WEEK-2 (CLOUD COMPUTING)

ARCHITECTURE – DEPLOYMENT MODELS

1. Deployment Models • Public Cloud • Private Cloud • Hybrid Cloud • Community Cloud

**PUBLIC CLOUD**

* Cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them.
* Examples of Public Cloud: Google App Engine Microsoft Windows Azure IBM Smart Cloud Amazon EC2
* In Public setting, the provider's computing and storage resources are potentially large; the communication links can be assumed to be implemented over the public Internet; and the cloud serves a diverse pool of clients (and possibly attackers).

FEATURES:

* Workload locations are hidden from clients (public): subscriber don’t know where their data is located in the cloud server. The providers continuously relocate the data without the knowledge of the user.
* Risks from multi-tenancy (public): – A single machine may be shared by the workloads of any combination of subscribers
* Network dependency (public): • Subscribers connect to providers via the public Internet.
* Limited visibility and control over data regarding security (public): – The details of provider system operation are usually considered proprietary information and are not divulged to subscribers.
* Elasticity: illusion of unlimited resource availability (public): – Public clouds are generally unrestricted in their location or size. The size can infinitely be increased by paying the price of the storage theoritaclly but technically its not possible to have infinite amount of storage.
* Low up-front costs to migrate into the cloud (public) • Restrictive default service level agreements (public): – The default service level agreements of public clouds specify limited promises that providers make to subscribers

**Private Cloud**

* The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
* Examples of Private Cloud: – Eucalyptus – Ubuntu Enterprise Cloud - UEC – Amazon VPC (Virtual Private Cloud) – VMware Cloud Infrastructure Suite – Microsoft ECI data center.
* Types:

Contrary to popular belief, private cloud may exist off premises and can be managed by a third party. Thus, two private cloud scenarios exist, as follows: • On-site Private Cloud – Applies to private clouds implemented at a customer’s premises. • Outsourced Private Cloud – Applies to private clouds where the server side is outsourced to a hosting company.

Features: (On-site Private Cloud)

* Organizations considering the use of an on-site private cloud should consider: – Network dependency (on-site-private): –
* Subscribers still need IT skills (on-site-private): • Subscriber organizations will need the traditional IT skills required to manage user devices that access the private cloud, and will require cloud IT skills as well. –
* Workload locations are hidden from clients (on-site-private): • To manage a cloud's hardware resources, a private cloud must be able to migrate workloads between machines without inconveniencing clients.
* Risks from multi-tenancy (on-site-private): – Workloads of different clients may reside concurrently on the same systems and local networks, separated only by access policies implemented by a cloud provider's software.
* Data import/export, and performance limitations (on-site-private): – On-demand bulk data import/export is limited by the on-site private cloud's network capacity, and real-time or critical processing may be problematic because of networking limitations.
* Potentially strong security from external threats (on-site-private):
* Significant-to-high up-front costs to migrate into the cloud (on-site-private): – An on-site private cloud requires that cloud management software be installed on computer systems within a subscriber organization.
* Limited resources (on-site-private): – An on-site private cloud, at any specific time, has a fixed computing and storage capacity that has been sized to correspond to anticipated workloads and cost restrictions.

Features: (Outsource Private Cloud)

* Outsourced private cloud has two security perimeters, one implemented by a cloud subscriber (on the right) and one implemented by a provider. • Two security perimeters are joined by a protected communications link. • The security of data and processing conducted in the outsourced private cloud depends on the strength and availability of both security perimeters and of the protected communication link.
* Network Dependency (outsourced-private):
* Workload locations are hidden from clients (outsourced-private):
* Risks from multi-tenancy (outsourced-private):
* Data import/export, and performance limitations (outsourced-private):
* Potentially strong security from external threats (outsourced-private):

Important point –

Modest-to-significant up-front costs to migrate into the cloud (outsourced private): – In the outsourced private cloud scenario, the resources are provisioned by the provider – Main start-up costs for the subscriber relate to: • Negotiating the terms of the service level agreement (SLA) • Possibly upgrading the subscriber's network to connect to the outsourced private cloud • Switching from traditional applications to cloud-hosted applications, • Porting existing non-cloud operations to the cloud • Training

**Community Cloud**

* Community Cloud Cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
* Examples of Community Cloud: Google Apps for Government Microsoft Government Community Cloud

**Features of On-site community Cloud**:

Community cloud is made up of a set of participant organizations. Each participant organization may provide cloud services, consume cloud services, or both • At least one organization must provide cloud services • Each organization implements a security perimeter

* Network Dependency (on-site community):
* Subscribers still need IT skills (on-site-community)
* Workload locations are hidden from clients (on-site-community):
* Data import/export, and performance limitations (on-site-community): – The communication links between the various participant organizations in a community cloud can be provisioned to various levels of performance, security and reliability, based on the needs of the participant organizations. The network-based limitations are thus similar to those of the outsourced-private cloud scenario.
* Potentially strong security from external threats (on-site-community):
* Highly variable up-front costs to migrate into the cloud (on-site community):

Features with Outsourced Community Cloud:

* Network dependency (outsourced-community)
* Workload locations are hidden from clients (outsourced community).
* Risks from multi-tenancy (outsourced-community):
* Data import/export, and performance limitations (outsourced community)
* Potentially strong security from external threats (outsourced community):
* Modest-to-significant up-front costs to migrate into the cloud (outsourced-community):
* Extensive resources available (outsourced-community).

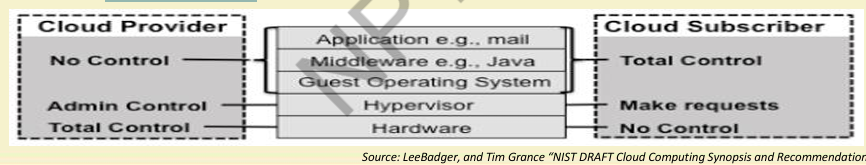
**Hybrid Cloud**

The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability • Examples of Hybrid Cloud: – Windows Azure (capable of Hybrid Cloud) – VMware vCloud (Hybrid Cloud Services)

* Hybrid Cloud • A hybrid cloud is composed of two or more private, community, or public clouds. • They have significant variations in performance, reliability, and security properties depending upon the type of cloud chosen to build hybrid cloud.
* A hybrid cloud can be extremely complex • A hybrid cloud may change over time with constituent clouds joining and leaving.

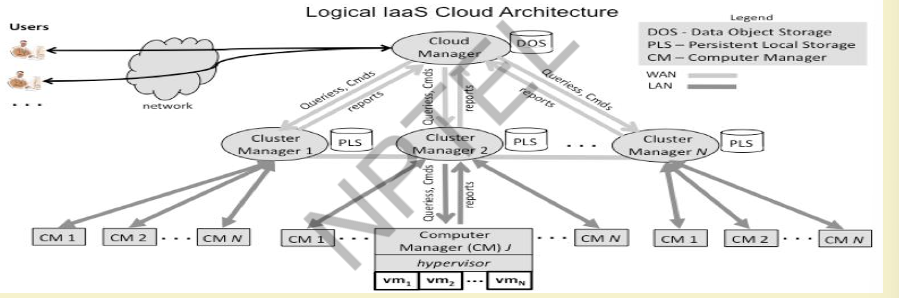
V2:

**CLOUD COMPUTING Virtualization**

* **IaaS – Infrastructure as a Service**
* IaaS (Infrastructure as a Service) refers to a cloud computing model where a service provider delivers fundamental computing resources, such as servers, storage, and networking, to customers over the internet. These resources can be accessed on-demand and are typically billed based on usage.
* IaaS component stack comprises of hardware, operating system, middleware, and applications layers
* Operating system layer is split into two layers. – Lower (and more privileged) layer is occupied by the Virtual Machine Monitor (VMM), which is also called the Hypervisor
* Higher layer is occupied by an operating system running within a VM called a guest operating system
* 

The IaaS component stack typically consists of four primary layers:

* **Hardware:** This includes the physical infrastructure, such as servers, storage devices, networking equipment, and data centers.
* **Operating System:** This layer provides the foundation for running applications and other software.
* **Middleware:** This layer acts as a bridge between the operating system and applications, providing services like application servers, database management systems, and message queues.
* **Applications:** This layer includes the software applications that users interact with, such as web applications, databases, and custom-built software.
* **IaaS Cloud Architecture**

****

Three-level hierarchy of components in IaaS cloud systems:

– Top level is responsible for **central control**

– Middle level is responsible for management of possibly large computer clusters that may be geographically distant from one another

– Bottom level is responsible for running the host computer systems on which virtual machines are created.

• Subscriber queries and commands generally flow into the system at the top and are forwarded down through the layers that either answer the queries or execute the commands

* **Cloud Manager Operation**: It serves as the central point of interaction between cloud subscribers and the cloud infrastructure.
* Cloud Manager is the public access point to the cloud where subscribers sign up for accounts, manage the resources they rent from the cloud, and access data stored in the cloud.
* • Cloud Manager has mechanism for:
* – Authenticating subscribers
* – Generating or validating access credentials that subscriber uses when communicating with VMs. –
* --Top-level resource management.
* For a subscriber’s request cloud manager determines if the cloud has enough free resources to satisfy the request
* Data Object Storage (DOS)
* DOS generally stores the subscriber’s metadata like user credentials, operating system images. • DOS service is (usually) single for a cloud.
* Operation of the Cluster Managers:
* Each Cluster Manager is responsible for the operation of a collection of computers that are connected via high speed local area networks
* • Cluster Manager receives resource allocation commands and queries from the Cloud Manager, and calculates whether part or all of a command can be satisfied using the resources of the computers in the cluster.
* • Cluster Manager queries the Computer Managers for the computers in the cluster to determine resource availability, and returns messages to the Cloud Manager
* Each Cluster Manager is connected to Persistent Local Storage (PLS) • PLS provide persistent disk-like storage to Virtual Machine
* Operation of the Computer Managers:

At the lowest level in the hierarchy computer manger runs on each computer system and uses the concept of virtualization to provide Virtual Machines to subscribers

• Computer Manger maintains status information including how many virtual machines are running and how many can still be started

• Computer Manager uses the command interface of its hypervisor to start, stop, suspend, and reconfigure virtual machines

**VIRTUALIZATION**

Virtualization • Virtualization is a broad term (virtual memory, storage, network, etc) • Focus: Platform virtualization • Virtualization basically allows one computer to do the job of multiple computers, by sharing the resources of a single hardware across multiple environments

• Virtualization is way to run multiple operating systems and user applications on the same hardware – E.g., run both Windows and Linux on the same laptop • How is it different from dual-boot? – Both OSes run simultaneously • The OSes are completely isolated from each other

**Types of Hypervisors**

* **Type 1 (Bare-Metal):** This type of hypervisor is installed directly on the physical hardware, bypassing the need for an underlying operating system. Examples include VMware ESXi, KVM, and Microsoft Hyper-V.
* **Type 2 (Hosted):** This type of hypervisor runs as an application within a host operating system. Examples include Oracle VM VirtualBox, VMware Workstation, and Parallels Desktop.

**Relationship between VMM and VM:**

* **VMM creates and manages VMs:** The VMM is responsible for creating, configuring, and running multiple VMs on a single physical server.
* **VMs rely on VMM:** VMs depend on the VMM for access to hardware resources and for isolation from other VMs.
* **VMs provide a virtual environment:** VMs offer a complete computing environment that can be used to run applications and services.

**Approaches to Server Virtualization:**

**Full Virtualization**

Full virtualization is a technique used in server virtualization where a virtual machine (VM) is created to emulate a complete physical computer system. This emulation includes the CPU, memory, storage, network interface, and other hardware components.

Key Characteristics:

* Hardware Emulation: The hypervisor (or VMM) creates a virtual environment that mimics the behavior of real hardware.
* Guest OS Independence: The guest operating system running on the VM can be completely different from the host operating system.
* Full Control: The guest OS has full control over its virtual resources, as if it were running on a physical machine.

Benefits:

* Flexibility: Allows running various operating systems and applications on a single physical server.
* Isolation: Ensures that VMs are isolated from each other, preventing interference.
* Compatibility: Supports a wide range of hardware and software.

Drawbacks:

* Performance Overhead: Emulation can introduce some performance overhead, especially for CPU-intensive tasks.
* Complexity: Full virtualization can be more complex to implement and manage compared to other virtualization techniques.

Common Use Cases:

* Testing and Development: Creating isolated environments for testing new software or hardware.
* Server Consolidation: Running multiple servers on a single physical machine to reduce costs and improve efficiency.
* Disaster Recovery: Creating virtual replicas of physical servers for backup and recovery purposes.

**Para-Virtualization**

**Para-virtualization** is another approach to server virtualization that offers a more efficient and performant alternative to full virtualization. Instead of emulating the entire physical hardware environment, para-virtualization modifies the guest operating system to work directly with the hypervisor.

**Key Characteristics:**

* **Guest OS Modification:** The guest operating system must be modified to support para-virtualization. This typically involves replacing or modifying device drivers to interact directly with the hypervisor.
* **Direct Access:** Para-virtualization allows the guest OS to have direct access to hardware resources, bypassing the emulation layer used in full virtualization.
* **Reduced Overhead:** By eliminating the need for emulation, para-virtualization can offer significantly better performance than full virtualization, especially for CPU-intensive workloads.

**Benefits:**

* **Improved Performance:** Lower overhead and direct access to hardware resources result in better performance.
* **Efficiency:** More efficient use of system resources.
* **Compatibility:** While it requires guest OS modifications, para-virtualization can still support a wide range of operating systems.

**Drawbacks:**

* **Guest OS Modifications:** The need for guest OS modifications can limit the compatibility with certain operating systems or applications.
* **Complexity:** Para-virtualization can be more complex to implement and manage than full virtualization.

**Common Use Cases:**

* **High-Performance Computing:** Para-virtualization is often used in environments that require high levels of performance, such as scientific computing or financial modeling.
* **Server Consolidation:** Consolidate multiple servers onto a single physical machine to reduce costs and improve efficiency.
* **Cloud Computing:** Para-virtualization is commonly used in cloud environments to provide efficient and scalable virtual machines

**Hardware-assisted virtualization**

• Guest OS runs at ring 0

• VMM uses processor extensions (such as Intel® VT or AMD-V) to intercept and emulate privileged operations in the guest

• Hardware-assisted virtualization removes many of the problems that make writing a VMM a challenge

• VMM runs in a more privileged ring than 0, a Virtual-1 ring is created.

Hardware-assisted virtualization Server virtualization

• Pros – It allows to run unmodified OSs (so legacy OS can be run without problems)

• Cons – Speed and Flexibility

• An unmodified OS does not know it is running in a virtualized environment and so, it can’t take advantage of any of the virtualization features – It can be resolved using para-virtualization partially

Network virtualization:

It is a networking environment that allows multiple service providers to dynamically compose multiple heterogeneous virtual networks that co-exist together in isolation from each other, and to deploy customized end-to end services on-the-fly as well as manage them on those virtual networks for the end-users by effectively sharing and utilizing underlying network resources leased from multiple infrastructure providers.

Typical Approach

• Networking technology – IP, ATM

• Layer of virtualization

• Architectural domain – Network resource management, Spawning networks

• Level of virtualization – Node virtualization, Full virtualization