UNIT I

CLOUD ARCHITECTURE AND MODEL

Introduction

TECHNOLOGIES FOR NETWORK-BASED SYSTEMS

It will focus on viable approaches to building distributed operating systems for handling massive parallelism in a distributed environment.

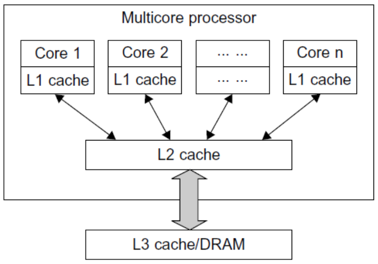
Multicore CPUs and Multithreading Technologies

The growth of component and network technologies over the past 30 years, are crucial to the development of HPC and HTC systems.

Advances in CPU Processors

Today, advanced CPUs or microprocessor chips assume a multicore architecture with dual, quad, six, or more processing cores. These processors exploit parallelism at ILP and TLP levels. However, the clock rate reached its limit on CMOS-based chips due to power limitations. At the time of this writing, very few CPU chips run with a clock rate exceeding 5 GHz. In other words, clock rate will not continue to improve unless chip technology matures. This limitation is attributed primarily to excessive heat generation with high frequency or high voltages. The ILP is highly exploited in modern CPU processors which include multiple-issue superscalar architecture, dynamic branch prediction, and speculative execution, among others. In addition, DLP and TLP are highly explored in graphics processing units (GPUs) that adopt a many-core architecture with hundreds to thousands of simple cores.

Both multi-core CPU and many-core GPU processors can handle multiple instruction threads at different magnitudes today. Figure shows the architecture of a typical multicore processor. Each core is essentially a processor with its own private cache (L1 cache).



Multiple cores are housed in the same chip with an L2 cache that is shared by all cores. In the future, multiple CMPs could be built on the same CPU chip with even the L3 cache on the chip. Multicore and multithreaded CPUs are equipped with many high-end processors.

Multicore CPU and Many-Core GPU Architectures

Multicore CPUs may increase from the tens of cores to hundreds or more in the future. But the CPU has reached its limit in terms of exploiting massive DLP due to the aforementioned memory wall problem. This has triggered the development of many-core GPUs with hundreds or more thin cores. The GPU also has been applied in large clusters to build supercomputers in MPPs. In the future, the processor industry is also keen to develop asymmetric or heterogeneous chip multiprocessors that can house both fat CPU cores and thin GPU cores on the same chip.

Multithreading Technology

Typical instruction scheduling patterns are shown here. Only instructions from the same thread are executed in a superscalar processor. Fine-grain multithreading switches the execution of instructions from different threads per cycle. Course-grain multithreading executes many instructions from the same thread for quite a few cycles before switching to another thread. The multicore CMP executes instructions from different threads completely. The SMT allows simultaneous scheduling of instructions from different threads in the same cycle.

GPU (Graphics processing unit)

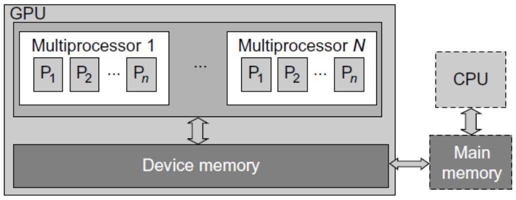
A GPU is a graphics coprocessor or accelerator mounted on a computer's graphics card or video card. A GPU offloads the CPU from tedious graphics tasks in video editing applications. Traditional CPUs are structured with only a few cores. However, a modern GPU chip can be built with hundreds of processing cores. Unlike CPUs, GPUs have a throughput architecture that exploits massive parallelism by executing many concurrent threads slowly, instead of executing a single long thread in a conventional microprocessor very quickly.

How GPUs Work

Early GPUs functioned as coprocessors attached to the CPU. Today, the NVIDIA GPU has been upgraded to 128 cores on a single chip. Furthermore, each core on a GPU can handle eight threads of instructions. This translates to having up to 1,024 threads executed concurrently on a single GPU. This is true massive parallelism, compared to only a few threads that can be handled by a conventional CPU. Modern GPUs are not restricted to accelerated graphics or video coding. They are used in HPC systems to power supercomputers with massive parallelism at multicore and multithreading levels. GPUs are designed to handle large numbers of floating-point operations in parallel.

GPU Programming Model

Figure shows the interaction between a CPU and GPU in performing parallel execution of floating-point operations concurrently. The CPU is the conventional multicore processor with limited parallelism to exploit. The GPU has a many-core architecture that has hundreds of simple processing cores organized as multiprocessors. Each core can have one or more threads. Essentially, the CPU's floating-point kernel computation role is largely offloaded to the many-core GPU. The CPU instructs the GPU to perform massive data processing. The bandwidth must be matched between the on-board main memory and the on-chip GPU memory.



MEMORY, STORAGE, AND WIDE-AREA NETWORKING

Memory Technology

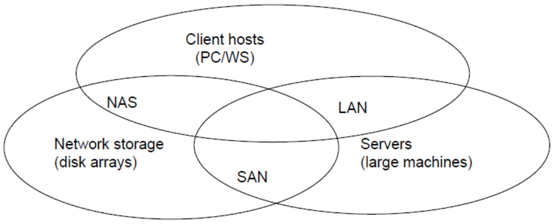
Memory chips have experienced a 4x increase in capacity every three years. The capacity increase of disk arrays will be even greater in the years to come. Faster processor speed and larger memory capacity result in a wider gap between processors and memory

Disks and Storage Technology

The rapid growth of flash memory and solid-state drives (SSDs) also impacts the future of HPC and HTC systems. Eventually, power consumption, cooling, and packaging will limit large system development. Power increases linearly with respect to clock frequency and quadratic ally with respect to voltage applied on chips. Clock rate cannot be increased indefinitely. Lowered voltage supplies are very much in demand.

System-Area Interconnects

A storage area network (SAN) connects servers to network storage such as disk arrays. Network attached storage (NAS) connects client hosts directly to the disk arrays.

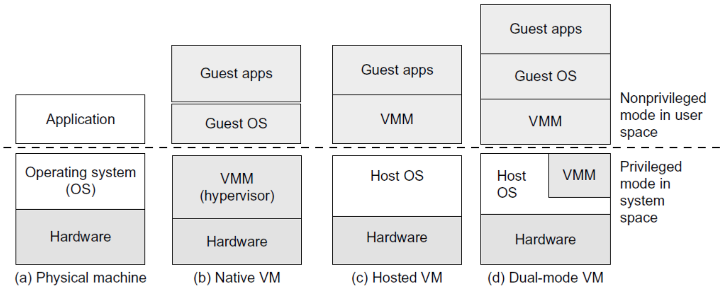


Wide-Area Networking

High-bandwidth networking increases the capability of building massively distributed systems. Most data centers are using Gigabit Ethernet as the interconnect in their server clusters.

Virtual Machines and Virtualization Middleware

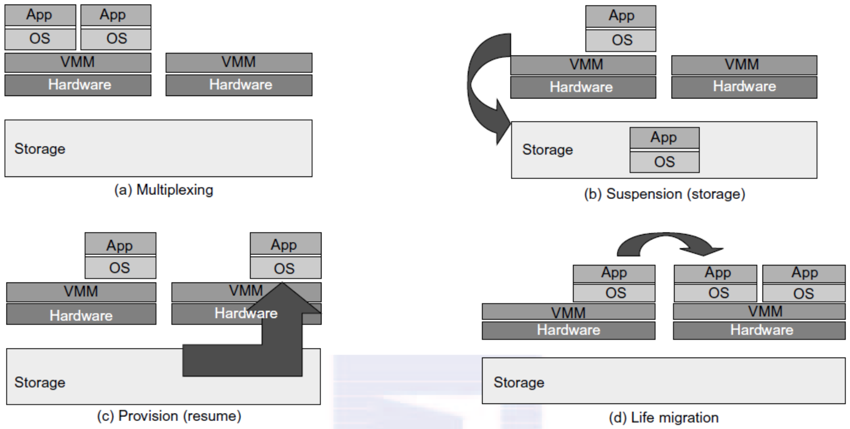
Virtual machines (VMs) offer novel solutions to underutilized resources, application inflexibility, software manageability, and security concerns in existing physical machines. To build large clusters, grids, and clouds, we need to access large amounts of computing, storage, and networking resources in a virtualized manner. We need to aggregate those resources, and hopefully, offer a single system image. The VM is built with virtual resources managed by a guest OS to run a specific application. Between the VMs and the host platform, one needs to deploy a middleware layer called a virtual machine monitor (VMM). The VMM called a hypervisor in privileged mode.



VM Primitive Operations

The VMM provides the VM abstraction to the guest OS. Low-level VMM operations are indicated below

- First, the VMs can be multiplexed between hardware machines
- Second, a VM can be suspended and stored in stable storage
- Third, a suspended VM can be resumed or provisioned to a new hardware platform
- Finally, a VM can be migrated from one hardware platform to another



(a) Multiplexing (b) Suspension (storage)

(c) Provision (resume) (d) Life migration

The VM approach ill significantly enhance the utilization of server resources. Multiple server functions can be consolidated on the same hardware platform to achieve higher system efficiency.

Virtual Infrastructures

Virtual infrastructure is connects resources like compute, storage, and networking to distributed applications. It is a dynamic mapping of system resources to specific applications. The result is decreased costs and increased efficiency and responsiveness.

Data Center Virtualization for Cloud Computing

Data center design emphasizes the performance/price ratio over speed performance alone. In other words, storage and energy efficiency are more important than shear speed performance.

Data Center Growth and Cost Breakdown

A large data center may be built with thousands of servers. Smaller data centers are typically built with hundreds of servers. The cost to build and maintain data center servers has increased over the years. According to a 2009 IDC report , typically only 30 percent of data center costs goes toward purchasing IT equipment (such as servers and disks), 33 percent is attributed to the chiller, 18 percent to the uninterruptible power supply (UPS), 9 percent to computer room air conditioning (CRAC), and the remaining 7 percent to power distribution, lighting, and transformer costs. Thus, about 60 percent of the cost to run a data center is allocated to management and maintenance.

Low-Cost Design

Given a fixed budget, commodity switches and networks are more desirable in data centers. Similarly, using x86 servers is more desired over expensive mainframes. The software layer handles network traffic balancing, fault tolerance, and expandability. Currently, nearly all cloud computing data centers use Ethernet as their fundamental network technology.

Convergence of Technologies

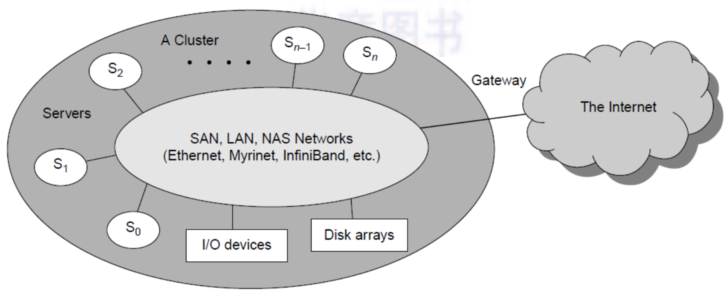
cloud computing is enabled by the convergence of technologies in four areas: (1) hardware virtualization and multi-core chips, (2) utility and grid computing, (3) SOA, Web 2.0, and WS mashups, and (4) autonomic computing and data center automation. The cloud provides services on demand at the infrastructure, platform, or software level. At the platform level, MapReduce offers a new programming model that transparently handles data parallelism with natural fault tolerance capability.

SYSTEM MODELS FOR DISTRIBUTED AND CLOUD COMPUTING

Massive systems are considered highly scalable, and can reach web-scale connectivity, either physically or logically. They are classified into four groups: clusters, P2P networks, computing grids, and Internet clouds over huge data centers.

Cluster Architecture

A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource. Figure shows the architecture of a typical server cluster built around a low-latency, high bandwidth interconnection network. 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. Most clusters have loosely coupled node computers.



Single-System Image

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. A cluster with multiple system images is nothing but a collection of independent computers

Hardware, Software, and Middleware Support

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. 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. 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.

Major Cluster Design Issues

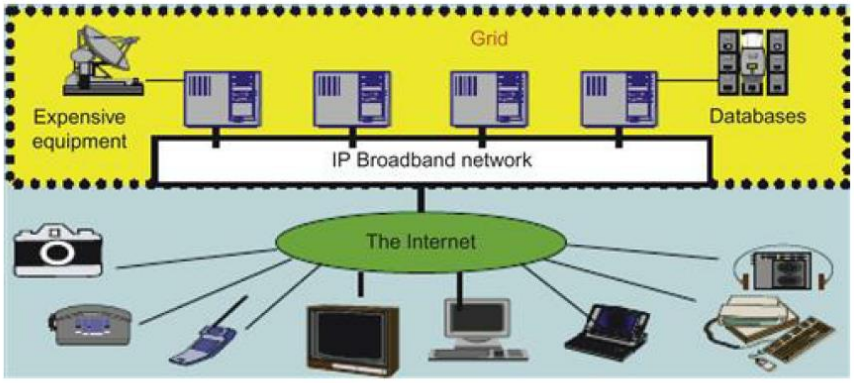
- scalable performance
- efficient message passing,
- high system availability
- seamless fault tolerance
- cluster-wide job management

Grid computing

Grid computing is envisioned to allow close interaction among applications running on distant computers simultaneously.

Computational Grids

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. 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 shows an example computational grid built over multiple resource sites owned by different organizations.



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 integrated resources pool as shown in the upper half of the figure. At the client end, wired or wireless terminal devices are present. The grid integrates the computing, communication, contents, and transactions as rented services.

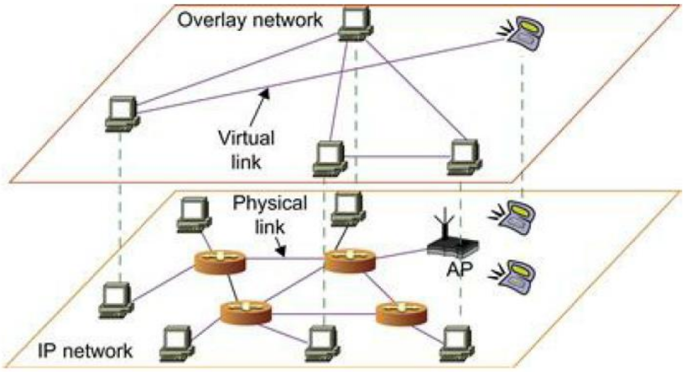
Peer-to-Peer Network

The P2P architecture offers a distributed model of networked systems. A P2P network is client-oriented instead of server-oriented.

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.

The architecture of a P2P network at two abstraction levels is shown in the figure. 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.



Overlay Networks

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 as shown in Figure. 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 as shown in the above Figure.

There are two types of overlay networks:

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.

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.

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.

Other P2P platforms, such as JXTA, .NET, and FightingAID@home, support naming, discovery, communication, security, and resource aggregation in some P2P applications.

Computing challenges in P2P

There are 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; and different network connections and protocols make it too complex to apply in real applications. We need system scalability 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. P2P performance is affected by routing efficiency and self-organization by participating peers. Fault tolerance, failure management, and load balancing are other important issues in using overlay networks. Lack of trust among peers poses another problem. 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.

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.

NIST Architecture

National Institute of Standards and Technology (NIST) provides architecture intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing. It exhibits characteristics of cloud like on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

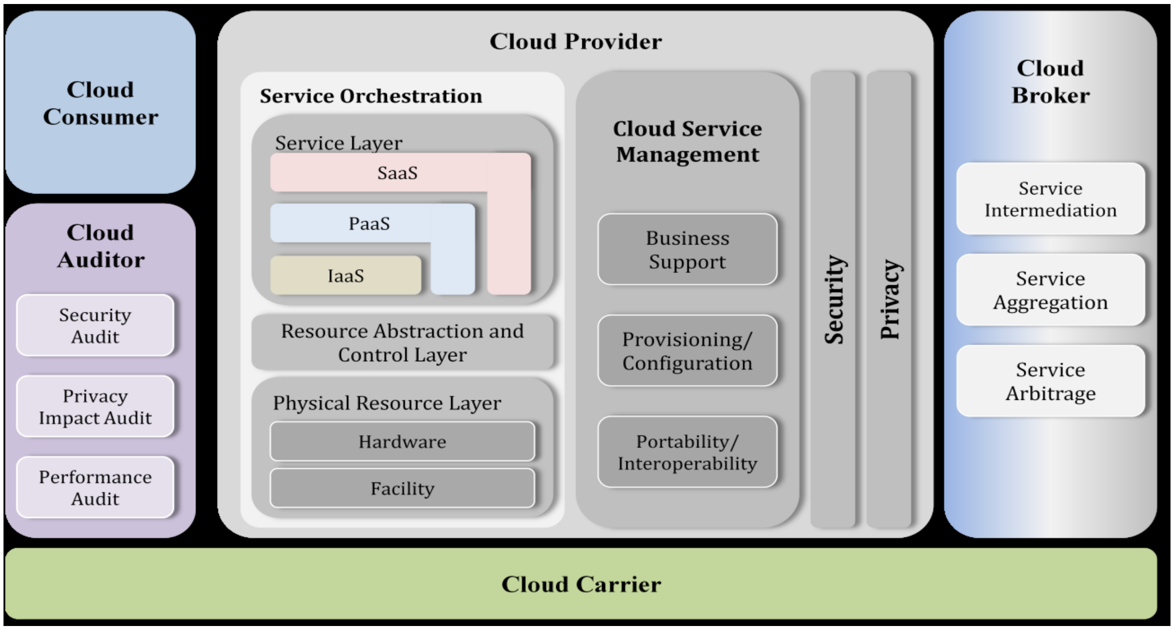
Services Models:

- Software as a Service (Saas)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

Deployment Model:

- Public Cloud
- Private Cloud
- Hybrid Cloud
- Community cloud

The following figure shows the architecture defined by NIST. There are five major actors in the cloud system that includes *cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker*. Each entity has its own roles or functionalities in the cloud to provide uninterrupted services to the customers from anywhere.



Cloud Consumer:

A person or organization that maintains a business relationship with, and uses service from Cloud Providers.

- In SaaS, consumer can access software applications
- In PaaS, can be application developers, tester, deployer or administrator
- In IaaS, access to virtual computers, network-accessible storage, network infrastructure components, and other fundamental computing resources

Cloud Provider:

A person, organization, or entity responsible for making a service available to interested parties. In PaaS, It manages the computing infrastructure for the platform and runs the cloud software that provides the components of the platform

Cloud Carrier:

An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

Cloud Broker:

An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers

Cloud Auditor

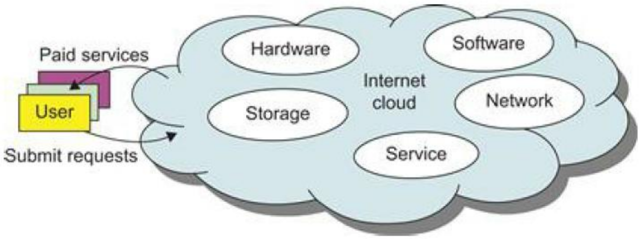
A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

Cloud Computing

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. a cloud allows workloads to be deployed and scaled out quickly through rapid provisioning of virtual or physical machines. The cloud supports redundant, selfrecovering, highly scalable programming models that allow workloads to recover frommany unavoidable hardware/software failures.

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.

Cloud Characteristics:

On demand self services: Users are able to provision cloud computing resources without requiring human interaction, mostly done though a web-based self-service portal.

Broad network access: Cloud computing resources are accessible over the network, supporting heterogeneous client platforms such as mobile devices and workstations

Resource pooling: Service multiple customers from the same physical resources, by securely separating the resources on logical level. Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

Rapid elasticity: Resources are provisioned and released on-demand and/or automated based on triggers or parameters. This will make sure your application will have exactly the capacity it needs at any point of time.

Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction. Resource usage are monitored, measured, and reported (billed) transparently based on utilization. In short, pay for use.

Multi Tenacity: It refers to the need for policy-driven enforcement, segmentation, isolation, governance, service levels, and chargeback/billing models for different consumer constituencies

Cloud models

Infrastructure as a Service (IaaS)

This model allows users to use virtualized IT resources for computing, storage, and networking. In short, the service is performed by rented cloud infrastructure. The user can deploy and run his applications over his chosen OS environment. The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components. This IaaS model encompasses storage as a service, compute instances as a service, and communication as a service.

To deploy the applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed.

Example: Amazon EC2, GoGrid, Aneka, FlexiScale

Platform as a Service (PaaS)

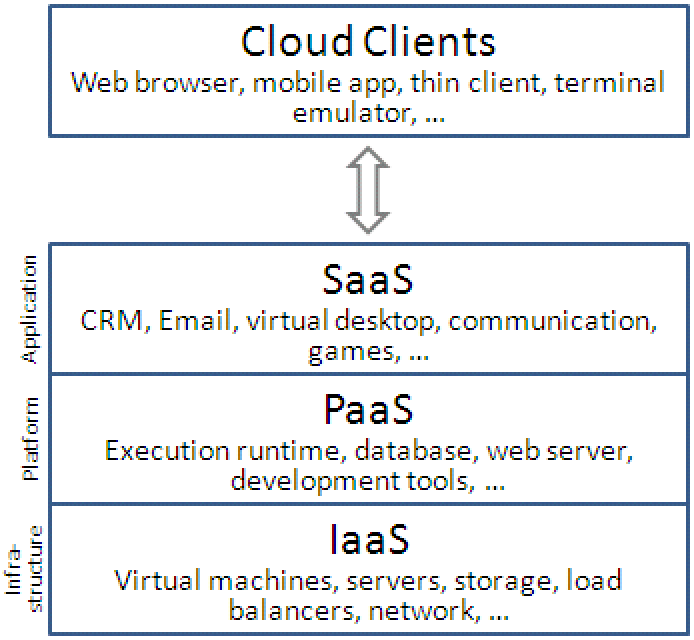
To be able to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment. Such a platform includes operating system and runtime library support. The platform cloud is an integrated computer system consisting of both hardware and software infrastructure. The user application can be developed on this virtualized cloud platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET). The user does not manage the underlying cloud infrastructure. The cloud provider supports user application development and testing on a well-defined service platform.

Example: Google App Engine, Microsoft Azure, Salesforce.com, Amazon elastic MapReduce

Software as a Service (SaaS)

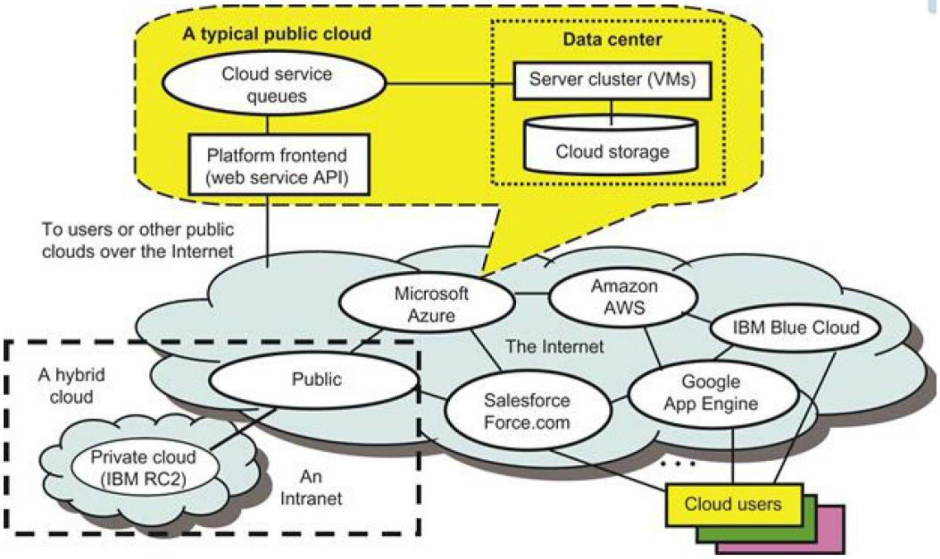
This refers to browser-initiated application software over thousands of cloud customers. The SaaS model provides software applications as a service. As a result, on the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are kept rather low, compared with conventional hosting of user applications. It eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. The pricing model for SaaS applications is typically a monthly or yearly flat fee per user.

Example: Google Gmail, Google Docs, Microsoft SharePoint, CRM Software of Salesforce.com



Public Clouds

A public cloud is built over the Internet and can be accessed by any user who has paid for the service. Public clouds are owned by service providers and are accessible through a subscription. Figure 4.1 shows the architecture of a typical public cloud. Many public clouds are available, including Google App Engine (GAE), Amazon Web Services (AWS), Microsoft Azure, IBM Blue Cloud, and Salesforce.com. A public cloud delivers a selected set of business processes. The application and infrastructure services are offered on a flexible price-peruse basis.



Private Clouds

A private cloud is built within the domain of an intranet owned by a single organization. Therefore, it is client owned and managed, and its access is limited to the owning clients and their partners. Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains. A private cloud is supposed to deliver more efficient and convenient cloud services. It may impact the cloud standardization, while retaining greater customization and organizational control.

Private clouds can be more expensive compared with public cloud. So, this is not usually not an option for small-to-medium sized business and typically suitable for large enterprises.

Hybrid Clouds

A hybrid cloud is built with both public and private clouds. Private clouds can also support a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud. It provides access to clients, the partner network, and third parties. It is also called “surge computing,” a public cloud can be used to perform tasks that can be deployed easily on a public cloud.

Public Vs Private Cloud

Public

- Generally speaking, a public cloud consists of a service or set of services that are purchased by a business or organization and delivered via the Internet by a third-party provider.
- These are based on shared physical hardware which is owned and operated by third-party providers
- the primary benefits of the public cloud are the speed with which you can deploy IT resources, and the utility billing it offers
- By spreading infrastructure costs are across a number of users, each can operate on a low-cost, pay as you go approach to the provisioning of IT services.

Private

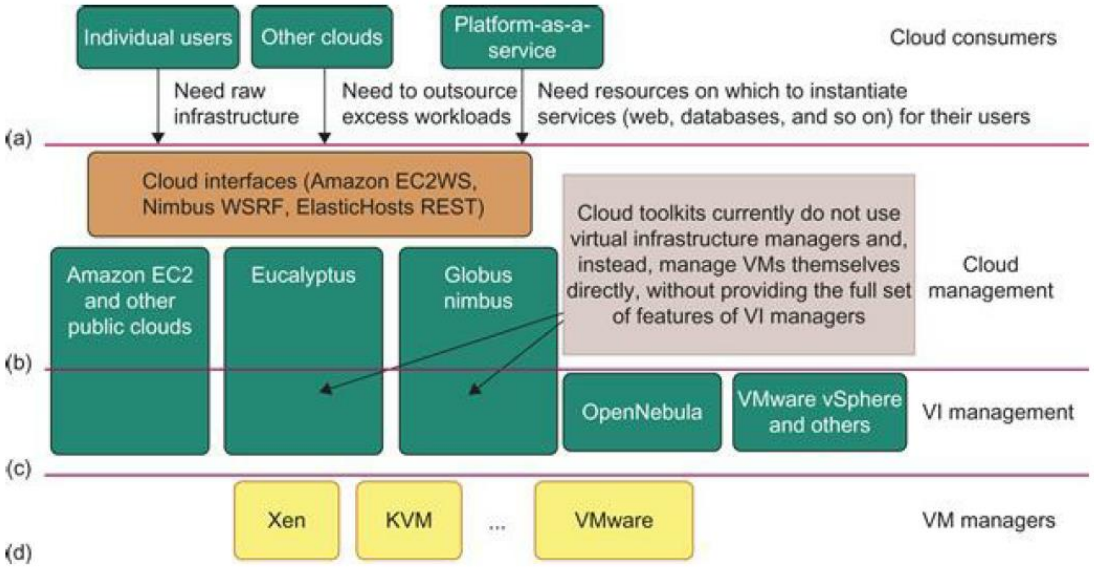
- It is the infrastructure purely dedicated to your business, hosted either on-site or at a service provider data centre.
- It is ideal for larger businesses or those with strict data, regulation and governance obligations
- The private cloud delivers all the agility, scalability and efficiency of the public cloud, but in addition provides greater levels of control and security
- Another key benefit of private cloud is the ability to customise the compute, storage and networking components to best suit your specific IT requirements

	Public Cloud	**Private Cloud**
Pros	• *High scalability* • *Cost effectiveness* • *Increased reliability* • *Low Maintenance*	• *Enhanced Security* • *More control over resources* • *Higher performance* • *flexibility*
Cons	• *Lack of control*	• *Scalability*

	• *Regulatory* • *Slow speeds*	• *On-site maintenance*

Cloud Ecosystems

With the emergence of various Internet clouds, an ecosystem of providers, users, and technologies has appeared. This ecosystem has evolved around public clouds. Strong interest is growing in open source cloud computing tools that let organizations build their own IaaS clouds using their internal infrastructures. Private and hybrid clouds are not exclusive, since public clouds are involved in both cloud types. A private/hybrid cloud allows remote access to its resources over the Internet using remote web service interfaces such as that used in Amazon EC2.



(a) Consumers demand a flexible platform
(b) Cloud manager provides virtualized resources over an IaaS platform;
(c) VI manager allocates VMs;
(a) VM managers handle VMs installed on servers

There are four levels of ecosystem development in a private cloud. At the user end, consumers demand a flexible platform. At the cloud management level, the cloud manager provides virtualized resources over an IaaS platform. At the virtual infrastructure (VI) management level, the manager allocates VMs over multiple server clusters. Finally, at the VM management level, the VM managers handle VMs installed on individual host machines. An ecosystem of cloud tools attempts to span both cloud management and VI management.

VI tools support dynamic placement and VM management on a pool of physicalresources, automatic load balancing, server consolidation, and dynamic infrastructure resizing and partitioning. In addition to public clouds such as Amazon EC2, Eucalyptus and Globus Nimbus are open source tools for virtualization of cloud infrastructure.

Service Management

Cloud service management is processes, activities and methods that are generated by cloud providers by taking cloud consumer perspective as a measure of service assurance. Cloud service management is supported by three support services: architecture service, business support and operational support. Cloud management tools help providers administrate the systems and applications that facilitate the on-demand service delivery model. The goal of these practices is to improve the efficiency of the cloud environment and achieve a high level of customer satisfaction. The core elements of cloud service management mirror those of traditional ITSM -- including

- Cloud service-level agreement (SLA) management
- Cloud capacity management
- Availability management
- Billing.

These processes are supported with tools that track provisioning and change management, configuration management, release management, incident management, perfor mance management and service continuity.

Challenges

- Event correlation
- Incident prioritization
- Capacity management
- Performance management
- Visibility remains a common challenge in managing highly elastic and complex virtual systems

Computing on demand (ODC)

It is an enterprise-level model of technology and computing in which resources are provided on an as-needed and when-needed basis. ODC make computing resources such as storage capacity, computational speed and software applications available to users as and when needed for specific temporary projects, known or unexpected workloads, routine work, or long-term technological and computing requirements.

Advantages:

- Low initial cost
- Virtually unlimited resource pool
- Instant access of resources, stand up new infrastructure in minutes
- Self-service provisioning
- Dynamically scalable
- Flexibility of multiple virtual data centers

Cloud Services

Cloud services means services made available to users on demand via the Internet from a cloud computing provider's servers. Cloud services are designed to provide easy, scalable access to applications, resources and services, and are fully managed by a cloud services provider. A cloud service can dynamically scale to meet the needs of its users, and because the service provider supplies the hardware and software necessary for the service, there’s no need for a company to provision or deploy its own resources or allocate IT staff to manage the service. Examples of cloud services include online data storage and backup solutions, Web-based e-mail services, hosted office suites and document collaboration services, database processing, managed technical support services and more.