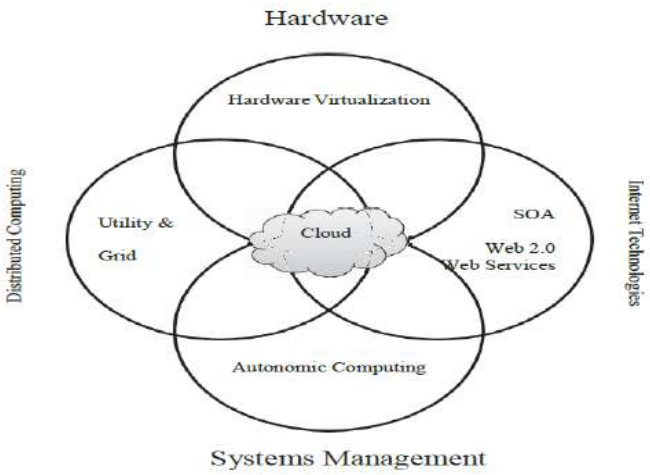
ROOTS OF CLOUD COMPUTING

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).

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 similar to 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.



Convergence of various advances leading to the advent of cloud computing

This model brings benefits to both consumers and providers of IT services. Consumers can attain reduction on IT-related costs by choosing to obtain cheaper services from external providers as opposed to heavily investing on IT infrastructure and personnel hiring.

The on-demand component of this model allows consumers to adapt their IT usage to rapidly increasing or unpredictable computing needs.

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

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

These facts reveal the potential of delivering computing services with the speed and reliability that businesses enjoy with their local machines. The benefits of economies of scale and high utilization allow providers to offer computing services for a fraction of what it costs for a typical company that generates its own computing power.

SOA, WEB SERVICES, WEB 2.0, AND MASHUPS

The emergence of Web services (WS) open standards has significantly contributed to advances in the domain of software integration. Web services can 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.

Over the years a rich WS software stack has been specified and standardized, resulting in a multitude of technologies to describe, compose, and orchestrate services, package and transport messages between services, publish and dis- cover services, represent quality of service (QoS) parameters, and ensure security in service access.

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.

The maturity of WS has enabled the creation of powerful services that can be accessed on-demand, in a uniform way. While some WS are published with the intent of serving end-user applications, their true power resides in its interface being accessible by other services. An enterprise application that follows the SOA paradigm is a collection of services that together perform complex business logic.

In the consumer Web, information and services may be programmatically aggregated, acting as building blocks of complex compositions, called *service mashups*. Many service providers, such as Amazon, del.icio.us, Facebook, and Google, make their service APIs publicly accessible using standard protocols such as SOAP and REST.

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

For example, Programmable Web is a public repository of service APIs and mashups currently listing thousands of APIs and mashups. Popular APIs such as Google Maps, Flickr, YouTube, Amazon eCommerce, and Twitter, when combined, produce a variety of interesting solutions, from finding video game retailers to weather maps. Similarly, Salesforce.com‘s

offers AppExchange, which enables the sharing of solutions developed by third-party developers on top of Salesforce.com components.

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 main focus being speeding up a broad range of scientific applications, such as climate modeling, drug design, and protein analysis.

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

UTILITY COMPUTING

In utility computing environments, users assign a utility value to their jobs, where utility is a fixed or time-varying valuation that captures various QoS constraints (deadline, importance, satisfaction). The valuation is the amount they are willing to pay a service provider to satisfy their demands. The service providers then attempt to maximize their own utility, where said utility may directly correlate with their profit. Providers can choose to prioritize

Hardware Virtualization

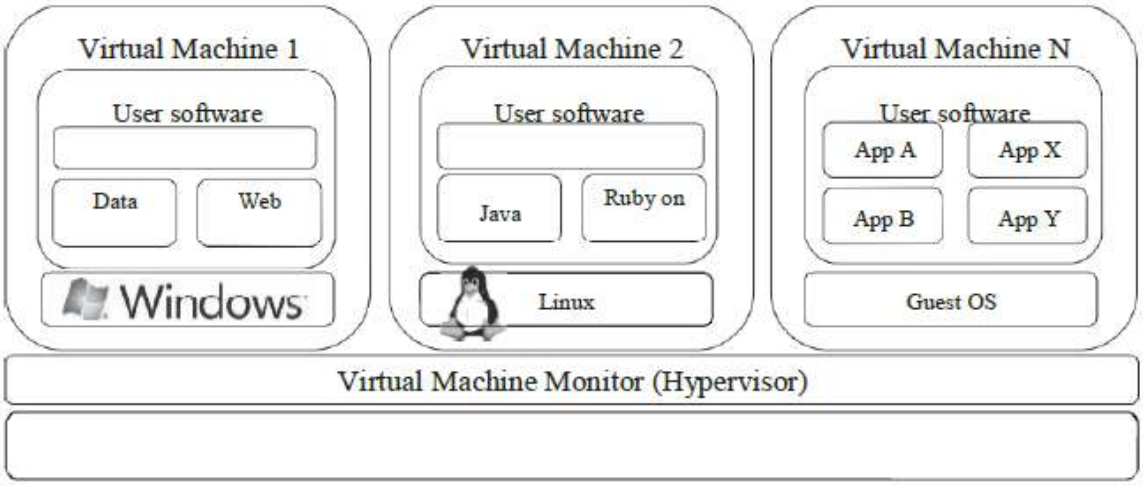
Cloud computing services are usually backed by large- scale data centers composed of thousands of computers. Such data centers are built to serve many users and host many disparate applications. For this purpose, hardware virtualization can be considered as a perfect fit to overcome most operational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

The idea of virtualizing a computer system's resources, including processors, memory, and I/O devices, has been well established for decades, aiming at improving sharing and utilization of computer systems.

Hardware virtualization allows running multiple operating systems and software stacks on a single 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.

The advent of several innovative technologies—multi- core chips, para virtualization, hardware-assisted virtualization, and live migration of VMs—has contributed to an increasing adoption of virtualization on server systems. Traditionally, perceived benefits were improvements on sharing and utilization, better manageability, and higher reliability.



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

Management of workload in a virtualized system, namely isolation, consolidation, and migration. Workload isolation is achieved since all program instructions are fully confined inside a VM, which leads to improvements in security. Better reliability is also achieved because software failures inside one VM do not affect others.

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

A number of VMM platforms exist that are the basis of many utility or cloud computing environments. The most notable ones, VMWare, Xen, and KVM.

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.

On-line marketplaces have been set up to allow the exchange of ready-made appliances containing popular operating systems and useful software combinations, both commercial and open-source.

Most notably, the VMWare virtual appliance marketplace allows users to deploy appliances on VMWare hypervisors or on partner's public clouds, and Amazon allows developers to share specialized Amazon Machine Images (AMI) and monetize their usage on Amazon EC2.

In a multitude of hypervisors, where each one supports a different VM image format and the formats are incompatible with one another, a great deal of interoperability issues arises. For instance, Amazon has its Amazon machine image (AMI) format, made popular on the Amazon EC2 public cloud.

Other formats are used by Citrix XenServer, several Linux distributions that ship with KVM, Microsoft Hyper-V, and VMware ESX.

AUTONOMIC COMPUTING

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

Autonomic, or self-managing, systems rely on monitoring probes and gauges (sensors), on an adaptation engine (autonomic manager) for computing optimizations based on monitoring data, and on effectors to carry out changes on the system. IBM's Autonomic Computing Initiative has contributed to define the four properties of autonomic systems: self-configuration, self- optimization, self-healing, and self-protection.

MIGRATING INTO A CLOUD

The promise of cloud computing has raised the IT expectations of small and medium enterprises beyond measure. Large companies are deeply debating it. Cloud computing is a disruptive model of IT whose innovation is part technology and part business model in short a disruptive techno-commercial model of IT.

We propose the following definition of cloud computing: —It is a techno-business disruptive model of using distributed large- scale data centers either private or public or hybrid offering customers a scalable virtualized infrastructure or an abstracted set of services qualified by service-level agreements (SLAs) and charged only by the abstracted IT resources consumed.

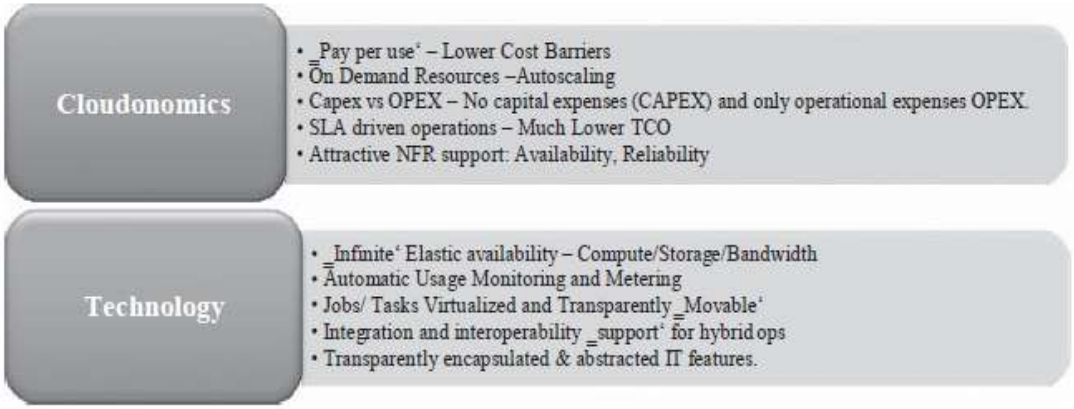


Figure The promise of the cloud computing services

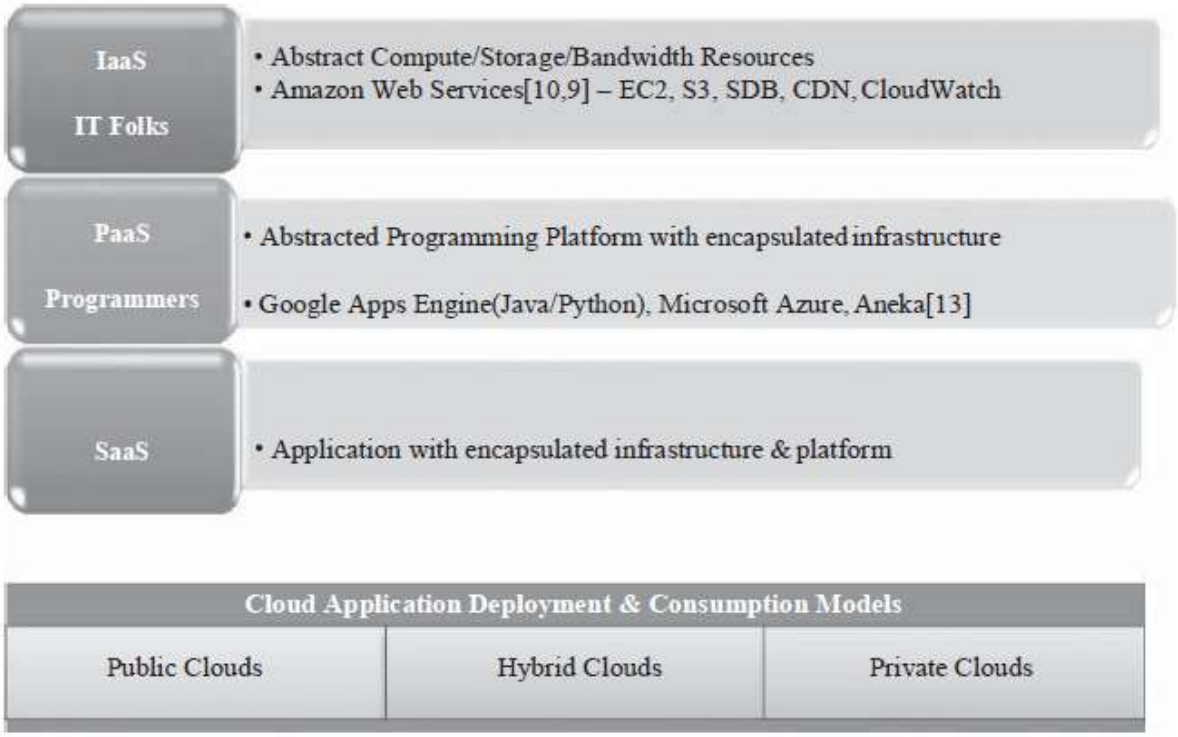
In the above Figure, the promise of the cloud both on the business front (the attractive cloudonomics) and the technology front widely aided the CxOs to spawn out several non-mission critical IT needs from the ambit of their captive traditional data centers to the appropriate cloud service.

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.

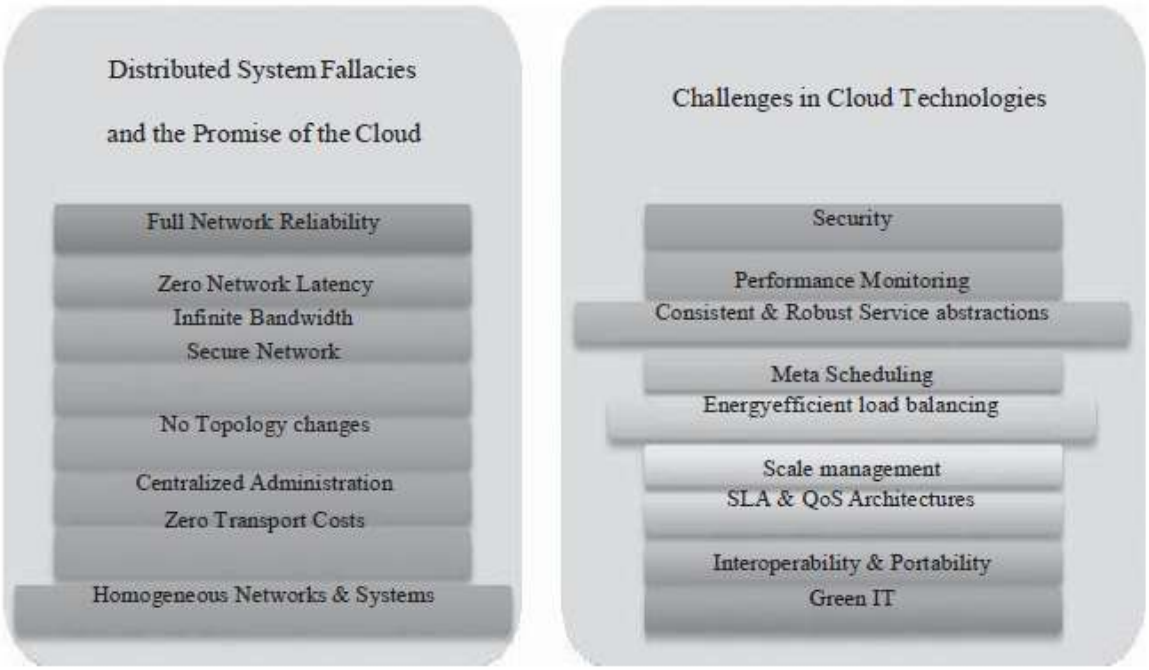
Many large enterprises run SAP to manage their operations. SAP itself is experimenting with running its suite of products: SAP Business One as well as SAP Netweaver on Amazon cloud offerings.

THE CLOUD SERVICE OFFERINGS AND DEPLOYMENT MODELS

Cloud computing has been an attractive proposition both for the CFO and the CTO of an enterprise primarily due its ease of usage. This has been achieved by large data center service vendors or now better known as cloud service vendors again primarily due to their scale of operations.



The cloud computing service offering and deployment models



'Under the hood' challenges of the cloud computing services implementations.

BROAD APPROACHES TO MIGRATING INTO THE CLOUD

Cloud Economics deals with the economic rationale for leveraging the cloud and is central to the success of cloud-based enterprise usage. Decision-makers, IT managers, and software architects are faced with several dilemmas when planning for new Enterprise IT initiatives.

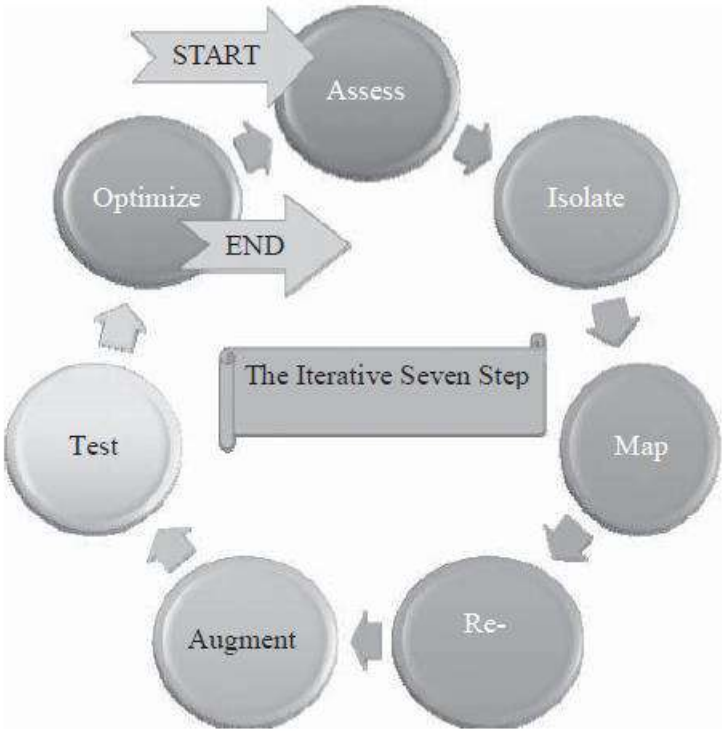
THE SEVEN-STEP MODEL OF MIGRATION INTO A CLOUD

Typically, migration initiatives into the cloud are implemented in phases or in stages. A structured and process-oriented approach to migration into a cloud has several advantages of capturing within itself the best practices of many migration projects.

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

The Seven-Step Model of Migration into the Cloud. (*Source:* Infosys Research.)



The iterative Seven-step Model of Migration into the Cloud. (*Source:* Infosys Research.)

Migration Risks and Mitigation

The biggest challenge to any cloud migration project is how effectively the migration risks are identified and mitigated. In the Seven-Step Model of Migration into the Cloud, the process step of testing and validating includes efforts to identify the key migration risks. In the optimization step, we address various approaches to mitigate the identified migration risks.

Migration risks for migrating into the cloud fall under two broad categories: the general migration risks and the security-related migration risks. In the former we address several issues including performance monitoring and tuning—essentially identifying all possible production level deviants; the business continuity and disaster recovery in the world of cloud computing service; the compliance with standards and governance issues; the IP and licensing issues; the quality of service (QoS) parameters as well as the corresponding SLAs committed to; the ownership, transfer, and storage of

data in the application; the portability and interoperability issues which could help mitigate potential vendor lock-ins; the issues that result in trivializing and non-comprehencing the complexities of migration that results in migration failure and loss of senior management‘s business confidence in these efforts.

THE ENTERPRISE CLOUD COMPUTING PARADIGM

Relevant Deployment Models for Enterprise Cloud Computing

There are some general cloud deployment models that are accepted by the majority of cloud stakeholders today:

1) Public clouds are provided by a designated service provider for general public under a utility based pay-per-use consumption model. The cloud resources are hosted generally on the service provider’s premises. Popular examples of public clouds are Amazon’s AWS (EC2, S3 etc.), Rackspace Cloud Suite, and Microsoft’s Azure Service Platform.

2) Private clouds are built, operated, and managed by an organization for its internal use only to support its business operations exclusively. Public, private, and government organizations worldwide are adopting this model to exploit the cloud benefits like flexibility, cost reduction, agility and so on.

3) Virtual private clouds are a derivative of the private cloud deployment model but are further characterized by an isolated and secure segment of resources, created as an overlay on top of public cloud infrastructure using advanced network virtualization capabilities. Some of the public cloud vendors that offer this capability include Amazon Virtual Private Cloud, OpSource Cloud, and Skytap Virtual Lab.

4) Community clouds are shared by several organizations and support a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). They may be managed by the organizations or a third party and may exist on premise or off premise. One example of this is OpenCirrus formed by HP, Intel, Yahoo, and others.

5) Managed clouds arise when the physical infrastructure is owned by and/or physically located in the organization’s data centers with an extension of management and security control plane controlled by the managed service provider. This deployment model is not widely agreed upon, however, some vendors like ENKI and NaviSite‘s NaviCloud offers claim to be managed cloud offerings.

6) Hybrid clouds are a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application.

UNIT-IV

INFRASTRUCTURE AS A SERVICE (IaaS)
&
PLATFORM AS A SERVICE (PaaS)

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

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

AS A SERVICE PROVIDERS

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

Features

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

The most relevant features are:

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

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

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

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

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

Amazon Reserved Instances is a form of advance reservation of capacity, allowing users to pay a fixed amount of money in advance to guarantee resource availability at any time 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 provider 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

1. 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.
2. 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.
3. Examples for vendors who publicly provide IaaS:
   - Amazon Elastic Compute Cloud (EC2).
   - GoGrid
   - Joyent Accelerator
   - Rackspace
   - AppNexus
   - FlexiScale and Manjrasoft Aneka
4. Amazon Elastic Compute Cloud (EC2) is an IaaS service that provides elastic compute capacity in the cloud.
5. 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.
6. 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.
7. Provides complete control of computing resources run on Amazon's computing and infrastructure environment easily
8. Reduces the time required for obtaining and booting a new server's instances to minutes
9. Allows a quick scalable capacity and resources, up and down as the computing requirements change Offers different instances' size according to
   - The resources' needs (small, large, and extra-large)
   - The high CPU's needs it provides (medium and extra-large high CPU instances)
   - High-memory instances (extra-large, double extra-large, and quadruple extra-large instance)
10. Amazon EC2 is a widely known example for vendors that provide public cloud services.
11. Eucalyptus and Open-Nebula are two complementary and enabling technologies for open source cloud tools, which play an invaluable role in

infrastructure as a service and in building private, public, and hybrid cloud architecture.

- The Amazon EC2 (Elastic Compute Cloud) is a Web service that allows users to provision new machines into Amazon's virtualized infrastructure in a matter of minutes using a publicly available API
- EC2 instance is typically a virtual machine with a certain amount of RAM, CPU, and storage capacity.

12. Amazon EC2 provides its customers with three flexible purchasing models to make it easy for the cost optimization.
    1. On-Demand instances: which allow you to pay a fixed rate by the hour with no commitment.
    2. Reserved instances: which allow you to pay a low, one-time fee and in turn receive a significant discount on the hourly usage charge for that instance. It ensures that any reserved instance you launch is guaranteed to succeed (provided that you have booked them in advance). This means that users of these instances should not be affected by any transient limitations in EC2 capacity.
    3. Spot instances: which enable you to bid whatever price you want for instance capacity, providing for even greater savings, if your applications have flexible start and end times.
13. Amazon Elastic Load Balancer is another service that helps in building fault-tolerant applications by automatically provisioning incoming application workload across available Amazon EC2 instances and in multiple availability zones.

Private Cloud and Infrastructure Services

A private cloud aims at providing public cloud functionality, but on private resources:

1. Maintaining control over an organization's data and resources to meet security and governance's requirements in an organization.
2. Private cloud exhibits a highly virtualized cloud data center located inside your organization's firewall.
3. 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:

1. Allow service provisioning and compute capability for an organization's users in a self-service manner.
2. Automate and provide well-managed virtualized environments.
3. Optimize computing resources, and servers' utilization.
4. Support specific workloads.

Examples for vendors and frameworks that provide Iaas in private setups

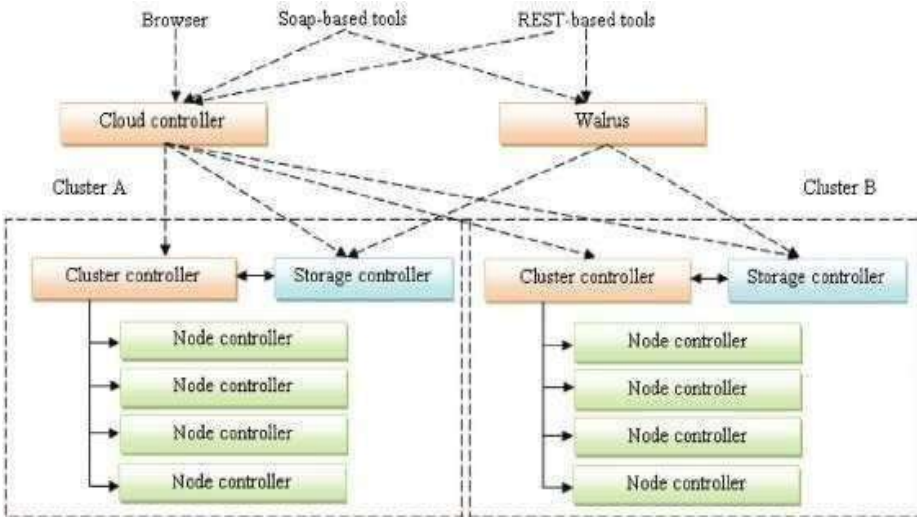
1. Eucalyptus (elastic utility computing architecture linking your programs to useful systems)
2. Open Nebula

Eucalyptus: Eucalyptus is an open-source infrastructure for the implementation of cloud computing on computer clusters. It is considered one of the earliest tools developed as a surge computing (in which data center's private cloud could augment its ability to handle workload's spikes by a design that allows it to send overflow work to a public cloud) tool. Its name is an acronym for **"elastic utility computing architecture for linking your programs to useful systems."**

Eucalyptus features:

1. Interface compatibility with EC2, and S3 (both Web service and Query/REST [Representational State Transfer] interfaces).
2. Simple installation and deployment.
3. Support for most Linux distributions (source and binary packages).
4. Support for running VMs that run atop the Xen hypervisor or KVM.
5. Support for other kinds of VMs, such as VMware, is targeted for future releases.
6. Secure internal communication using SOAP (Simple Object Access Protocol) with WS security.
7. Cloud administrator's tool for system's management and user's accounting.
8. The ability to configure multiple clusters each with private internal network addresses into a single cloud.
9. Eucalyptus aims at fostering the research in models for service's provisioning, scheduling, SLA formulation, and hypervisors' portability.

Eucalyptus Architecture:



1. **Node controller (NC)**controls the execution, inspection, and termination of VM instances on the host where it runs.
2. **Cluster controller (CC)** gathers information about and schedules VM execution on specific node controllers, as well as manages virtual instance network.
3. **Storage controller (SC)** is a put/get storage service that implements Amazon's S3(Simple Storage Service) interface and provides a way for storing and accessing VM images and user data.
4. Cloud controller (CLC) is the entry point into the cloud for users and administrators. It queries node managers for information about resources, makes high-level scheduling decisions, and implements them by making requests to cluster controllers.
5. Walrus (W) is the controller component that manages access to the storage services within Eucalyptus. Requests are being communicated to Walrus using the SOAP (Simple Object Access Protocol) or REST (Representational State Transfer) based interface

Hybrid Cloud and Infrastructure Services

A third type of cloud setup named Hybrid cloud

1. A combination of private/internal and external cloud resources existing together by enabling outsourcing of noncritical services and functions in public cloud and keeping the critical ones internal.
2. Main function of Hybrid cloud is to release resources from a public cloud and handle sudden demand usage called cloud bursting.

Distributed Management of Virtualization

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

OVF (Open Virtualization Format)

1. VMAN's OVF (Open Virtualization Format) in a collaboration between industry key players: Dell, HP, IBM, Microsoft, XenSource, and Vmware.
2. OVF specification provides a common format to package and securely distribute virtual appliances across multiple virtualization platforms.
3. VMAN profiles define a consistent way of managing a heterogeneous virtualized environment

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

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

VM 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:

1. 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.
2. 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).
3. Thirdly, you need to customize and configure the machine (e.g., IP address, Gateway) to configure an associated network and storage resources.
4. 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 the virtual machine's life cycle become a challenge.

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 are different 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

Migration Techniques:

- Stage 0: Pre-Migration
  - An active virtual machine exists on the physical host A
- Stage 1: Reservation
  - A request is issued to migrate an OS from host A to B.
  - The necessary resources exist on B and on a VM container of that size.
- Stage 2: Iterative Pre-Copy
  - During the first iteration, all pages are transferred from A to B
  - Subsequent iterations copy only those pages dirtied during the previous transfer phase
- Stage 3: Stop-and-Copy
  - Running OS instance at A is suspended
  - The network traffic is redirected to B
  - CPU state and any remaining inconsistent memory pages are then transferred
  - At the end of this stage, there is a consistent suspended copy of the VM at both A and B.
  - Copy at A is considered primary and is resumed in case of failure

- Stage 4: Commitment
    - Host B indicates to A that it has successfully received a consistent OS image
    - Host A acknowledges this message as a commitment of the migration transaction
    - Host A may now discard the original VM
    - Host B becomes the primary host
- Stage 5: Activation
    - The migrated VM on B is now activated

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.

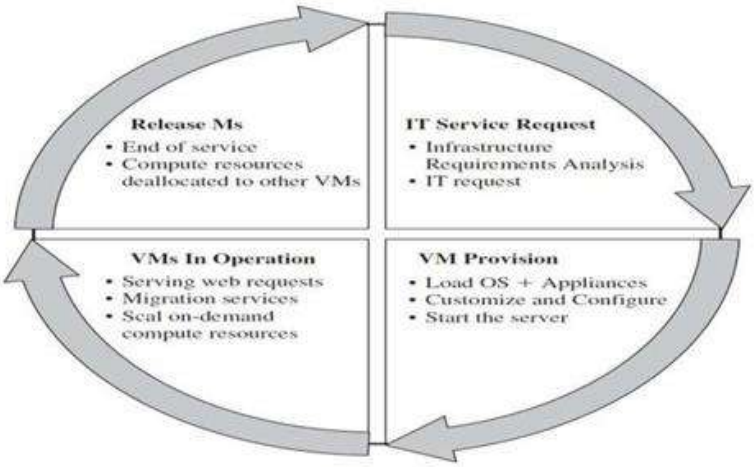
Migration of Virtual Machines to Alternate Platforms

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.

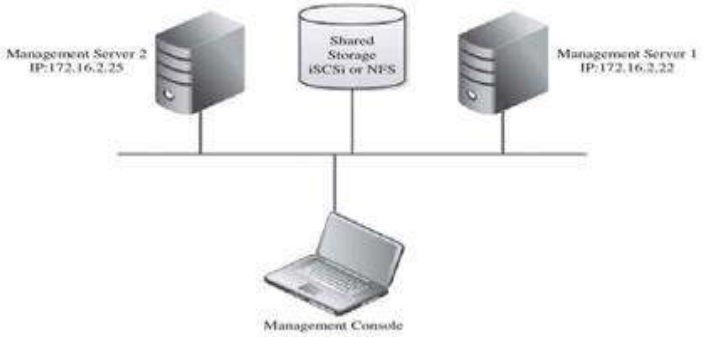
Here, we will use ConVirt (open source framework for the management of open source virtualization like Xen and KVM known previously as XenMan). Deployment Scenario. ConVirt deployment consists of at least one ConVirt workstation, where ConVirt is installed and ran, which provides the main console for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

There are two essential deployment scenarios for ConVirt:

1. Basic configuration in which the Xen or KVM virtualization platform is on the local machine, where ConVirt is already installed.

2. 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:

1. Installing ConVirt on at least one computer. See reference 28 for installation details.
2. 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).
3. Starting ConVirt and discovering the managed servers you have prepared.

Note:

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 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.
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 in house 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").

1. 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.

2. 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.

3. 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.

4. 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.

5. 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 for VMs, setting up and managing networking for interrelated VMs, and reducing the various overheads involved in using VMs.

1. 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.

2. 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.

3. 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:

1. SLA-oriented capacity planning that guarantees that there is enough capacity to guarantee service elasticity with minimal over-provisioning.
2. 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

1. 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.

2. 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.

3. 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.

4. Such applications are executed on the Aneka cloud, represented by a collection of nodes connected through the network hosting the Aneka container.

UNIT-V
SOFTWARE AS A SERVICE (SAAS)
&
DATA SECURITY IN THE CLOUD

SAAS

• SaaS is a model of software deployment where an application is hosted as a service provided to customers across the Internet.

• Saas alleviates the burden of software maintenance/support but users relinquish control over software versions and requirements

SaaS Maturity Model

Level 1: Ad-Hoc/Custom – One Instance per customer

Level 2: Configurable per customer

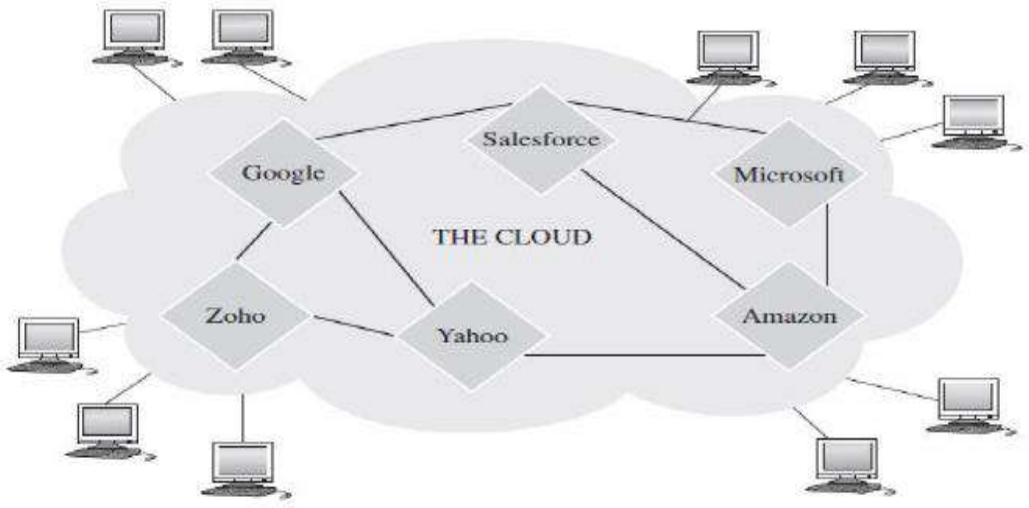
Level 3: configurable & Multi-Tenant-Efficient

Level 4: Scalable, Configurable & Multi-Tenant-Efficient

SaaS INTEGRATION PRODUCTS AND PLATFORMS

• Cloud-centric integration solutions are being developed and demonstrated for showcasing their capabilities for integrating enterprise and cloud applications.

• Composition and collaboration will become critical and crucial for the mass adoption of clouds.



The Smooth and Spontaneous Cloud Interaction via OpenClouds

Jitterbit:

• Jitterbit is a fully graphical integration solution that provides users a versatile platform

• suite of productivity tools to reduce the integration efforts sharply.

• Jitterbit can be used standalone or with existing EAI infrastructures

• Help us quickly design, implement, test, deploy, and manage the integration projects

Two major components:

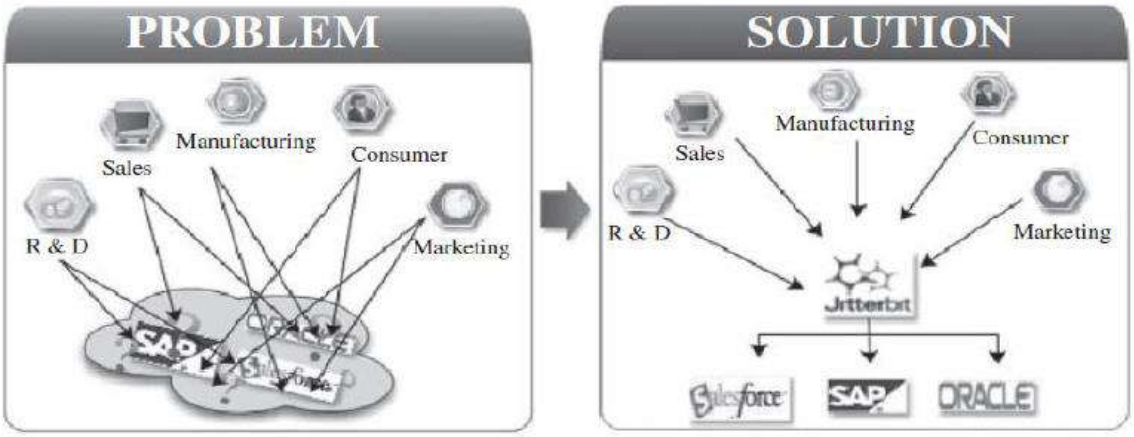
• Jitterbit Integration Environment

• An intuitive point-and-click graphical UI that enables to quickly configure, test, deploy and manage integration projects on the Jitterbit server.

• Jitterbit Integration Server

• A powerful and scalable run-time engine that processes all the integration operations, fully configurable and manageable from the Jitterbit application.

Linkage with On premise and on demand Applications



Linkage of On-Premise with Online and On-Demand Applications.

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, so it's not surprising that the company also offers cloud development services.

- The 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.

Features of App Engine

- 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.
- Data storage, retrieval, and search include features such as HRD migration tool, Google Cloud SQL, logs, datastore, dedicated Memcache, blobstore, Memcache and search.
- Communications include features such as XMPP. channel, URL fetch, mail, and Google Cloud Endpoints.
- Process management includes features like scheduled tasks and task queue
- Computation includes images.
- 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.

- Pre cloud 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 problems with 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; you can access it from any computer connected to the Internet.

Collaborating via Web-Based Communication Tools

GMAIL

- Gmail offers a few unique features that set it apart from the web-based email crowd.
- First, Gmail doesn't use folders. With Gmail you can't organize your mail into folders, as you can with the other services.
- 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.
- 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.
- In addition, Gmail groups together related email messages in what Google calls conversations

Yahoo! Mail Yahoo! Mail (mail.yahoo.com)

- It's another web mail service, provided by the popular Yahoo! search site.
- The basic Yahoo! Mail is free and can be accessed from any PC, using any web browser.
- 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

- 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 from a multitude of nodes and geographic locations and, as such, must have a data security 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

CLOUD COMPUTING AND 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 of data.

Identity, Reputation, and Trust

- Reputation is a real-world commodity; that is a basic requirement of human-to-human relationships
- Our basic societal communication structure is built upon the idea of reputation and trust.
- Reputation and its counter value, trust, is easily transferable to a digital realm:
    - 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.
- These types of reputation systems can be extremely useful when used with a digital identity.
- 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:

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

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

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

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

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:

- 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.
- 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.

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

- **Identity providers (IdP**), who provision and manage information cards, with specific claims, to users.
- **Users** who own and utilize the cards to gain access to Web sites and other resources that support information cards.
- **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.
- **Relying parties.** These are the applications, services, and 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.

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

- Information cards are built around a set of open standards devised by a consortium that includes Microsoft, IBM, Novell, and so on.
- 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.
- However, the information card system can be used in many more ways.
- 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.

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

- 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 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 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.

Data-centric mashups are those

- 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 to alleviate 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.