INDEX		
UNIT	**Topic**	**Page No**
I	Scalable computing over the Internet	1
	System Models for Distributed and Cloud Computing	6
	Software Environments for distributed Systems and Clouds	11
	Performance, Security & Energy Efficiency	13
II	Implementation Levels of Virtualization	18
	Virtualization Structures/Tools and Mechanisms	21
	Virtualization of CPU, Memory, and I/O Devices	23
	Virtual Clusters and Data Centers	26

UNIT -1

Scalable Computing over the Internet

High-Throughput Computing-HTC

HTC paradigm pays more attention to high-flux computing. The main application for high-flux computing is in Internet searches and web services by millions or more users simultaneously. The performance measures high throughput or the number of tasks completed per unit of time. HTC technology needs to improve batch processing speed, and also address the acute problems of cost, energy savings, security, and reliability at many data and enterprise computing centers

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.
  - Many data centers and supercomputers are centralized systems, but they are used in parallel, distributed, and cloud computing applications.

• **Parallel computing**

- In parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory
- Inter Process communication is accomplished through shared memory or via message passing.
- A computer system capable of parallel computing is commonly known as a parallel computer
- Programs running in a parallel computer are called parallel programs. The process of writing parallel programs is often referred to as parallel programming

• **Distributed computing**

- A distributed system consists of multiple autonomous computers, each having its own private memory, communicating through a computer network.

- Information exchange in a distributed system is accomplished through message passing.
- A computer program that runs in a distributed system is known as a distributed program.
- The process of writing distributed programs is referred to as distributed programming.
- Distributed computing system uses multiple computers to solve large-scale problems overthe Internet using a centralized computer to solve computational problems.

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.
- Cloud computing can also be a form of utility computing or service computing

Degrees of Parallelism

- **Bit-level parallelism** (BLP) :
  - converts bit-serial processing toword-level processing gradually.
- **Instruction-levelparallelism** (ILP)
  - the processor executes multiple instructions simultaneously rather thanonly one instruction at a time.
  - ILP is executed through pipelining, superscalarcomputing, VLIW (very long instruction word) architectures, and multithreading.
  - ILP requiresbranch prediction, dynamic scheduling, speculation, and compiler support to work efficiently.
- **Data-level parallelism** (DLP)
  - DLP through SIMD (single instruction, multipledata) and vector machines using vector or array types of instructions.
  - DLP requires even more hardware support and compiler assistance to work properly.
- **Task-level parallelism (TLP):**
  - Ever since the introduction of multicoreprocessors and chip multiprocessors (CMPs), we have been exploring TLP
  - TLP is far from being very successful due to difficulty in programming and compilation of

code for efficient execution onmulticore CMPs.

- **Utility Computing**
  - Utility computing focuses on a business model in which customers receive computing resources from a paid service provider. All grid/cloud platforms are regarded as utility service providers.

- **The Internet of Things (IoT)**
  - Traditional Internet connects machines to machines or web pages to web pages.
  - IoT was introduced in 1999 at MIT
  - networked interconnection of everyday objects, tools, devices, or computers
  - a wireless network of sensors that interconnect all things in our daily life.
  - Three communication patterns co-exist: namely H2H (human-to-human), H2T (human-tothing),and T2T (thing-to-thing).
  - connect things (including human and machine objects) at any time and any place intelligently with low cost
  - IPv6 protocol, 2128 IP addresses are available to distinguish all the objects on Earth, including all computers and pervasive devices
  - IoT needs to be designed to track 100 trillion static or moving objects simultaneously.
  - IoT demands universal addressability of all of the objects orthings.
  - The dynamic connections will grow exponentially into a new dynamic network of networks, called the Internet of Things (IoT).

Cyber-Physical Systems

- A cyber-physical system (CPS) is the result of interaction between computationalprocesses and the physical world.
- CPS integrates "cyber" (heterogeneous, asynchronous) with "physical" (concurrent and information-dense) objects
- CPS merges the "3C" technologies of computation, communication, and control into an intelligent closed feedback system

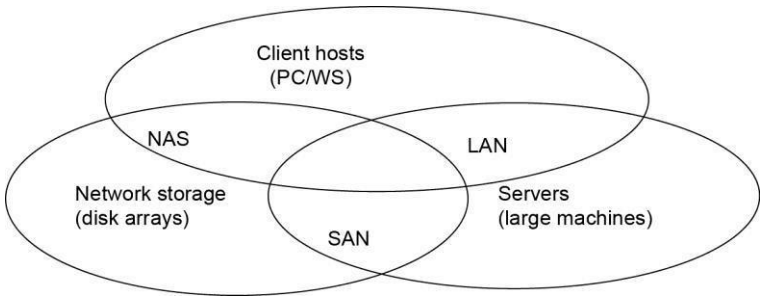
IoT emphasizes various networking connections among physical objects, while the CPS emphasizes exploration of virtual reality (VR) applications in the physical world

TECHNOLOGIES FOR NETWORK-BASED SYSTEMS

Multicore CPUs and Multithreading Technologies: 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. Both multi-core CPU and many-core GPU processors can handle multiple instruction threads at different magnitudes today. Multiple cores are housed in the same chip with an L2 cachethat is shared by all cores. In the future, multiple CMPs could be built on the same CPU chip with even the L3cache on the chip. Multicore and multithreaded CPUs are equipped with many high-end processors, including the Intel i7, Xeon, AMD Opteron, Sun Niagara, IBM Power 6, and X cell processors. Each core could be also multithreaded.

Memory, Storage, and Wide-Area Networking: Memory chips have experienced a 4x increase in capacity every three years. For hard drives, capacity increased from 260 MB in 1981 to 250 GB in 2004. Disks or disk arrays have exceeded 3 TB in capacity. The rapid growth of flash memory and solid-state drives (SSDs) also impacts the future of HPC and HTC systems.

System-Area Interconnects: The nodes in small clusters are mostly interconnected by an Ethernet switch or a local area network(LAN).

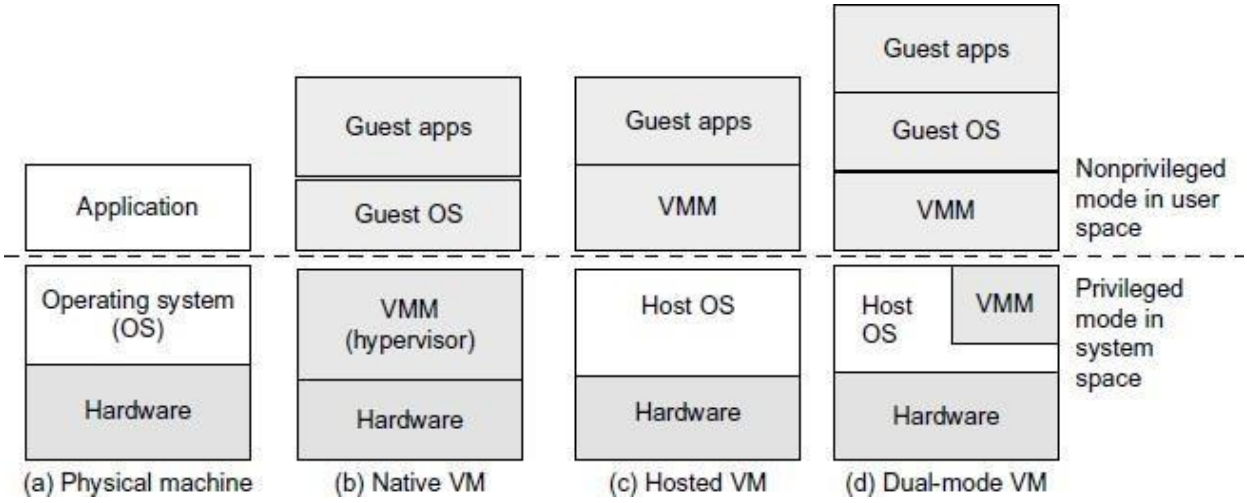


As Figure shows, a LAN typically is used to connect client hosts to big servers.A storage area network (SAN) connects servers to network storage such as disk arrays. Network attached storage (NAS) connects client hosts directly to the disk arrays. All three types of networks often appear in a large cluster built with commercial network components.

Wide-Area Networking: High-bandwidth networking increases the capability of building massively distributed systems. The rapid growth of Ethernet bandwidth from 10 Mbps in1979 to 1 Gbps in 1999, and 40 ~ 100 GE in 2011. It has been speculated that 1 Tbps network links will become available by 2013.

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. Today, 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. In particular, a cloud of provisioned resources must rely on virtualization of processors, memory, and I/O facilities dynamically.



Three VM architectures in (b), (c), and (d), compared with the traditional physical machine shown in (a).

The host machine is equipped with the physical hardware. 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).

Figure shows a native VM installed with the use of a VMM called a hypervisor in privileged Mode.
The guest OS could be a Linux system and the hypervisor is the XEN system developed at
Cambridge University. This hypervisor approach is also called bare-metal VM, because the hypervisor handles the bare hardware (CPU, memory, and I/O) directly. Architecture is the host VM shown in Figure(c). Here the VMM runs in non-privileged mode. The host OS need not be modified. The VM can also be implemented with a dual mode, as shown in Figure 1.12(d). Part of the VMM runs at the user level and another part runs at the supervisor level. In this case, the host OS may have to be modified to some extent. Multiple VMs can be ported to a given hardware system to support the virtualization process. The VM approach offers hardware independence of the OS and applications.

VM Primitive Operations: The VMM provides the VM abstraction to the guest OS. With full virtualization, the VMM exports a VM abstraction identical to the physical machine so that a standard OS such as Windows 2000 or Linux can run just as it would on the physical hardware

Low-level VMM operations are

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

These VM operations enable a VM to be provisioned to any available hardware platform. They also enable flexibility in porting distributed application executions. Furthermore, the VM approach will significantly enhance the utilization of server resources.

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

Table 1.2 Classification of Parallel and Distributed Computing Systems

Functionality, Applications	Computer Clusters [10,28,38]	Peer-to-Peer Networks [34,46]	Data/ Computational Grids [6,18,51]	Cloud Platforms [1,9,11,12,30]
Architecture, Network Connectivity, and Size	Network of compute nodes interconnected by SAN, LAN, or WAN hierarchically	Flexible network of client machines logically connected by an overlay network	Heterogeneous clusters interconnected by high-speed network links over selected resource sites	Virtualized cluster of servers over data centers via SLA
Control and Resources Management	Homogeneous nodes with distributed control, running UNIX or Linux	Autonomous client nodes, free in and out, with self-organization	Centralized control, server-oriented with authenticated security	Dynamic resource provisioning of servers, storage, and networks
Applications and Network-centric Services	High-performance computing, search engines, and web services, etc.	Most appealing to business file sharing, content delivery, and social networking	Distributed supercomputing, global problem solving, and data center services	Upgraded web search, utility computing, and outsourced computing services
Representative Operational Systems	Google search engine, SunBlade, IBM Road Runner, Cray XT4, etc.	Gnutella, eMule, BitTorrent, Napster, KaZaA, Skype, JXTA	TeraGrid, GriPhyN, UK EGEE, D-Grid, ChinaGrid, etc.	Google App Engine, IBM Bluecloud, AWS, and Microsoft Azure

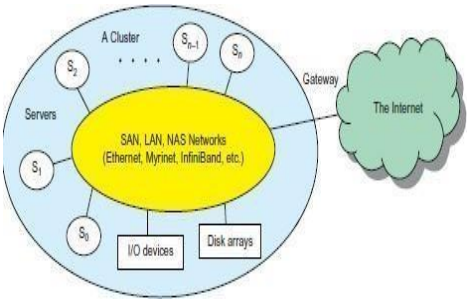
- A massive system is with millions of computers connected to edge networks.
- Massive systems are considered highly scalable
- massive systems are classified into four groups: clusters, P2P networks, computing grids, and Internet clouds

Computing cluster

- A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.

Cluster Architecture

- the architecture consists of a typical server cluster built around a low-latency, high bandwidth interconnection network.
- 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
- cluster is connected to the Internet via a virtual private network (VPN) gateway.
- gateway IP address locates the cluster



- Clusters have loosely coupled node computers.
- All resources of a server node are managed by their own OS.
- Most clusters have multiple system images as a result of having many autonomous nodes under different OS control

Single-System Image -Cluster

- an ideal cluster should merge multiple system images intoa single-system image (SSI)
- cluster operating system or some middleware have to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.
- 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

Hardware, Software, and Middleware Support –Cluster

- Clusters exploring massive parallelism are commonly known as MPPs –Massive Parallel Processing

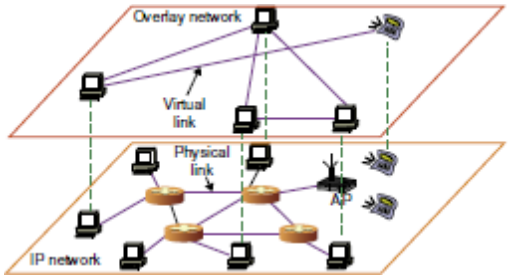
- The building blocks are computer nodes (PCs, workstations, servers, or SMP), special communication software such as PVM or MPI, and a network interface card in each computer node.
- Most clusters run under the Linux OS.
- nodes are interconnected by a high-bandwidth network
- Special cluster middleware supports are needed to create SSI or high availability (HA).
- all distributed memory to be shared by all servers by forming distributed shared memory (DSM).
- SSI features are expensive
- achieving SSI, many clusters are loosely coupled machines
- virtual clusters are created dynamically, upon user demand

Grid Computing

- A web service such as HTTP enables remote access of remote web pages
- computing grid offers an infrastructure that couples computers, software/middleware, special instruments, and people and sensors together
- Enterprises or organizations present grids as integrated computing resources. They can also beviewed as virtual platforms to support virtual organizations.
- The computers used in a grid are primarilyworkstations, servers, clusters, and supercomputers

Peer-to-Peer Network-P2P

- P2P architecture offers a distributed model of networked systems.
- P2P network is client-oriented instead of server-oriented
- In a P2P system, every node acts as both a client and a server
- 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 system is self-organizing with distributed control.
- P2P two layer of abstractions as given in the figure

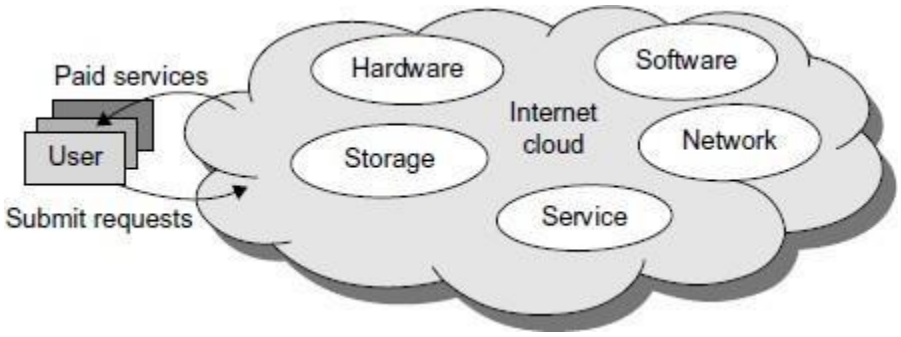


- Each peer machine joins or leaves the P2P network voluntarily
- Only the participatingpeers form the physical network at any time.
- Physical network is simply an ad hoc networkformed at various Internet domains randomly using the TCP/IP and NAI protocols.

Peer-to-Peer Network-Overlay network

- 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.
- 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.
- 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

Cloud Computing



Virtualized resources from data centers to form an Internet cloud

- 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."
- Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically

The Cloud Landscape

Infrastructure as a Service (IaaS)

- This model puts together infrastructures demanded by users—namely servers, storage, networks, and the data center fabric.
- The user can deploy and run on multiple VMs running guest OSes on specific applications.
- The user does not manage or control the underlying cloud infrastructure, but can specify when to request and release the needed resources.

Platform as a Service (PaaS)

- This model enables the user to deploy user-built applications onto a virtualized cloud platform.
- PaaS includes middleware, databases, development tools, andsome runtime support such as Web 2.0 and Java.
- The platform includes both hardware andsoftware integrated with specific programming interfaces.
- The provider supplies the API andsoftware tools (e.g., Java, Python, Web 2.0, .NET). The user is freed from managing the cloudinfrastructure.

Software as a Service (SaaS)

- This refers to browser-initiated application software overthousands of paid cloud customers.
- The SaaS model applies to business processes, industryapplications, consumer relationship management (CRM), enterprise resources planning (ERP),human resources (HR), and collaborative applications.
- On the customer side, there is no upfrontinvestment in servers or software licensing.
- On the provider side, costs are rather low, comparedwith conventional hosting of user applications

Internet clouds offer four deployment modes: private, public, managed, and hybrid

SOFTWARE ENVIRONMENTS FOR DISTRIBUTED SYSTEMSAND CLOUDS

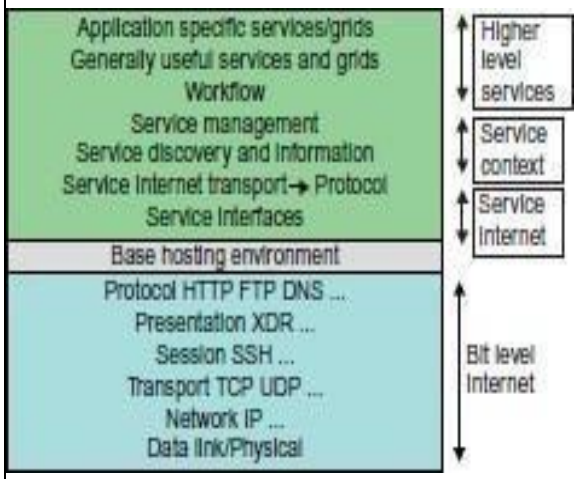
Service-Oriented Architecture (SOA)

- In grids/web services, Java, and CORBA, an entity is, respectively, a service, a Java object, and a CORBA distributed object in a variety of languages.
- These architectures build on the traditional seven Open Systems Interconnection (OSI) layers that provide the base networking abstractions.
- On top of this we have a base software environment, which would be
    - .NET or Apache Axis for web services,
    - the Java Virtual Machine for Java, and a broker network for CORBA
- On top of this base environment one would build a higher level environment reflecting the special features of the distributed computing environment.
- SOAapplies to building grids, clouds, grids of clouds, clouds of grids, clouds of clouds (also known asinterclouds),
- SS (sensor service : A large number of sensors provide data-collectionservices (ZigBee device, a Bluetoothdevice, WiFi access point, a personal computer, a GPA, or a wireless phoneetc
- Filter services : to eliminate unwanted raw data, in orderto respond to specific requests from the web, the grid, or web services

Layered Architecture for Web Services and Grids

- **Entity Interfaces**
- Java method interfaces correspond to the Web Services Description Language (WSDL),
- CORBA interface - definition language (IDL) specifications

- These interfaces are linked with customized, high-level communication systems: SOAP, RMI, and IIOP
- These communication systems support features including particular message patterns (such as Remote Procedure Call or RPC), fault recovery, and specialized routing.
- Communication systems are built on message-oriented middleware (enterprise bus) infrastructure such as Web-Sphere MQ or Java Message Service (JMS)



Cases of fault tolerance- the features in the Web Services Reliable Messaging (WSRM)
Security -reimplements the capabilities seen in concepts such as Internet Protocol Security (IPsec)

Several models with, for example, JNDI (Jini and Java Naming and DirectoryInterface) illustrating different approaches within the Java distributed object model. The CORBA TradingService, UDDI (Universal Description, Discovery, and Integration), LDAP (Lightweight Directory Access Protocol), and ebXML (Electronic Business using eXtensibleMarkup Language

earlier years, CORBA and Java approaches were used in distributed systems rather than today'sSOAP, XML, or REST (Representational State Transfer).

Web Services and Tools

REST approach:

- delegates most ofthe difficult problems to application (implementation-specific) software. In a web services language
- minimal information in the header, and the message body (that is opaque to genericmessage processing) carries all the needed information.
- architectures are clearly more appropriatefor rapid technology environments.
- REST can use XML schemas but not those that are part of SOAP; "XML overHTTP" is a popular design choice in this regard.
- Above the communication and managementlayers, we have the ability to compose new entities or distributed programs by integrating severalentities together.

CORBA and Java:

- the distributed entities are linked with RPCs, and the simplest way to buildcomposite

applications is to view the entities as objects and use the traditional ways of linking themtogether.

- For Java, this could be as simple as writing a Java program with method calls replaced byRemote Method Invocation (RMI),
- CORBA supports a similar model with a syntax reflecting the C++ style of its entity (object) interfaces.

Parallel and Distributed Programming Models

Table 1.7 Parallel and Distributed Programming Models and Tool Sets

Model	Description	Features
MPI	A library of subprograms that can be called from C or FORTRAN to write parallel programs running on distributed computer systems [6,28,42]	Specify synchronous or asynchronous point-to-point and collective communication commands and I/O operations in user programs for message-passing execution
MapReduce	A web programming model for scalable data processing on large clusters over large data sets, or in web search operations [16]	*Map* function generates a set of intermediate key/value pairs; *Reduce* function merges all intermediate values with the same key
Hadoop	A software library to write and run large user applications on vast data sets in business applications (http://hadoop.apache.org/core)	A scalable, economical, efficient, and reliable tool for providing users with easy access of commercial clusters

PERFORMANCE, SECURITY, AND ENERGY EFFICIENCY

Performance Metrics:

- In a distributed system, performance is attributed to a large number of factors.
- System throughput is often measured in MIPS, Flops (tera floating-point operationsper second), or TPS (transactions per second).
- System overhead is often attributed to OS boot time, compile time, I/O data rate, and the runtime support system used.
- Other performance-related metrics include the QoS for Internet and web services; system availability and dependability; and security resilience for system defense against network attacks

Dimensions of Scalability

Any resource upgrade Ina system should be backward compatible with existing hardware and

software resources. System scaling can increase or decrease resources depending on many practical factors

Size scalability

- This refers to achieving higher performance or more functionality by increasing the machine size.
- The word "size" refers to adding processors, cache, memory, storage, or I/O Channels. The most obvious way to determine size scalability is to simply count the number of processors installed.
- Not all parallel computer or distributed architectures are equally size scalable.
- For example, the IBM S2 was scaled up to 512 processors in 1997. But in 2008, theIBMBlueGene/L system scaled up to 65,000 processors.

• **Software scalability**

- This refers to upgrades in the OS or compilers, adding mathematical and engineering libraries, porting new application software, and installing more user- friendly programming environments.
- Some software upgrades may not work with large system configurations.
- Testing and fine-tuning of new software on larger systems is a nontrivial job.

• **Application scalability**

- This refers to matching problem size scalability with machine size scalability.
- Problem size affects the size of the data set or the workload increase. Instead of increasing machine size, users can enlarge the problem size to enhance system efficiency or cost-effectiveness.

• **Technology scalability**

- This refers to a system that can adapt to changes in building technologies, such as the component and networking technologies
- When scaling a system design with new technology one must consider three aspects: time, space, and heterogeneity.
- (1) Time refers to generation scalability. When changing to new-generation processors, one must consider the impact to the motherboard, power supply, packaging and cooling, and so forth. Based on past experience, most systems upgrade their commodity processors every three to five years.

- (2) Space is related to packaging and energy concerns. Technology scalability demands harmony and portability among suppliers.

- (3) Heterogeneity refers to the use of hardware components or software packages fromdifferent vendors. Heterogeneity may limit thescalability.

Amdahl's Law

- Let the program has been parallelized or partitioned for parallel execution on a cluster of many processing nodes.
- Assume that a fraction α of the code must be executed sequentially, called the sequential bottleneck.
- Therefore, (1 − α) of the code can be compiled for parallel execution by n processors.

  The total execution time of the program is calculated byα T + (1 − α)T/n, where the first term is the sequential execution time on a single processor and the second term is the parallel execution time on n processing nodes.
- I/O time or exception handling time is also not included in the following speedup analysis.

$$\text{Speedup} = S = T/[\alpha T + (1-\alpha)T/n] = 1/[\alpha + (1-\alpha)/n]$$

- Amdahl's Law states that the speedup factor of using the n-processor system over the use of a single processor is expressed by:

- the code is fully parallelizable with α = 0. As the cluster becomes sufficiently large, that is, $n \to \infty$, S approaches 1/α, an upper bound on the speedup S.

- this upper bound is independent of the cluster size n. The sequential bottleneck is the portion of the code that cannot be parallelized.

Gustafson's Law

- To achieve higher efficiency when using a large cluster, we must consider scaling the problem sizeto match the cluster capability. This leads to the following speedup proposed by John Gustafson(1988), referred as scaled-workload speedup.

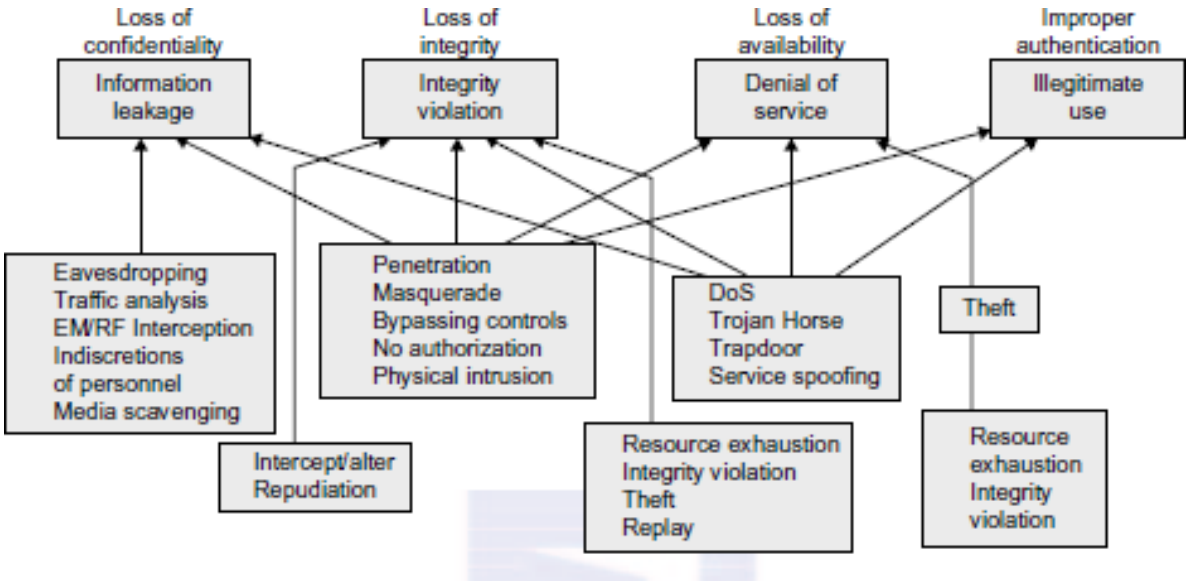
- Let W be the workload in a given program.

- When using an n-processor system, the user scales the workload to

W′ = αW + (1 − α)nW.Scaled workload W′ is essentially the sequential execution time in a single processor. The parallelexecution time of a scaled workload W′ on n processors is defined by a scaled-workload speedupas follows:

$$S' = W'/W = [\alpha W + (1-\alpha)nW]/W = \alpha + (1-\alpha)n$$

Network Threats and Data Integrity



ENERGY EFFICIENCY IN DISTRIBUTED COMPUTING

Primary performance goals in conventional parallel and distributed computing systems are highperformance and high throughput, considering some form of performance reliability (e.g., fault toleranceand security). However, these systems recently encountered new challenging issues includingenergy efficiency, and workload and resource outsourcing

Energy Consumption of Unused Servers: To run a server farm (data center) a company has to spend a huge amount of money for hardware,software, operational support, and energy every year. Therefore, companies should thoroughlyidentify whether their installed server farm (more specifically, the volume of provisioned resources)is at an appropriate level, particularly in terms of utilization.

Reducing Energy in Active Servers: In addition to identifying unused/underutilized servers for energy savings, it is also necessary toapply appropriate techniques to decrease energy consumption in active distributed systems with negligibleinfluence on their performance.

Application Layer: Until now, most user applications in science, business, engineering, and financial areas tend toincrease a system's speed or quality. By introducing energy-aware applications, the challenge is todesign sophisticated multilevel and multi-domain energy management applications without hurtingperformance.

Middleware Layer: The middleware layer acts as a bridge between the application layer and the resource layer. Thislayer provides resource broker, communication service, task analyzer, task scheduler, securityaccess, reliability control, and information service capabilities. It is also responsible for applyingenergy-efficient techniques, particularly in task scheduling.

Resource Layer: The resource layer consists of a wide range of resources including computing nodes and storageunits. This layer generally interacts with hardware devices and the operating system; therefore, itis responsible for controlling all distributed resources in distributed computing systems. Dynamic power management (DPM) and dynamic voltage-frequency scaling (DVFS) are two popular methods incorporated into recent computer hardware systems.In DPM, hardware devices, such as the CPU, have the capability to switch from idle mode to oneor more lower power modes. In DVFS, energy savings are achieved based on the fact that the power consumptionin CMOS circuits has a direct relationship with frequency and the square of the voltage supply.

Network Layer: Routing and transferring packets and enabling network services to the resource layer are the mainresponsibility of the network layer in distributed computing systems. The major challenge to buildenergy-efficient networks is, again, determining how to measure, predict, and create a balancebetween energy consumption and performance.

UNIT -2

Levels of Virtualization Implementation

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) aremultiplexed in the same hardware machine.
- After virtualization, different user applications managed by their own operating systems (guest OS) can run onthe same hardware independent of the host OS
- done by adding additional software, called a virtualization layer
- This virtualization layer is known as hypervisor or virtual machine monitor (VMM)

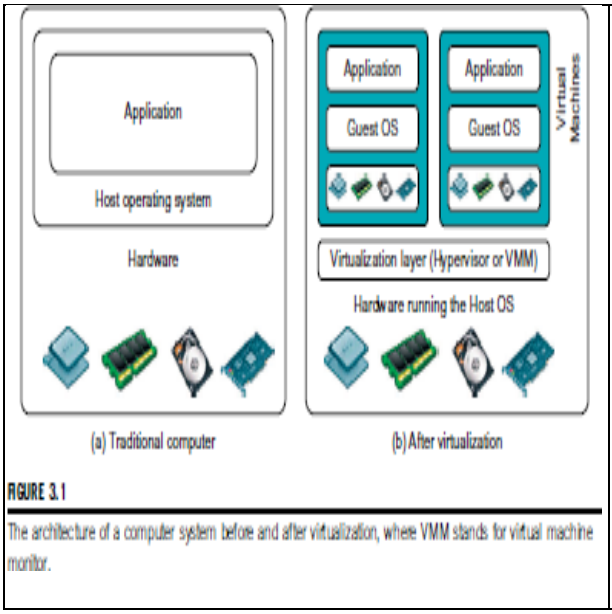
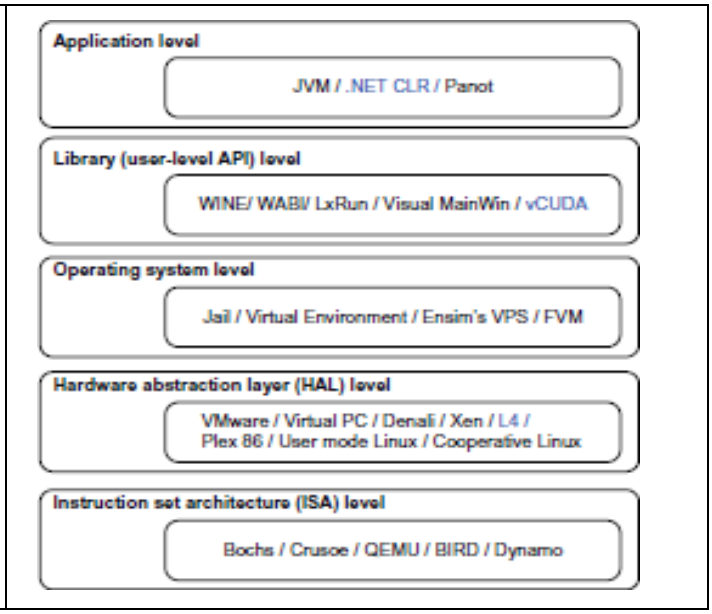


FIGURE 3.1
The architecture of a computer system before and after virtualization, where VMM stands for virtual machine monitor.



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

Instruction Set Architecture Level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation. With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.

- Instruction set emulation leads to virtual ISAs created on any hardware machine. The basic emulation method is through code interpretation. An interpreter program interprets 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.
- Instruction set emulation requires binary translation and optimization. A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

Hardware Abstraction Level

- Hardware-level virtualization is performed right on top of the bare hardware.
- Thisapproach generates a virtual hardware environment for a VM.
- The process managesthe underlying hardware through virtualization. The idea is to virtualize a computer's resources, such asits processors, memory, and I/O devices.
- The intention is to upgrade the hardware utilization rate bymultiple users concurrently. The idea was implemented in the IBM VM/370 in the 1960s.
- Morerecently, the Xen hypervisor has been applied to virtualize x86-based machines to run Linux or otherguest OS applications.

Operating System Level

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

Library Support Level

- Most applications use APIs exported by user-level libraries rather than using lengthy system callsby the OS.
- Since most systems provide well-documented APIs, such an interface becomes anothercandidate for virtualization.
- Virtualization with library interfaces is possible by controlling the communicationlink between applications and the rest of a system through API hooks.

User-Application Level

- Virtualization at the application level virtualizes an application as a VM.
- On a traditional OS, anapplication 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 sits as an application program on top of the operatingsystem,
- The layer exports an abstraction of a VM that can run programs written and compiledto a particular abstract machine definition.
- Any program written in the HLL and compiled for thisVM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are twogood examples of this class of VM.

VMM Design Requirements and Providers

- layer between real hardware and traditional operating systems. This layer is commonly called the Virtual Machine Monitor (VMM)
- three requirements for a VMM
- a VMM should provide an environment for programs which is essentially identical to the original machine
- programs run in this environment should show, at worst, only minor decreases in speed
- VMM should be in complete control of the system resources.
- VMM includes the following aspects:
- (1) The VMM is responsible for allocating hardware resources for programs;
- (2) it is not possible for a program to access any resource not explicitly allocated to it;
- (3) it is possible under certain circumstances for a VMM to regain control of resources already allocated.

Virtualization Support at the OS Level

- Why OS-Level Virtualization? :
    - it is slow to initialize a hardware-level VM because each VM creates its own image from scratch.
- 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), Virtual Private System (VPS), or simply container
- The benefits of OS extensions are twofold:
    - (1) VMs at the operating system level have minimal startup/shutdown costs, low resource requirements, and high scalability;
    - (2) for an OS-level VM, it is possible for a VM and its host environment to synchronize state changes when necessary**.**

Middleware Support for Virtualization

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

Hypervisor and Xen Architecture

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

Xen Architecture

- Xen is an open source hypervisor program developed by Cambridge University.
- Xen is a microkernel hypervisor
- The core components of a Xen system are the hypervisor, kernel, and applications
- The **guest OS**, which has control ability, is called **Domain 0**, and the others are called **Domain U**
- Domain 0 is designed to access hardware directly and manage devices

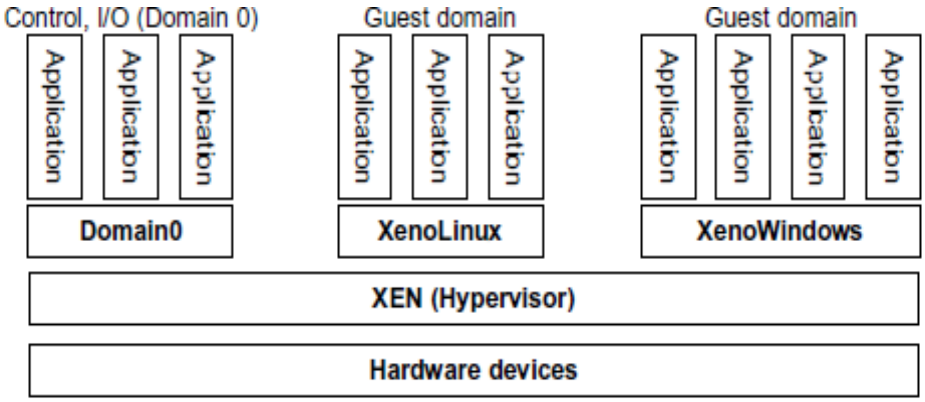


FIGURE 3.5

The Xen architecture's special domain 0 for control and I/O, and several guest domains for user applications.
(Courtesy of P. Barham, et al. [7])

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

Full virtualization

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

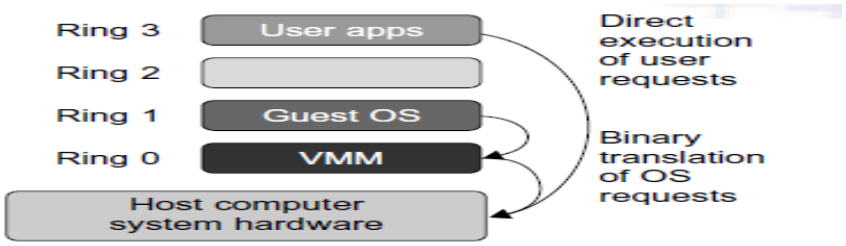


FIGURE 3.6
Indirect execution of complex instructions via binary translation of guest OS requests using the VMM plus direct execution of simple instructions on the same host.
(Courtesy of VM Ware [71])

Para-Virtualization

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

CPU Virtualization

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

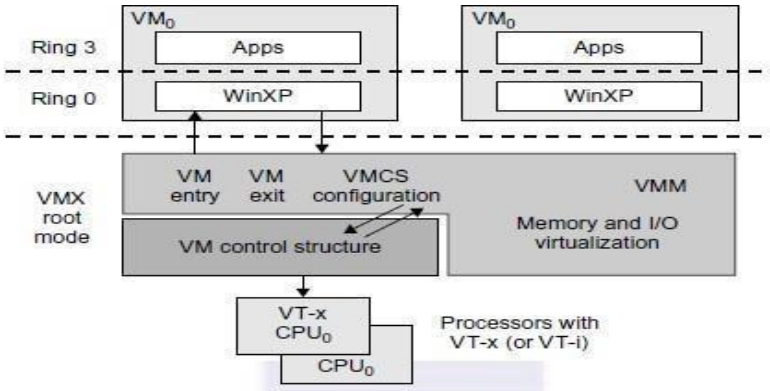


FIGURE 3.11
Intel hardware-assisted CPU virtualization.
(Modified from [68], Courtesy of Lizhong Chen, USC)

Memory Virtualization

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

respectively: virtual memory to physical memory and physical memory to machine memory.

- The VMM is responsible for mapping the guest physical memory to the actual machine memory.

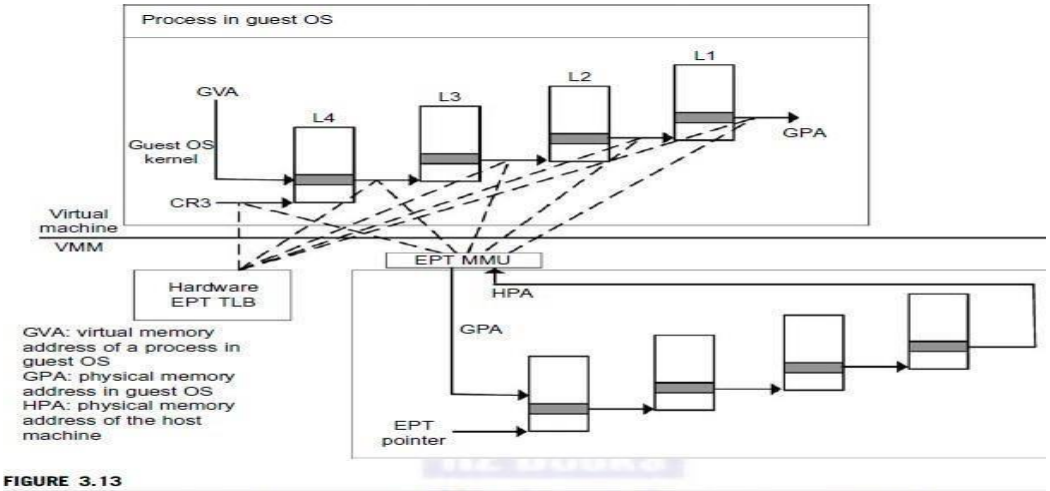


FIGURE 3.13
Memory virtualization using EPT by Intel (the EPT is also known as the shadow page table [68]).

I/O Virtualization

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

Virtualization in Multi-Core Processors

- **Muti-core virtualization** has raised some new **challenges**
- **Two difficulties**: Application programs must be parallelized to use all cores fully, and software must explicitly
- Assign tasks to the cores, which is a very complex problem

- 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
- **Dynamic heterogeneity** is emerging to mix the fat CPU core and thin GPU cores on the same chip

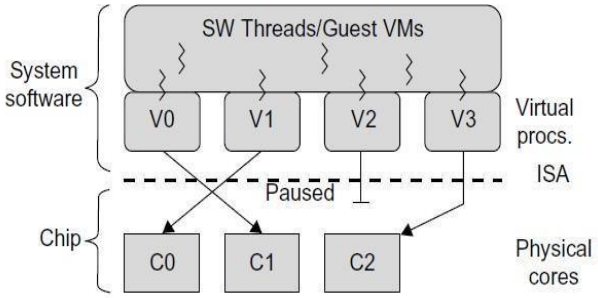


FIGURE 3.16

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

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

- In **many-core chip multiprocessors (CMPs)**→
- Instead of supporting **time-sharing jobs** on one or a few cores, use the abundant cores →**space-sharing**, where single-threaded or multithreaded jobs are simultaneously assigned to separate groups of cores

Physical versus Virtual Clusters

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

Virtual Clusters

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

- Third way to manage a virtual cluster is to use an **independent cluster manager** on both the host and guest systems.
- Finally, use an **integrated cluster** on the guest and host systems.
- This means the manager must be designed to distinguish between virtualized resources and physical resources

Virtualization for data-center automation

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

Server Consolidation in Data Centers

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

Virtual Storage Management

- **storage virtualization** has a different meaning in a system virtualization environment
- **system virtualization,** virtual storage includes the storage managed by VMMs and guest OSes data stored in this environment can be classified into two categories: VM images and application data.

Cloud OS for Virtualized Data Centers

- Data centers must be virtualized to serve as cloud providers
  - **Eucalyptus for Virtual Networking of Private Cloud** : Eucalyptus is an open source software system intended mainly for supporting Infrastructure as a Service (IaaS) clouds
- The system primarily supports **virtual networking** and the management of **VMs**;
- virtual storage is not supported.
- Its purpose is to build **private clouds**
  - three resource managers: Instance Manager, Group Manager, Cloud Manager