PPT ON

CLOUD COMPUTING

VII SEM (IARE-R16)

By
A. Praveen
Assistant Professor
Department of IT
IARE

UNIT 1
SYSTEMS MODELLING, CLUSTERING AND VIRTUALIZATION

Scalable Computing Over the Internet

- **Scalability: Scalability is the capability of a system or network or process to handle a growing amount of works like database storage, software usage and so on [1]. A scalable system should be able to handle the ever-increasing data, levels of computations and should be efficient.**
- **NOTE: Generally, a computer uses a centralized system to solve the problems. A parallel and distributed computing system uses multiple computers to solve large scale problems over the Internet [2].**
- **Parallel Computing: Execution of many processes is carried out simultaneously in this case. Large problems can be divided into smaller ones, solved at the same time and integrated later.**

Scalable Computing Over the Internet

- **Distributed Computing: Distributed computing may refer to systems situated at different physical locations or different actions being performed on the same system. Interchange/monitor their actions by passing messages.**
- **Distributed Computing is centred on data and based on networks.**
- **NOTE: *Data Center* is a centralised repository and distribution of data and information organised around a particular concept (ex: Telecommunications, Health data, business data etc.). A typical data center may have a capacity in Petabytes.**

Scalable Computing Over the Internet

- **Internet Computing: Data centers and super computer sites must be upgraded to meet the demands of millions of users who utilize the Internet.**
- **High Performance Computing (HPC), which was a standard for measuring the system performance, is no longer used.**
- **High Throughput Computing (HTC) came into existence with emergence of computing clouds. Here, the systems are parallel and distributed.**

Scalable Computing Over the Internet

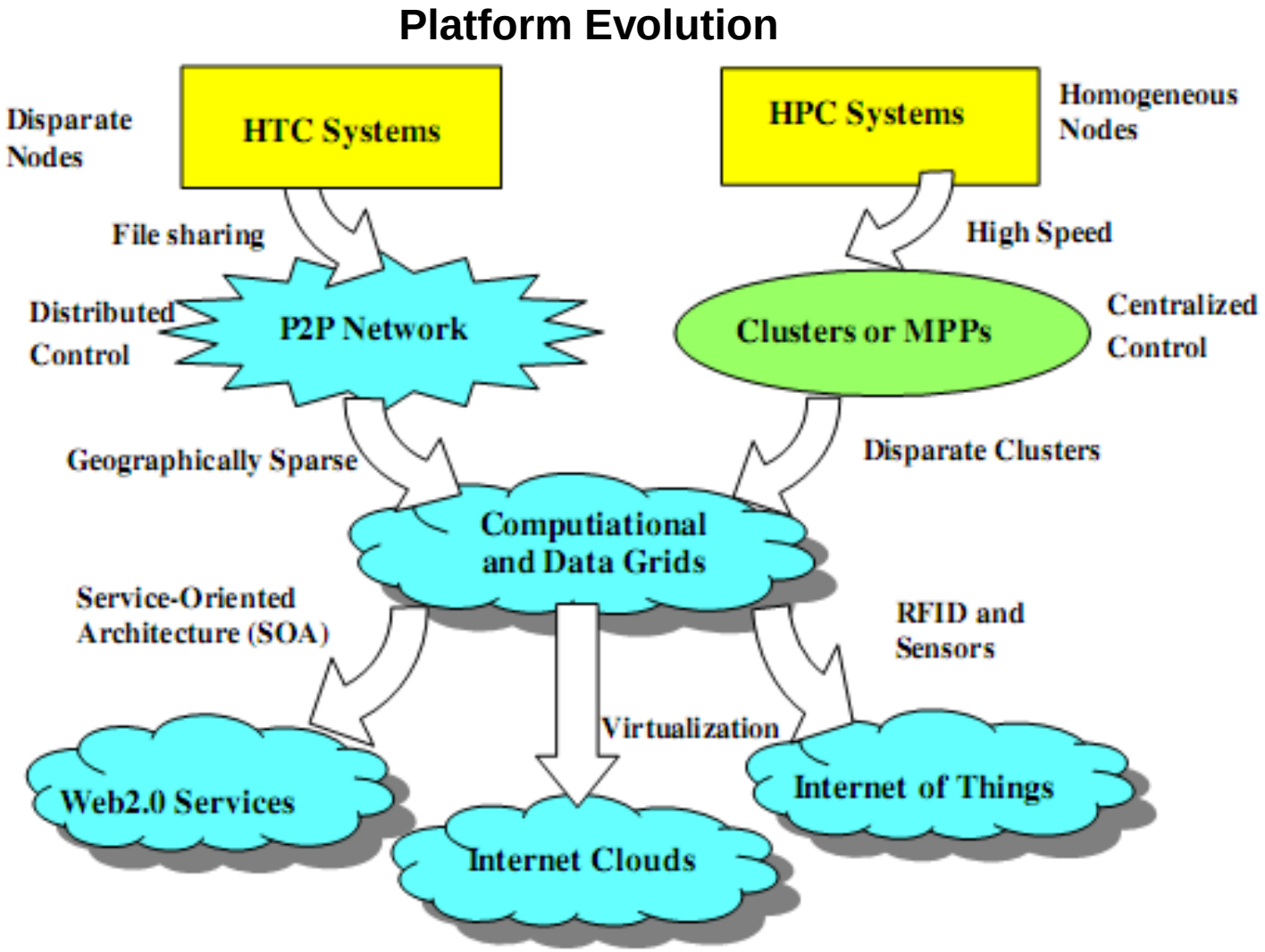


Figure 1.1 [2]: Evolutionary Trend towards parallel, distributed and cloud computing

Scalable Computing Over the Internet

- Computer technology has gone through five generations of development, each spanning at 10 to 20 years. By the start of 1990s, the use of HPC and HTC systems has sky-rocketed. These use clusters, grids, Internet and clouds.
- The general trend is to control shared web resources and massive data over the Internet. In the above figure 1.1, we can observe the evolution of HPC and HTC systems.
- NOTE: HPC contains super computers which are gradually replaced by clusters of inter-cooperating systems that share the data among them. A cluster is a collection of homogeneous computers, which are physically connected.

Scalable Computing Over the Internet

- **HTC shows the formation of peer-to-peer (P2P) networks for distributed file sharing and apps. A P2P system is built over many client machines and is globally distributed. This leads to formation of computational grids or data grids.**

Scalable Computing Over the Internet

- **High Performance Computing (HPC): HPC stressed upon the speed performance. The speed of HPC systems has increased from Gflops to Pflops (FLOP=> Floating Point Operations Per Second) these days, driven by the requirements from different fields like science, engineering, medicine and others [3]. The systems that generally have high speed are super computers, main frames and other servers.**

- **It should be noted here that the number of users (in HPC) is limited – less than 10% of all the users. The majority of the market now uses servers, PCs or mobile devices that conduct Internet searches and other assigned tasks.**

Scalable Computing Over the Internet

- **High Throughput Computing: The market-oriented computing is now going through a strategic change from HPC to HTC paradigm (concept). HTC concentrates more on high-flux computing (ex: Internet searches, web apps used by many users simultaneously). The performance goal has shifted from speed of the device to the number of tasks completed per unit of time (throughput).**
- **HTC needs not only to improve the speed but also to solve other problems like time availability, cost, security and reliability.**

Scalable Computing Over the Internet

- **New Computing Concepts: It can be seen from Figure 1.1 that SOA (Software Oriented Architecture) has made the web services available for all tasks. The Internet Clouds have become a major factor to consider for all types of tasks. Three new paradigms have come into existence:**
- **Radio-Frequency Identification (RFID): This uses electro-magnetic fields to automatically identify and track tags attached to objects [4]. These tags contain electronically stored information.**
- **Global Positioning System (GPS): It is a global navigation satellite system that provides the geographical location and time information to a GPS receiver [5].**

Scalable Computing Over the Internet

- **Internet of Things (IoT): It is the internetworking of different physical devices (vehicles, buildings etc.) embedded with electronic devices (sensors), software, and network connectivity [6]. Data can be collected and exchanged through this network (IoT).**

Computing Paradigm Distinctions

- **Centralized Computing: All computer resources like processors, memory and storage are centralized in one physical system. All of these are shared and inter-connected and monitored by the OS.**
- **Parallel Computing: All processors are tightly coupled with centralized shared memory or loosely coupled with distributed memory (parallel processing). Inter processor communication is achieved by message passing. This methodology is known as parallel computing.**
- **NOTE: Coupling is the inter-dependence between software/hardware modules.**

Computing Paradigm Distinctions

- **Distributed Computing: A distributed system consists of multiple autonomous computers with each device having its own private memory. They interconnect among themselves by the usage of a computer network. Here also, information exchange is accomplished by message passing.**
- **Cloud Computing: An Internet Cloud of resources can either be a centralized or a distributed computing system. The cloud applies parallel or distributed computing or both. Cloud can be built by using physical or virtual resources over data centers. CC is also called as utility/ service/concurrent computing.**

Computing Paradigm Distinctions

- **NOTE: IoT is a networked connection of general objects used everyday including computers, systems and sensors. IoT is supported by Internet Clouds to access any 'thing' at any place at any time. Internet Computer is a larger concept that covers all computing paradigms, emphasizing on distributed and cloud computing.**
- ***Explanation on the recent surge in networks of clusters, data grids.*** **Internet Clouds are the result of moving desktop computing to service-oriented computing using server clusters and huge databases at data centers.**
- **In the future, both HPC and HTC will demand multicore processors that can handle large number of computing threads per core. Both concentrate upon parallel and distributed computing. The main work lies in the fields of throughput, efficiency, scalability and reliability.**

Computing Paradigm Distinctions

Main Objectives:

- **Efficiency: Efficiency is decided by speed, programming and throughput demands' achievement.**
- **Dependability: This measures the reliability from the chip to the system at different levels. Main purpose here is to provide good QoS (Quality of Service).**
- **Adaption in the Programming Model: This measures the ability to support unending number of job requests over massive data sets and virtualized cloud resources under different models.**
- **Flexibility: It is the ability of distributed systems to run in good health in both HPC (science/engineering) and HTC (business).**

Degrees of ‘Parallelism’

- Bit-level parallelism (BLP) 8 bit, 16, 32, and 64.
- Instruction-level parallelism (ILP): The processor executes multiple instructions simultaneously. Ex: Pipelining, supercomputing, VLIW (very long instruction word), and multithreading.
- Pipelining: Data processing elements are connected in series where output of one element is input to the next.
- Multithreading: Multithreading is the ability of a CPU or a single core in a multi-core processor to execute multiple processes or threads concurrently, supported by the OS.

Degrees of 'Parallelism'

- **Data-level Parallelism (DLP): Here, instructions are given like arrays (single instruction, multiple data SIMD). More hardware support is needed.**
- **Task-level Parallelism (TLP): It is a process of execution where different threads (functions) are distributed across multiple processors in parallel computing environments.**
- **Job-level Parallelism (JLP): Job level parallelism is the highest level of parallelism where we concentrate on a lab or computer center to execute as many jobs as possible in any given time period [7]. To achieve this, we purchase more systems so that more jobs are running at any one time, even though any one user's job will not run faster.**

Scalable Computing Over the Internet

- **Usage of CC: It is used in different fields for different purposes. All applications demand computing economics, web-scale data collection, system reliability, and scalable performance.**
  - **Ex: Distributed transaction processing is practiced in the banking industry. Transactions represent 90 percent of the existing market for reliable banking systems. [Example of demonetization to increase Internet transactions.]**

Usage of CC

Table 1.1 Applications of High-Performance and High-Throughput Systems

Domain	Specific Applications
Science and engineering	Scientific simulations, genomic analysis, etc.
	Earthquake prediction, global warming, weather forecasting, etc.
Business, education, services industry, and health care	Telecommunication, content delivery, e-commerce, etc.
	Banking, stock exchanges, transaction processing, etc.
	Air traffic control, electric power grids, distance education, etc.
	Health care, hospital automation, telemedicine, etc.
Internet and web services, and government applications	Internet search, data centers, decision-making systems, etc.
	Traffic monitoring, worm containment, cyber security, etc.
	Digital government, online tax return processing, social networking, etc.
Mission-critical applications	Military command and control, intelligent systems, crisis management, etc.

Table 1.1 [2]

Scalable Computing Over the Internet

- Major computing paradigms and available services/capabilities are coming together to produce a technology convergence of cloud/utility computing where both HPC and HTC are utilised to achieve objectives like reliability and scalability. They also aim to reach autonomic operations that can be self-organized and support dynamic recovery. Ex: Interpretation of sensor data, effectors like Google Home and Amazon Echo, smart home devices etc.
- CC focuses on a business model where a customer receives different computing resources (storage, service, security etc.) from service providers like AWS, EMC, Salesforce.com.

Scalable Computing Over the Internet

- **A new hype (exciting) cycle is coming into picture where different important and significant works needed by the customer are offered as services by CC. Ex: SaaS, IaaS, Security as a Service, DM as a Service etc. Many others are also along the pipeline.**

Figures 1.2 and 1.3 [2] depict various actions discusse above (as in 2010).

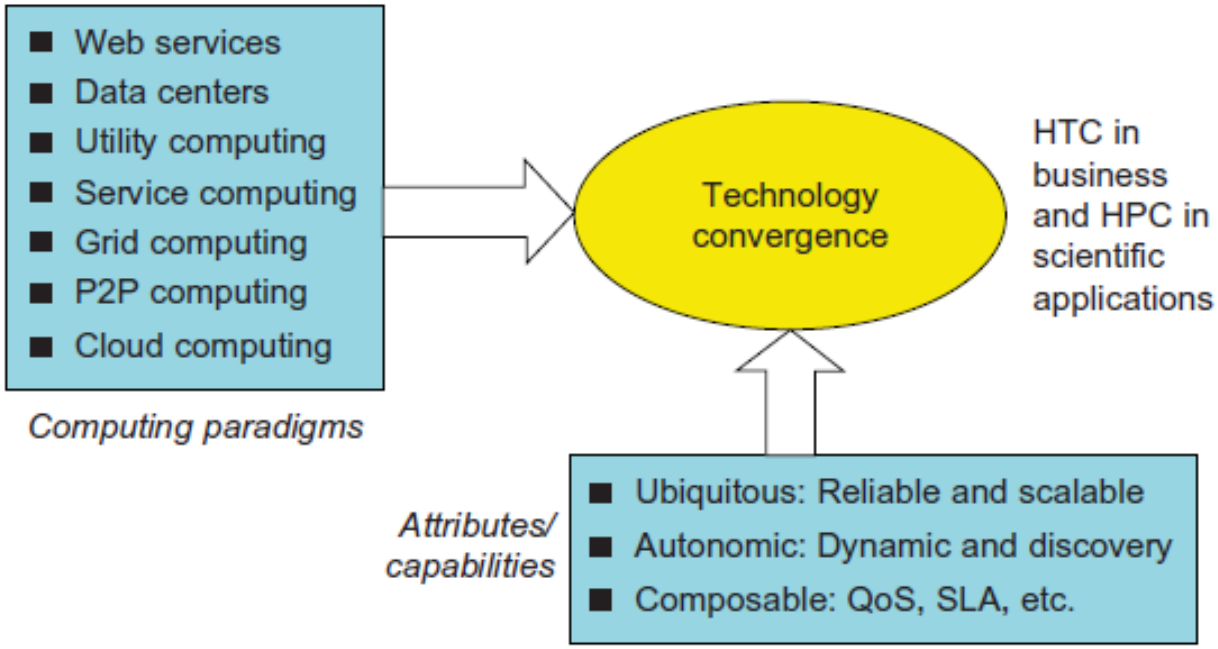


FIGURE 1.2

The vision of computer utilities in modern distributed computing systems.

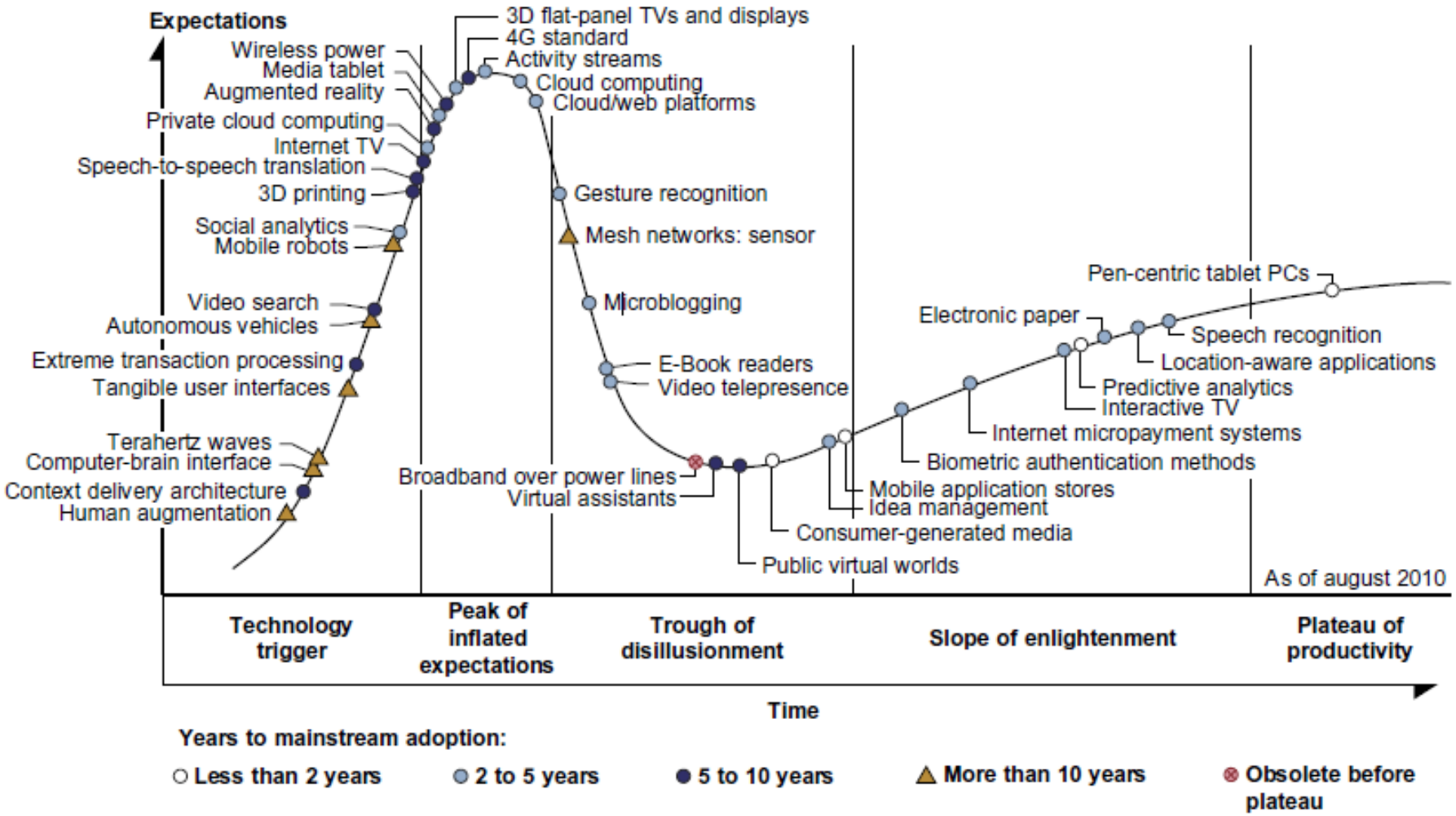


FIGURE 1.3

Hype cycle for Emerging Technologies, 2010

Internet of Things

- **Internet of Things: The IoT [8] refers the networked interconnection of everyday objects, tools, devices or computers. It can be seen as a wireless network of sensors that interconnect all things we use in our daily life. RFID and GPS are also used here. The IoT demands universal addressability of all the objects or things that may be steady or moving.**
- **These objects can be interconnected, can exchange data and interact with each other by the usage of suitable applications (web/mobile). In the IoT era, CC can be used efficiently and in a secure way to provide different services to the humans, computers and other objects. Ex: Smart cities, inter-connected networks, self-controlling street lights/traffic lights etc.**

- **NOTE: CPS means cyber–physical system where physical objects and computational processes interact with each other. Ex: Wrest bands to monitor BP. CPS merges the 3Cs which are computation, communication and control to provide intelligent feedbacks between the cyber and physical worlds.**

Technologies for Network based Systems

- **Multi-core CPUs and Multithreading Technologies: Over the last 30 years the speed of the chips and their capacity to handle variety of jobs has increased at an exceptional rate. This is crucial to both HPC and HTC system development. Note that the processor speed is measured in MIPS (millions of instructions per second) and the utilized network bandwidth is measured in Mbps or Gbps.**
- **Advances in CPU Processors: The advanced microprocessor chips (by Intel, NVIDIA, AMD, Qualcomm etc.) assume a multi-core architecture with dual core, quad core or more processing cores. They exploit parallelism at different levels. Moore's law has proven accurate at these levels. Moore's law is the observation that the number of transistors in a dense integrated circuit doubles approximately every two years.**

Technologies for Network based Systems

- **Multi-core CPU: A multi-core processor is a single computing component with two or more independent actual processing units (called "cores"), which are units that read and execute program instructions [9]. (Ex: add, move data, and branch). The multiple cores can run multiple instructions at the same time, increasing overall speed for programs open to parallel computing.**
- **Many-core GPU: (Graphics Processing Unit) Many-core processors are specialist multi-core processors designed for a high degree of parallel processing, containing a large number of simpler, independent processor cores [10]. Many-core processors are used extensively in embedded computers and high-performance computing. (Main frames, super computers).**

Technologies for Network based Systems

- **GPU Computing: A GPU is a graphics co-processor mounted on a computer's graphics card to perform high level graphics tasks in video editing apps. (Ex: Intel Xeon, NVIDIA). A modern GPU chip can be built with hundreds of processing cores. These days, parallel GPUs or GPU clusters are gaining more attention.**
  - **Starting as co-processors attached to the CPU, the GPUs these days possess 128 cores on a single chip (NVIDIA). Hence they have 1024 threads (128*8) executing tasks concurrently, on a single GPU. This can be termed as massive parallelism at multicore and multi-threading levels. GPUs are not restricted to videos only – they can be used in HPC systems to super computers for handling high level calculations in parallel.**

GPU Programming Model

- **GPU Programming Model: Figure 1.7 and 1.8 [2] show the interaction between a CPU and GPU in performing parallel execution of floating-point operations concurrently.**
  - ***Floating-point operations*** **involve floating-point numbers and typically take longer to execute than simple binary integer operations. A GPU has hundreds of simple cores organised as multiprocessors. Each core can have one or more threads. The CPU instructs the GPU to perform massive data processing where the bandwidth must be matched between main memory and GPU memory.**

GPU Programming Model

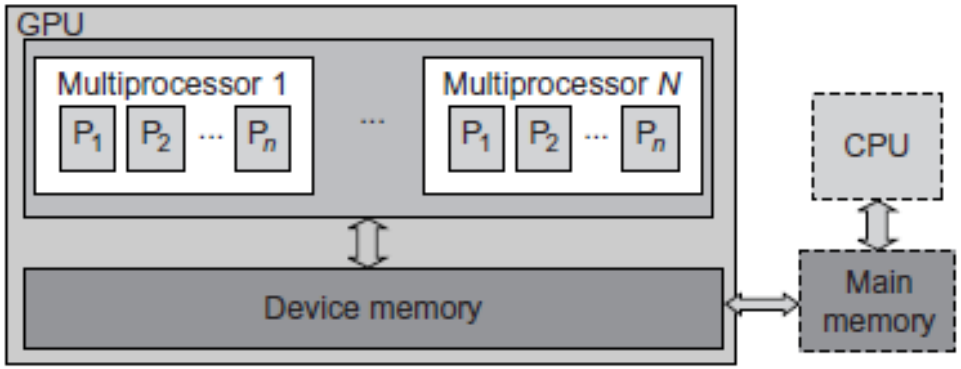


FIGURE 1.7

The use of a GPU along with a CPU for massively parallel execution in hundreds or thousands of processing cores.

GPU Programming Model

NOTE: Bandwidth is the bit-rate of available or consumed information capacity expressed typically in metric multiples of bits per second. Variously, bandwidth may be characterized as network bandwidth, data bandwidth, or digital bandwidth.

In future, thousand-core GPUs may feature in the field of Eflops/1018 flops systems.

GPU Programming Model

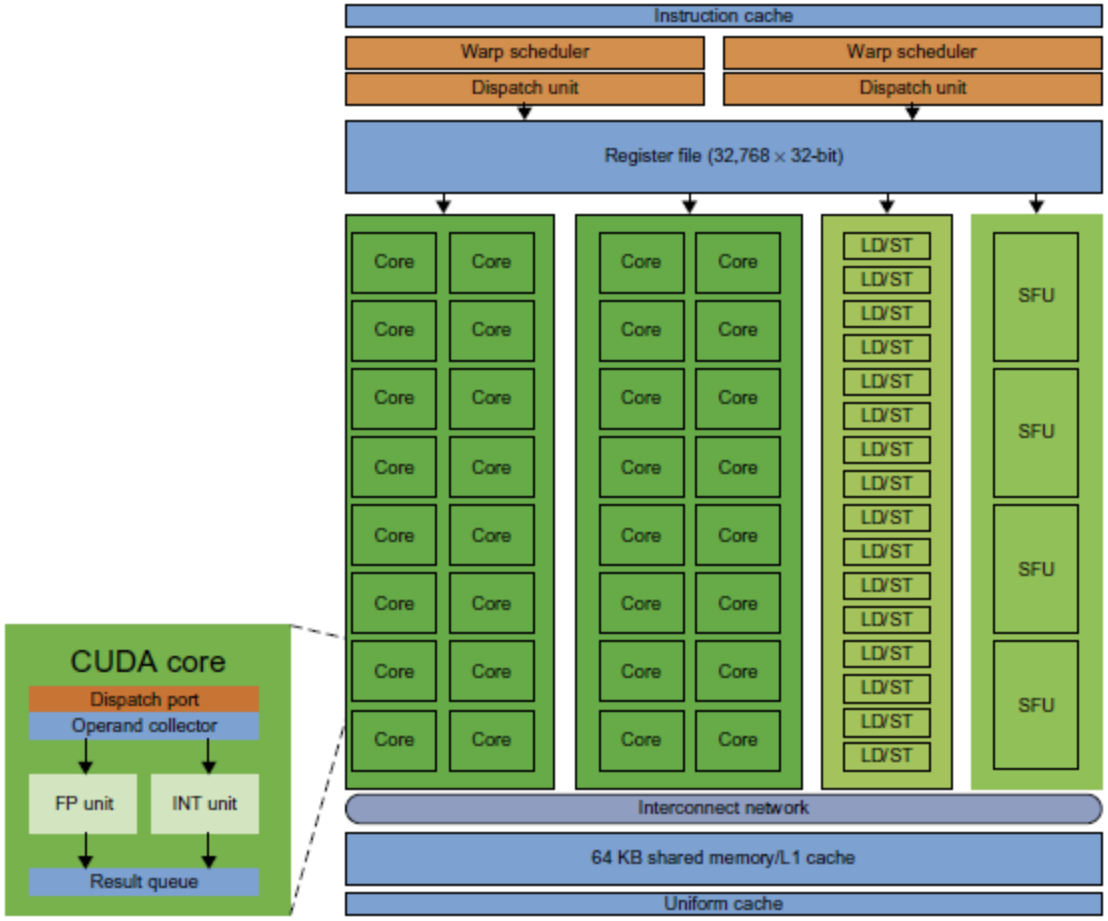


FIGURE 1.8

NVIDIA Fermi GPU built with 16 streaming multiprocessors (SMs) of 32 CUDA cores each; only one SM Is shown. More details can be found also in [49].

Power Efficiency of the GPU

- **Power Efficiency of the GPU: The major benefits of GPU over CPU are power and massive parallelism. Estimation says that 60 Gflops/watt per core is needed to run an exaflops system. [One exaflops is a thousand petaflops or a quintillion, 10^{18}, floating point operations per second]. A GPU chip requires one-tenth less of the power that a CPU requires. (Ex: CPU: 100, GPU: 90).**
- **CPU is optimized (use most effectively) for latency (time between request and response) in caches and memory; GPU is optimized for throughput with explicit (open) management of on-chip memory.**
- **Both power consumption and software are the future challenges in parallel and distributed systems.**

Power Efficiency of the GPU

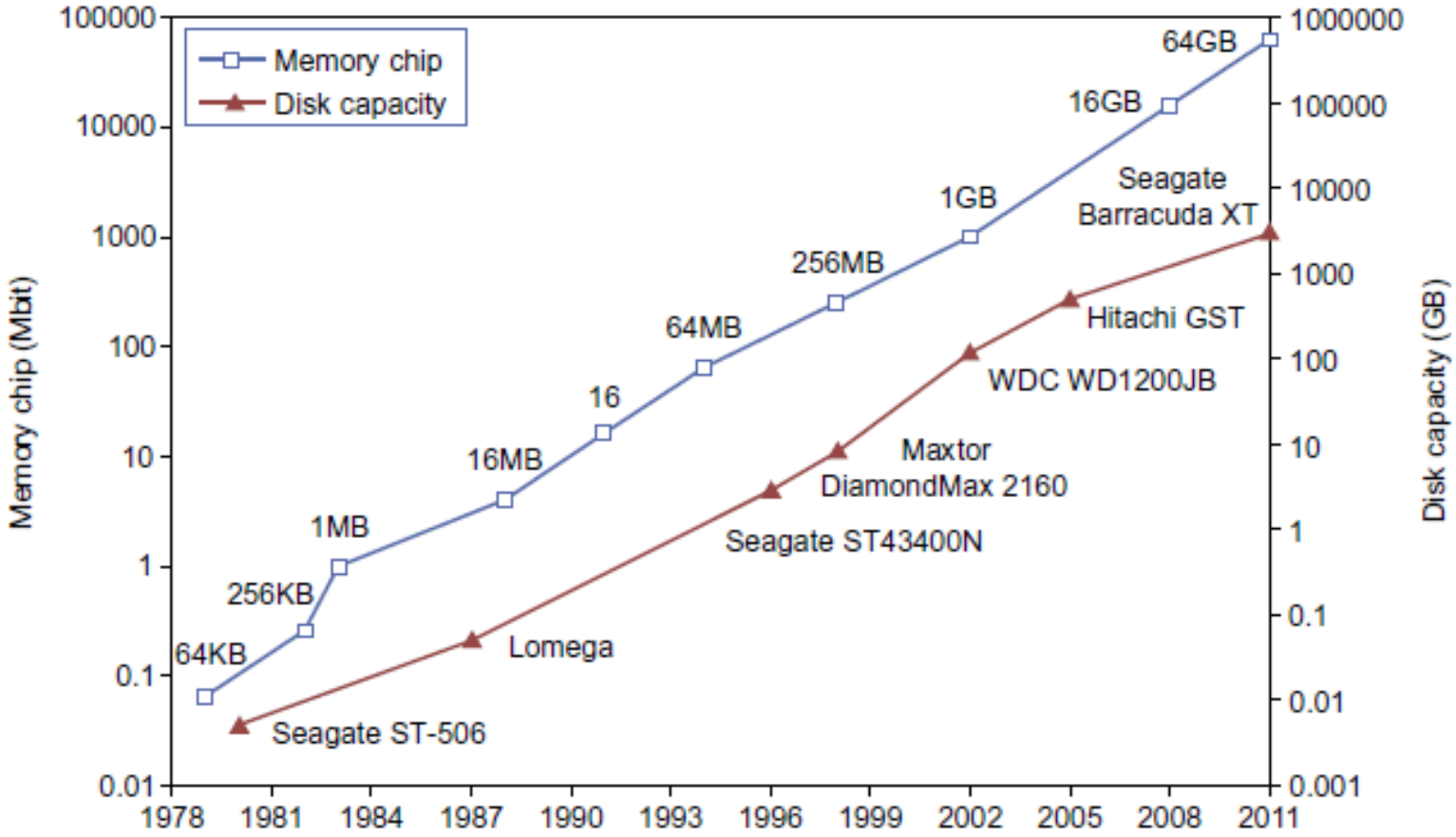


FIGURE 1.10

Improvement in memory and disk technologies over 33 years. The Seagate Barracuda XT disk has a capacity of 3 TB in 2011.

Memory, Storage and WAN:

- **Memory Technology: The upper curve in Figure 1.10 shows the growth of DRAM chip capacity from 16 KB to 64 GB. [SRAM is Static RAM and is 'static' because the memory does not have to be continuously refreshed like Dynamic RAM. SRAM is faster but also more expensive and is used inside the CPU. The traditional RAMs in computers are all DRAMs]. For hard drives, capacity increased from 260 MB to 3 TB and lately 5 TB (by Seagate). Faster processor speed and higher memory capacity will result in a wider gap between processors and memory, which is an ever-existing problem.**
- **Disks and Storage Technology: The rapid growth of flash memory and solid-state drives (SSD) also has an impact on the future of HPC and HTC systems. An SSD can handle 300,000 to 1 million write cycles per block, increasing the speed and performance. Power consumption should also be taken care-of before planning any increase of capacity.**

Memory, Storage and WAN:

- System-Area Interconnects: The nodes in small clusters are interconnected by an Ethernet switch or a LAN. As shown in Figure 1.11 [2], a LAN is used to connect clients to servers. A Storage Area Network (SAN) connects servers to network storage like disk arrays. Network Attached Storage (NAS) connects clients directly to disk arrays. All these types of network appear in a large cluster built with commercial network components (Cisco, Juniper). If not much data is shared (overlapped), we can build a small cluster with an Ethernet Switch + copper cables to link to the end machines (clients/servers).

Memory, Storage and WAN:

(a) **System-Area Interconnects**: The nodes in small clusters are interconnected by an Ethernet switch or a LAN. As shown in Figure 1.11 [2], a LAN is used to connect clients to servers. A Storage Area Network (SAN) connects servers to network storage like disk arrays. Network Attached Storage (NAS) connects clients directly to disk arrays. All these types of network appear in a large cluster built with commercial network components (Cisco, Juniper). If not much data is shared (overlapped), we can build a small cluster with an Ethernet Switch + copper cables to link to the end machines (clients/servers).

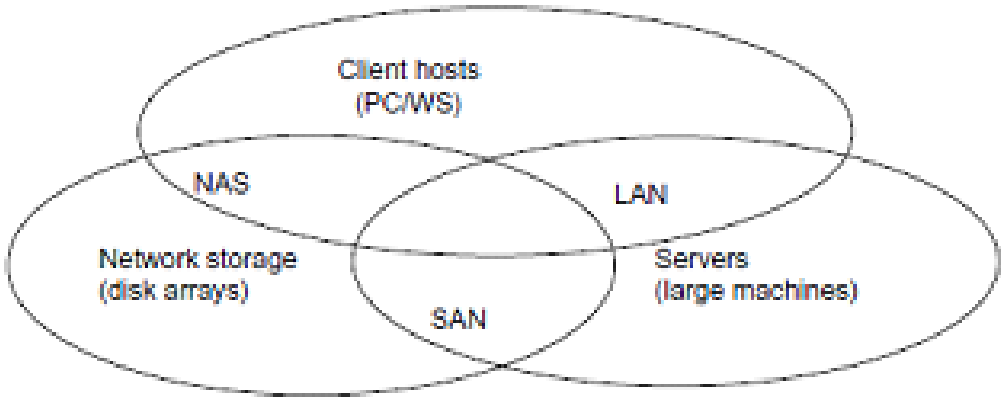


FIGURE 1.11

Three interconnection networks for connecting servers, client hosts, and storage devices; the LAN connects client hosts and servers, the SAN connects servers with disk arrays, and the NAS connects clients with large storage systems in the network environment.

Memory, Storage and WAN:

- **WAN: We can also notice the rapid growth of Ethernet bandwidth from 10 Mbps to 1 Gbps and still increasing. Different bandwidths are needed for local, national, and international levels of networks. It is also estimated that computers will be used concurrently in the coming future and higher bandwidth will certainly add more speed and capacity to aid the cloud/distributed computing. Note that most data centers use gigabit Ethernet as interconnect in their server clusters.**

Virtual Machines and Middleware

- **Virtual Machines and Middleware: A typical computer has a single OS image at a time. This leads to a rigid architecture that tightly couples apps to a specific hardware platform i.e., an app working on a system might not work on another system with another OS (non-portable).**
- **To build large clusters, grids and clouds, we need to increase the capacity of computing, storage and networking resources in a virtualized manner. A cloud of limited resources should aggregate all these dynamically to bring out the expected results.**

Virtual Machines and Middleware

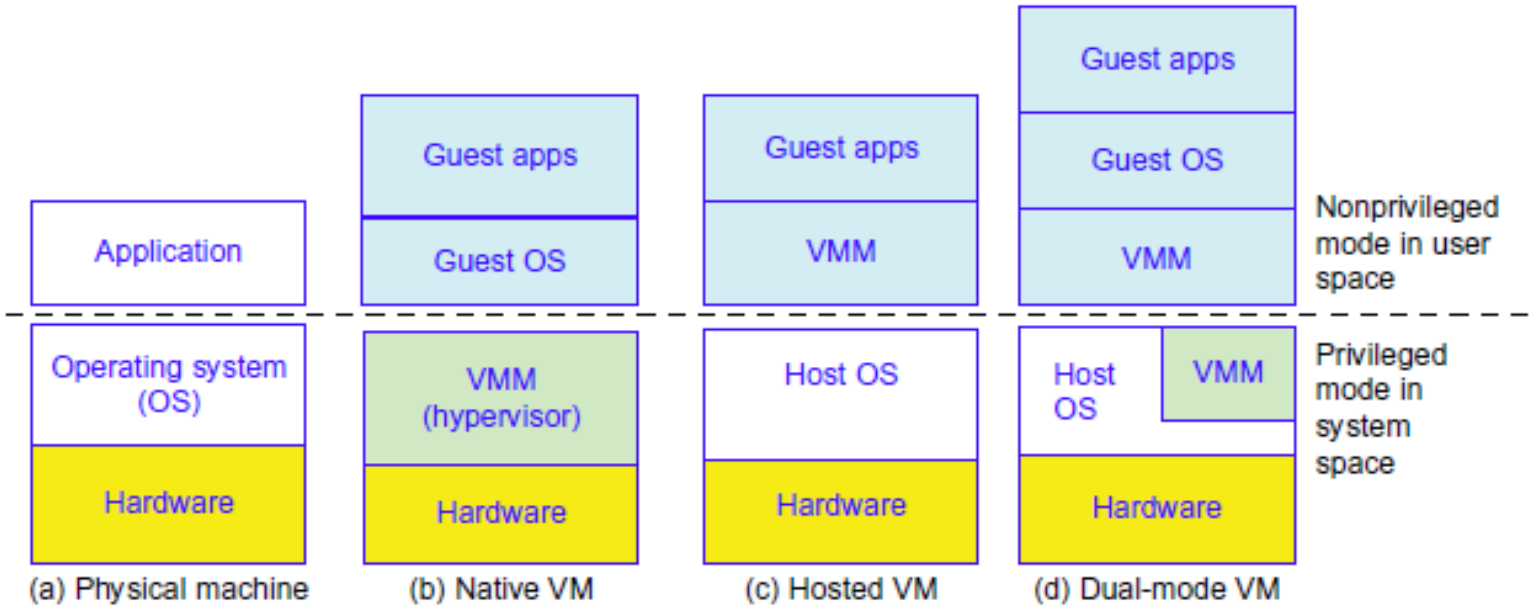


FIGURE 1.12

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

(Courtesy of M. Abde-Majeed and S. Kulkarni, 2009 USC)

Virtual Machines and Middleware

- **Virtual Machines: As seen in Figure 1.12 [2], the host machine is equipped with a physical hardware. The VM is built with virtual resources managed by a guest OS to run a specific application (Ex: VMware to run Ubuntu for Hadoop). Between the VMs and the host platform we need a middleware called VM Monitor (VMM). A hypervisor (VMM) is a program that allows different operating systems to share a single hardware host. This approach is called bare-metal VM because a hypervisor handles CPU, memory and I/O directly. VM can also be implemented with a dual mode as shown in Figure 1.12 (d). Here, part of VMM runs under user level and another part runs under supervisor level.**
- **NOTE: The VM approach provides hardware independence of the OS and apps. The VMA can run on an OS different from that of the host computer.**

Virtual Machines and Middleware

- **VM Primitive operations: A VMM operation provides VM abstraction to the guest OS. The VMM can also export an abstraction at full virtualization so that a standard OS can run it as it would on physical hardware. Low level VMM operations are indicated in Figure 1.13 [2].**

Virtual Machines and Middleware

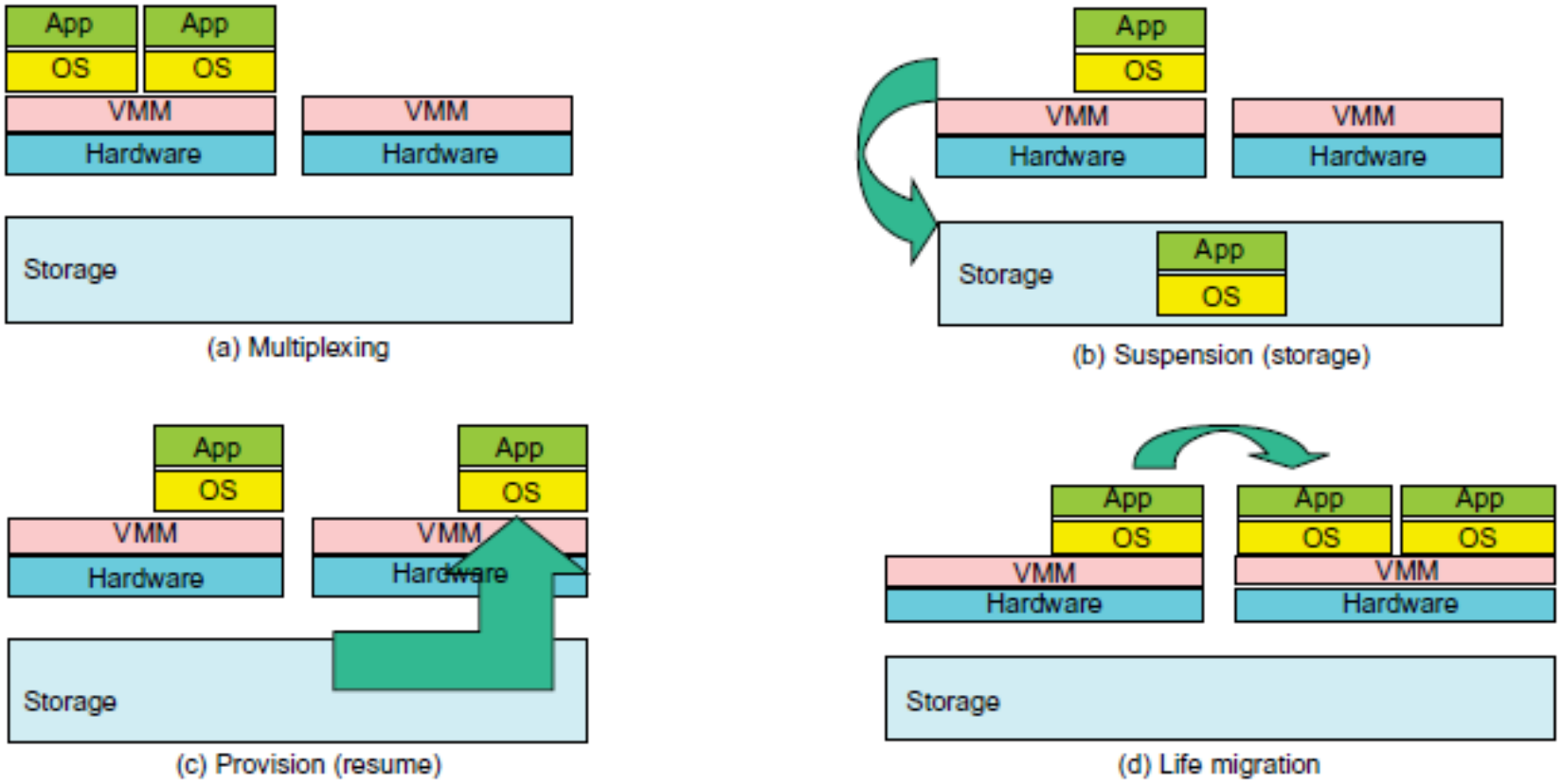


FIGURE 1.13

VM multiplexing, suspension, provision, and migration in a distributed computing environment.

(Courtesy of M. Rosenblum, Keynote address, ACM ASPLOS 2006 [41])

Virtual Machines and Middleware

- The VMs can be multiplexed between hardware machines as shown in 1.13 (a)
- A VM can be suspended and stored in a stable storage as shown in 1.13(b)
- A suspended VM can be resumed on a new hardware platform as shown in 1.13 (c)
- A VM can be migrated from one hardware platform to another as shown in 1.13 (d)

Virtual Machines and Middleware

- Advantages
- These VM operations can enable a VM to work on any hardware platform.
- They enable flexibility (the quality of bending easily without breaking) in porting distributed app executions.
- VM approach enhances the utilization of server resources – multiple server functions can be integrated on the same hardware platform to achieve higher system efficiency. [VMware claims that server resource utilization has increased from 5-15% to 60-80%].
- Eliminates server crashes due to VM usage or shows more transparency in the operations that are being carried out.

Virtual Machines and Middleware

- **Virtual Infrastructures: Virtual Infrastructure connects resources to distributed applications in such a way that a resource needed by an app is exactly mapped to it. This decreases the costs and increases efficiency and server response.**

Virtual Machines and Middleware

- **Data Center Virtualization for Cloud Computing**: Cloud architecture is built with products like hardware and network devices. Almost all cloud platforms use x86 (Family of Intel 8086 processors). Low-cost terabyte disks and gigabit Ethernet are used to build data centers. A data center takes into consideration the performance/price ratio instead of only speed.
- **Data Center Growth and Cost Breakdown**: Large data centers are built with thousands of servers and smaller ones have hundreds of the same. The cost of maintaining a data center has increased and much of this money is spent on management and maintenance which did not increase with time. Electricity and cooling also consume much of the allocated finance.
- **Low Cost Design Philosophy**: High-end switches or routers that provide high bandwidth networks cost more and do not match the financial design of cloud computing. For a fixed budget, typical switches and networks are more desirable.

Virtual Machines and Middleware

- Convergence of Technologies: CC is enabled by the convergence of technologies in four areas:
- Hardware virtualization and multi-core chips
- Utility and grid computing
- SOA, Web 2.0 and Web Service integration
- Autonomic Computing and Data Center Automation

Virtual Machines and Middleware

- Web 2.0 is the second stage of the development of the Internet, where static pages transformed into dynamic and the growth of social media.
- Data is increasing by leaps and bounds every day, coming from sensors, simulations, web services, mobile services and so on. Storage, acquisition and access of this huge amount of data sets requires standard tools that support high performance, scalable file systems, DBs, algorithms and visualization. With science becoming data-centric, storage and analysis of the data plays a huge role in the appropriate usage of the data-intensive technologies.

Cloud Computing

- Cloud Computing is basically focused on the massive data that is flooding the industry. CC also impacts the e-science where multi-core and parallel computing is required. To achieve the goals in these fields, one needs to work on workflows, databases, algorithms and virtualization issues.
- Cloud Computing is a transformative approach since it promises more results than a normal data center. The basic interaction with the information is taken up in a different approach to obtain a variety of results, by using different types of data to end up with useful analytical results.
- A typical cloud runs on an extremely large cluster of standard PCs. In each cluster node, multithreading is practised with a large number of cores in many-core GPU clusters. Hence, data science, cloud computing and multi-core computing are coming together to revolutionize the next generation of computing and take up the new programming challenges.

System Models for Cloud Computing

- System Models for Cloud Computing: Distributed and Cloud Computing systems are built over a large number of independent computer nodes, which are interconnected by SAN, LAN or WAN. Few LAN switches can easily connect hundreds of machines as a working cluster. A WAN can connect many local clusters to form large cluster of clusters. In this way, millions of computers can be brought together by using the above mentioned methodology, in a hierarchical manner.
- Large systems are highly scalable, and can reach web-scale connectivity either physically or logically. Table 1.2 [2] below shows massive systems classification as four groups: clusters, P2P networks, computing grids and Internet clouds over large data centers. These machines work collectively, cooperatively, or collaboratively at various levels.

System Models for Cloud Computing

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

System Models for Cloud Computing

- Clusters are more popular in supercomputing apps. They have laid the foundation for cloud computing. P2P are mostly used in business apps. Many grids formed in the previous decade have not been utilized per their potential due to lack of proper middleware or well-coded apps.
- NOTE: The advantages of cloud computing include its low cost and simplicity for providers and users.

Clusters of Cooperative Computers

- **Clusters of Cooperative Computers: A computing cluster consists of inter-connected standalone computers which work jointly as a single integrated computing resource. Particularly, this approach yields good results in handling heavy workloads with large datasets.**
- **The Figure 1.1.5 [2] below shows the architecture of a typical server cluster that has low latency and high bandwidth network. [Latency is the delay from input into a system to desired outcome]. For building a large cluster, an interconnection network can be utilized using Gigabit Ethernet, Myrinet or InfiniBrand switches.**

Clusters of Cooperative Computers

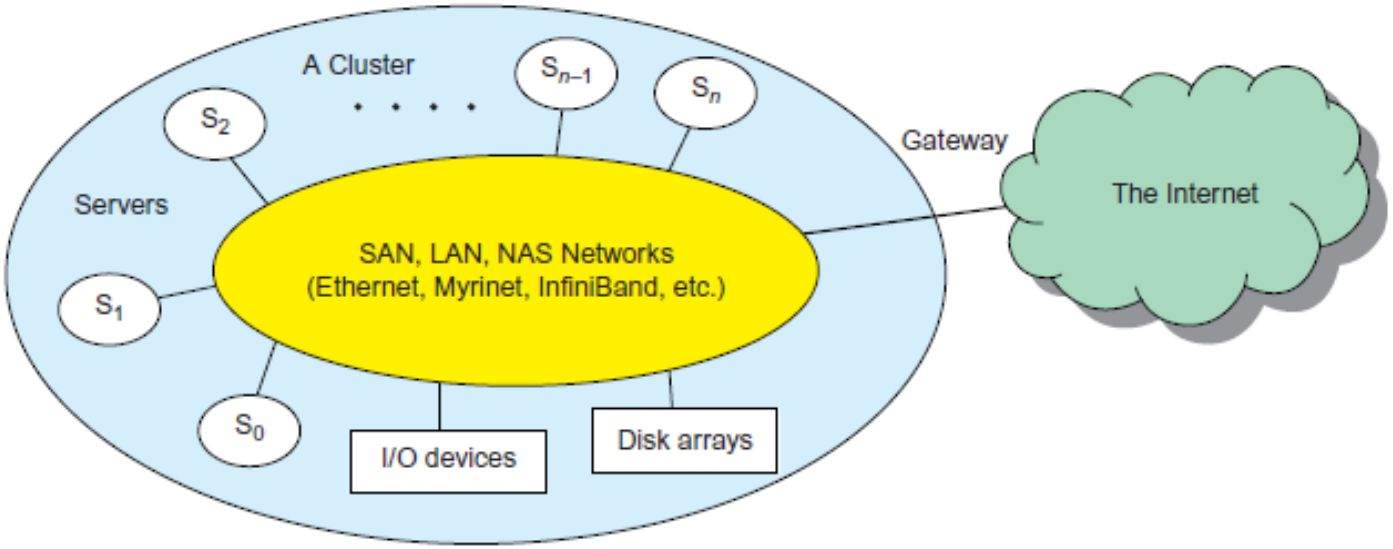


FIGURE 1.15

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

Through a hierarchical construction using SAN, LAN or WAN, scalable clusters can be built with increasing number of nodes. The concerned cluster is connected to the Internet through a VPN (Virtual Private Network) gateway, which has an IP address to locate the cluster. Generally, most clusters have loosely connected nodes, which are autonomous with their own OS.

Clusters of Cooperative Computers

- **Single-System Image (SSI): It was indicated that multiple system images should be integrated into a single-system image for a cluster. A cluster-OS is more desired these days, or a middleware to support SSI that includes sharing of CPUs, memory, I/O across all the nodes in the cluster. An SSI is an illusion (something that doesn't exist actually) that shows the integrated resources as a single and powerful resource. SSI can be created by software or hardware. Finally, a cluster is with multiple system images is only a collection of the resources of independent computers that are loosely inter-connected.**

Clusters of Cooperative Computers

- **HW, SW and MW Support: It should be noted that MPPs (Massively Parallel Processing) are clusters exploring high-level parallel processing. The building blocks here are the computer nodes (PCs, Symmetric Multi-Processors (SMPs), work stations or servers), communication software like Parallel Virtual Machine (PVM), Message Passing Interface (MPI), and a network interface card in each node. All the nodes are interconnected by high bandwidth network (Ex: Gigabit Ethernet).**
- **To create SSIs, we need special cluster middleware support. Note that both sequential and parallel apps can run on the cluster but parallel environments give effective exploitation of the resources. Distributed Shared memory (DSM) makes all the data to be shared by all the clusters, thus bringing all the resources into availability of every user. But SSI features are expensive and difficult to achieve; so users generally prefer loosely coupled machines.**

Clusters of Cooperative Computers

- **Major Cluster Design Issues: A cluster-wide OSs or a single OS controlling the cluster virtually is not yet available. This makes the designing and achievement of SSI difficult and expensive. All the apps should rely upon the middleware to bring out the coupling between the machines in cluster or between the clusters. But it should also be noted that the major advantages of clustering are scalable performance, efficient message passing, high system availability, good fault tolerance and a cluster-wide job management which react positively to the user demands.**

Grid Computing Infrastructures

- **Grid Computing Infrastructures: Grid computing is designed to allow close interaction among applications running on distant computers simultaneously.**
- **Computational Grids: A computing grid provides an infrastructure that couples computers, software/hardware, sensors and others together. The grid can be constructed across LAN, WAN and other networks on a regional, national or global scale. They are also termed as virtual platforms. Computers, workstations, servers and clusters are used in a grid. Note that PCs, laptops and others can be viewed as access devices to a grid system. Figure 1.6 [2] below shows an example grid built by different organisations over multiple systems of different types, with different operating systems.**

Grid Computing Infrastructures

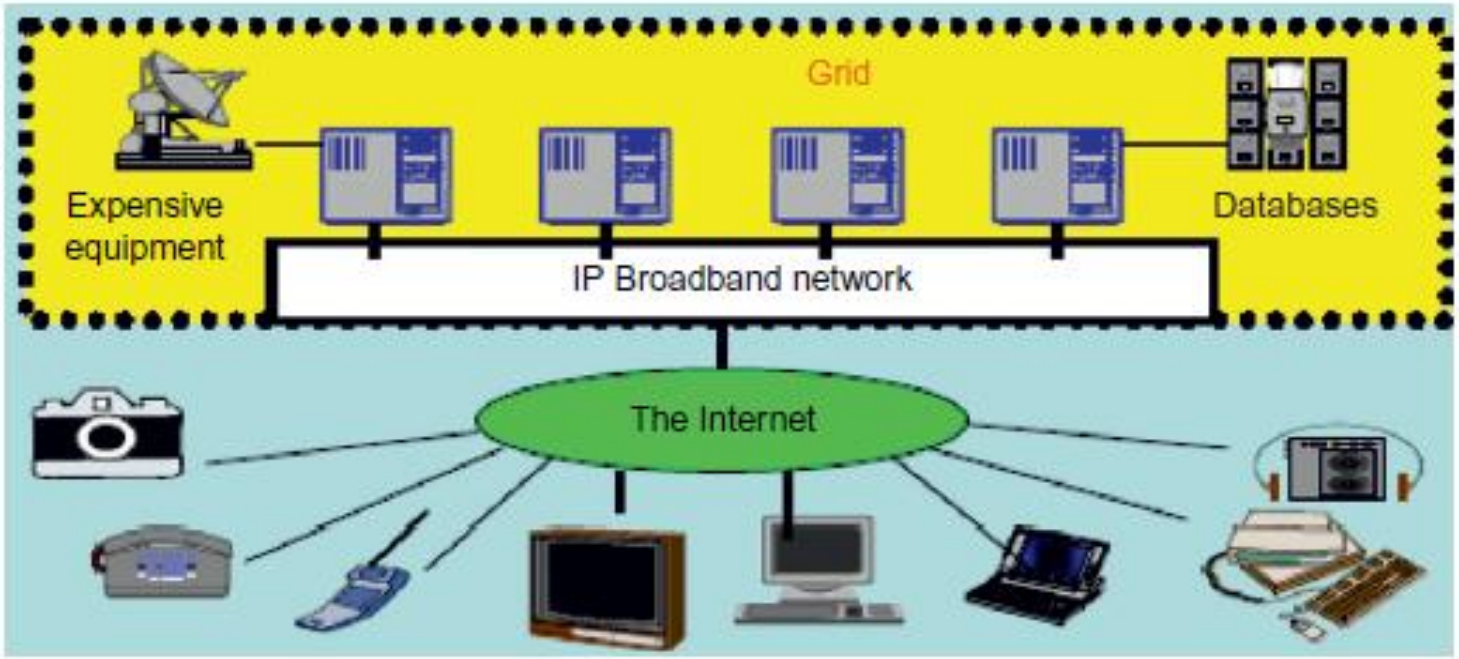


FIGURE 1.16

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

Grid Computing Infrastructures

- **Grid Families: Grid technology demands new distributed computing models, software/middleware support, network protocols, and hardware infrastructures. National grid projects are followed by industrial grid platforms by IBM, Microsoft, HP, Dell-EMC, Cisco, and Oracle. New grid service providers (GSPs) and new grid applications have emerged rapidly, similar to the growth of Internet and web services in the past two decades. Grid systems are classified in essentially two categories: computational or data grids and P2P grids. Computing or data grids are built primarily at the national level.**

Peer-to-Peer Network Families

- **Peer-to-Peer Network Families: In the basic client-server architecture, the client machines are connected to a central server for different purposes and these are essentially P2P networks. The P2P architecture offers a distributed model of networked systems. Note that P2P network is client-oriented instead of server-oriented.**
- **P2P Systems: Here, every node acts as both a client and a server. Peer machines are those connected to the Internet; all client machines act autonomously to join or leave the P2P system at their choice. No central coordination DB is needed. The system is self-organising with distributed control.**
- **Basically, the peers are unrelated. Each peer machine joins or leaves the P2P network at any time. The participating peers form the physical network at any time. This physical network is not a dedicated interconnection but a simple ad-hoc network at various Internet domains formed randomly.**

Peer-to-Peer Network Families

- **Overlay Networks: As shown in Figure 1.17 [2], an overlay network is a virtual network formed by mapping each physical machine with its ID, through a virtual mapping.**

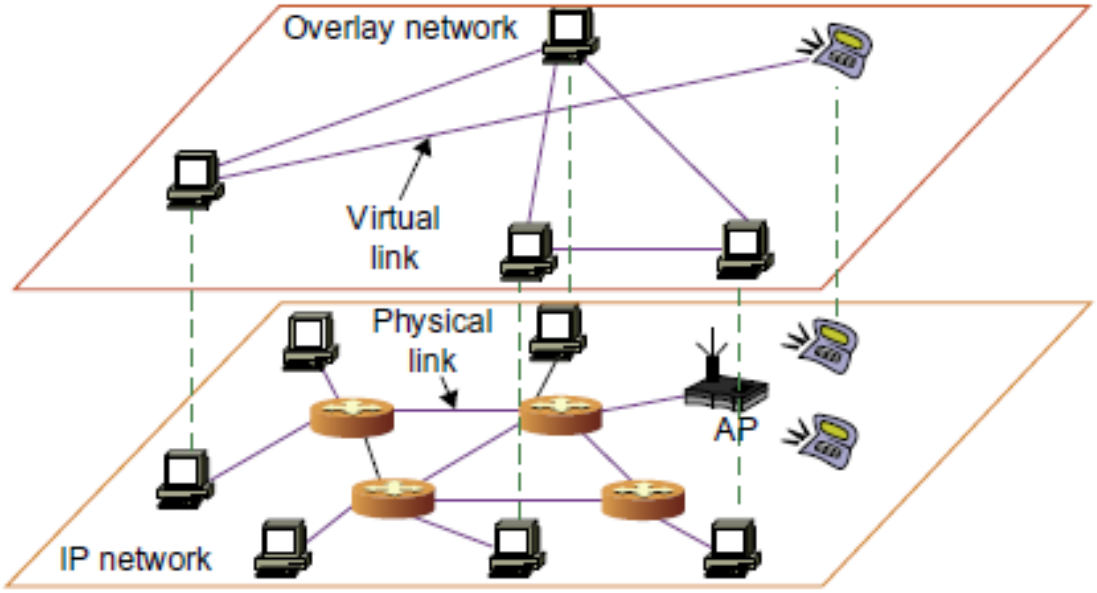


FIGURE 1.17

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

Peer-to-Peer Network Families

- If a new peer joins the system, its peer ID is added as a node in the overlay network. The P2P overlay network distinguishes the logical connectivity among the peers. The types here are unstructured and structured.
- Unstructured P2P ON is a random one and has no fixed route of contact – flooding is used to send queries to all nodes. This resulted in sudden increase of network traffic and unsure results. On the other hand, structured ONs follow a pre-determined methodology of connectivity for inserting and removing nodes from the overlay graph.

Peer-to-Peer Network Families

- **P2P Application Families: There exist 4 types of P2P networks: distributed file sharing, collaborative platform, distributed P2P computing and others. Ex: BitTorrent, Napster, Skype, Geonome, JXTA, .NET etc.**
- **P2P Computing Challenges: The main problems in P2P computing are those in hardware, software and network. Many hardware models exist to select from; incompatibility exists between the software and the operating systems; different network connections and protocols make it too complex to apply in real-time applications. Further, data location, scalability, performance, bandwidth etc. are the other challenges.**

Peer-to-Peer Network Families

- P2P performance is further affected by routing efficiency and self-organization among the peers. Fault tolerance, failure management, load balancing, lack of trust among the peers (for security, privacy and copyright violations), storage space availability are the other issues that have to be taken care of. But it should also be noted that the distributed nature of P2P network increases robustness since the failure of some peers doesn't affect the full network – fault tolerance is good.
- Disadvantages here are that since the total system is not centralized, management of the total network is difficult – anyone can logon and put in any type of data. Security is less.
- NOTE: P2P computing or networking is a distributed application architecture that partitions tasks or workloads between peers [11].
- It can be concluded that P2P networks are useful for small number of peers but not for large networks with multiple peers.

Cloud Computing over Internet

- **Cloud Computing over Internet: Cloud Computing is defined by IBM as follows: A cloud is a pool of virtualized computer resources. A cloud can host a variety of different workloads that include batch-style backend jobs and interactive and user-facing applications.**
- **Since the explosion of data the trend of computing has changed – the software apps have to be sent to the concerned data. Previously, the data was transferred to the software for computation. This is the main reason for promoting cloud computing.**
- **A cloud allows workloads to be deployed and scaled out through rapid provisioning of physical or virtual systems. The cloud supports redundant, self-recovering, and highly scalable programming models that allow workloads to recover from software or hardware failures. The cloud system also monitors the resource use in such a way that allocations can be rebalanced when required.**

Cloud Computing over Internet

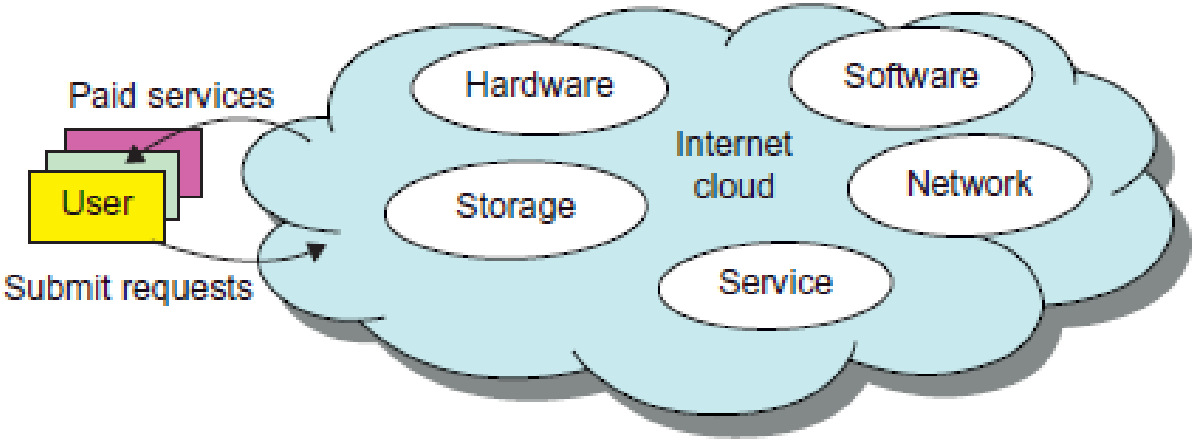


FIGURE 1.18

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

Cloud Computing over Internet

- **Internet Clouds: The idea in CC is to move desktop computing to a service-oriented platform using server clusters and huge DBs at data centers. CC benefits both users and providers by using its low cost and simple resources through machine virtualization. Many user applications are satisfied simultaneously by CC and finally, its design should satisfy the security norms, be trustworthy and dependable. CC is viewed in two ways: a centralized resource pool or a server cluster practising distributed computing.**
- **The Cloud Landscape: A distributed computing system is controlled by companies or organisations. But these traditional systems encounter several bottlenecks like constant maintenance, poor utilization, and increasing costs and updates of software or hardware. To get rid of these, CC should be utilized as on-demand computing.**

Cloud Computing over Internet

- CC offers different types of computing as services:
- Infrastructure as a Service (IaaS): This model provides different infrastructures like servers, storage, networks and the data center fabric (here, databases) to the user on demand. A typical user can deploy and run multiple VMs where guest operating systems can be used for specific applications. Note that that the user cannot manage or control the cloud infrastructure but can specify when tor request and release the concerned resources. Ex: AWS, MS Azure, Cisco Metapod, Google Compute Engine etc.
- Platform as a Service (PaaS): In this model, the user can install his own apps onto a virtualized cloud platform. PaaS includes middleware, DBs, development tools, and some computing languages. It includes both hardware and software. The provider supplies the API and the software tools (ex: Java, Python, .NET). The user need not manage the cloud infrastructure which is taken care of by the provider.

Cloud Computing over Internet

- **Software as a Service (SaaS): It is browser-initiated application software paid cloud customers. This model is used in business processes, industry applications, CRM, ERP, HR and collaborative (joint) applications. Ex: Google Apps, Twitter, Facebook, Cloudera, Salesforce etc.**

Cloud Computing over Internet

- Inter clouds offer four deployment models: private, public, managed and hybrid.
- Private Cloud: Private cloud is a type of cloud computing that delivers similar advantages to public cloud, including scalability and self-service, but through a proprietary architecture.
- Public Cloud: A public cloud is one based on the standard cloud computing model, in which a service provider makes resources, such as applications and storage, available to the general public over the Internet.
- Managed Cloud: Managed cloud hosting is a process in which organizations share and access resources, including databases, hardware and software tools, across a remote network via multiple servers in another location. [12]
- Hybrid Cloud: A hybrid cloud is an integrated cloud service utilising both private and public clouds to perform distinct functions within the same organisation. [13]

Cloud Computing over Internet

- **NOTE: The different service level agreements (SLAs) mean that the security responsibility is shared among all the cloud providers, consumers, and the third-party cloud-enabled software service providers.**

Software Environments for Distributed Systems and Clc

- **Software Environments for Distributed Systems and Clouds – SOA: In grids that use Java/CORBA, an entity is a service or an object. Such architectures build on the seven OSI layers (APSTNDP) that provide networking abstractions. Above this we have a base service environment like .NET, Java etc. and a broker network for CORBA, which enables collaboration between systems on different operating systems, programming languages and hardware [14]. By using this base, one can build a higher level environment reflecting the special features of distributed computing. The same is reflected in the figure 1.20 [2] below.**

Layered Architecture for Web Services and Grids

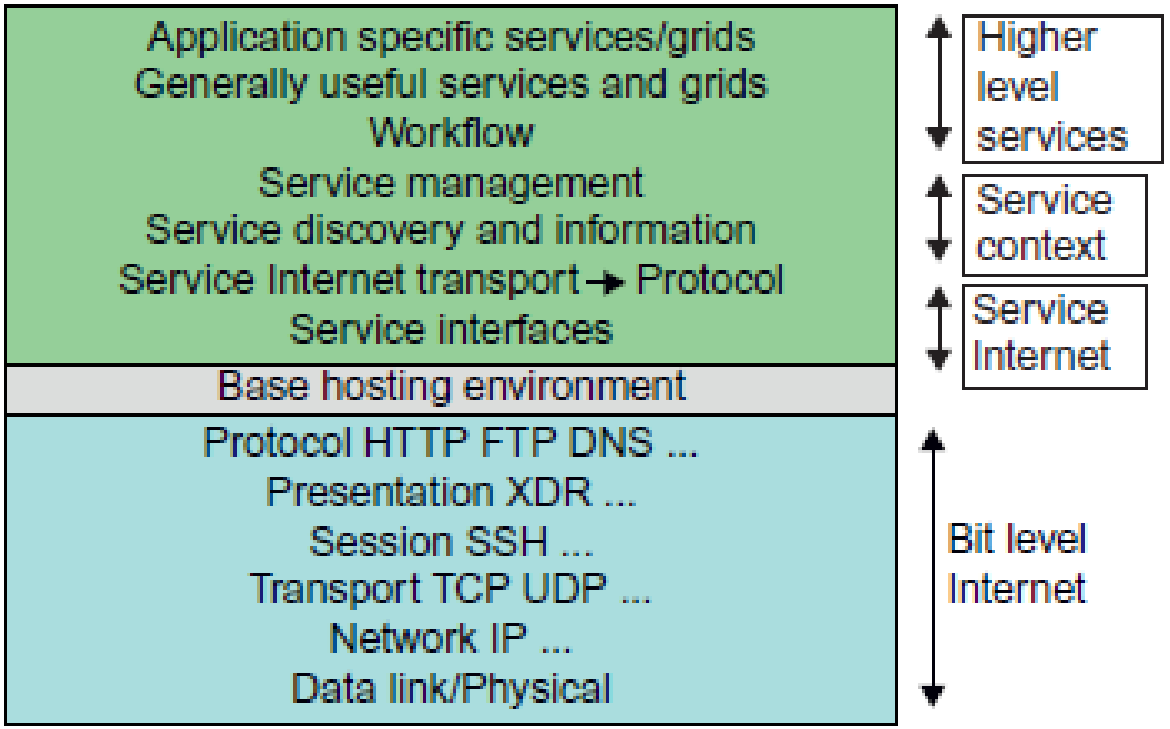


FIGURE 1.20

Layered achitecture for web services and the grids.

Layered Architecture for Web Services and Grids

- **Layered Architecture for Web Services and Grids: The entity interfaces correspond to the WSDL (web services description language) like XML, Java and CORBA interface definition language (IDL) in the distributed systems. These interfaces are linked with high level communication systems like SOAP, RMI and IIOP. These are based on message-oriented middleware infrastructures like JMS and Web Sphere MQ.**
- **At entity levels, for fault tolerance, the features in (Web Services Reliable Messaging) WSRM and its framework are same as the levels of OSI model. Entity communication is supported by higher level services for services, metadata, and the management of entities, which can be discussed later on. Ex: JNDI, CORBA trading service, UDDI, LDAP and ebXML. Note that the services have a common service: a shared memory. This enables effective exchange of information. This also results in higher performance and more throughputs.**

Layered Architecture for Web Services and Grids

- **Web Services and Tools: Loose Coupling and support of heterogeneous implementations make services (SaaS, IaaS etc.) more attractive than distributed objects. It should be realised that the above figure corresponds to two choices of service architecture: web services or (Representational State Transfer) REST systems.**
- **In web services, the aim is to specify all aspects of the offered service and its environment. This idea is carried out by using SOAP. Consequently, the environment becomes a universal distributed OS with fully distributed capability carried out by SOAP messages. But it should be noted that this approach has had mixed results since the protocol can't be agreed upon easily and even if so, it is hard to implement.**

Layered Architecture for Web Services and Grids

- The Evolution of SOA: Software Oriented Architecture applies to building grids, clouds, their combinations and even inter-clouds and systems of systems. The data collections is done through the sensors like ZigBee device, Bluetooth device, Wi-Fi access point, a PC, a mobile phone and others. All these devices interact among each other or with grids, clouds and databases at distant places.
- Raw Data->Data-> Information-> Knowledge->Wisdom-> Decisions
- Grids Vs Clouds: Grid systems apply static resources, while a cloud stresses upon elastic resources. Differences between grid and cloud exist only in dynamic resource allocation based on virtualization and autonomic computing. A 'grid of clouds' can also be built and can do a better job than a pure cloud because it can support resource allocation. Grid of clouds, cloud of grids, cloud of clouds and inter-clouds are also possible.

Distributed Operating Systems

- **Distributed Operating Systems: To promote resource sharing and fast communication, it is best to have a distributed operating system that can manage the resources efficiently. In distributed systems or more generally, a network needs an operating system itself since it deals with many heterogeneous platforms. But such an OS offers low transparency to the users.**

Distributed Operating Systems

Table 1.6 Feature Comparison of Three Distributed Operating Systems

Distributed OS Functionality	AMOEBA Developed at Vrije University [46]	DCE as OSF/1 by Open Software Foundation [7]	MOSIX for Linux Clusters at Hebrew University [3]
History and Current System Status	Written in C and tested in the European community; version 5.2 released in 1995	Built as a user extension on top of UNIX, VMS, Windows, OS/2, etc.	Developed since 1977, now called MOSIX2 used in HPC Linux and GPU clusters
Distributed OS Architecture	Microkernel-based and location-transparent, uses many servers to handle files, directory, replication, run, boot, and TCP/IP services	Middleware OS providing a platform for running distributed applications; The system supports RPC, security, and threads	A distributed OS with resource discovery, process migration, runtime support, load balancing, flood control, configuration, etc.
OS Kernel, Middleware, and Virtualization Support	A special microkernel that handles low-level process, memory, I/O, and communication functions	DCE packages handle file,time, directory, security services, RPC, and authentication at middleware or user space	MOSIX2 runs with Linux 2.6; extensions for use in multiple clusters and clouds with provisioned VMs
Communication Mechanisms	Uses a network-layer FLIP protocol and RPC to implement point-to-point and group communication	RPC supports authenticated communication and other security services in user programs	Using PVM, MPI in collective communications, priority process control, and queuing services

Distributed Operating Systems

- **Amoeba vs DCE: Distributed Computing Environment is a middleware-based system for DCEs. Amoeba was developed by academicians in Holland. But it should be noticed that DCE, Amoeba and MOSIX2 are all research prototypes used only in academia.**
- **MOSIX2 vs Linux Clusters: MOSIX is a distributed OS, which runs with a virtualization layer in the Linux environment. This layer provides a single-system image to user apps. MOSIX supports both sequential and parallel apps and the resources are discovered and migrated among the Linux nodes. (MOSIX uses Linux Kernel). A MOSIX enabled grid can extend indefinitely as long as interoperation the clusters exists.**

Distributed Operating Systems

- **Transparency in programming environments that handle user data, OS, and hardware plays a key role in the success of clouds. This concept is divided into 4 levels as seen below [2]: Data, app, OS, and hardware. Users will be able to chose the OS they like as well as the app they like – this is the main concept behind Software as a Service (SaaS).**

Distributed Operating Systems

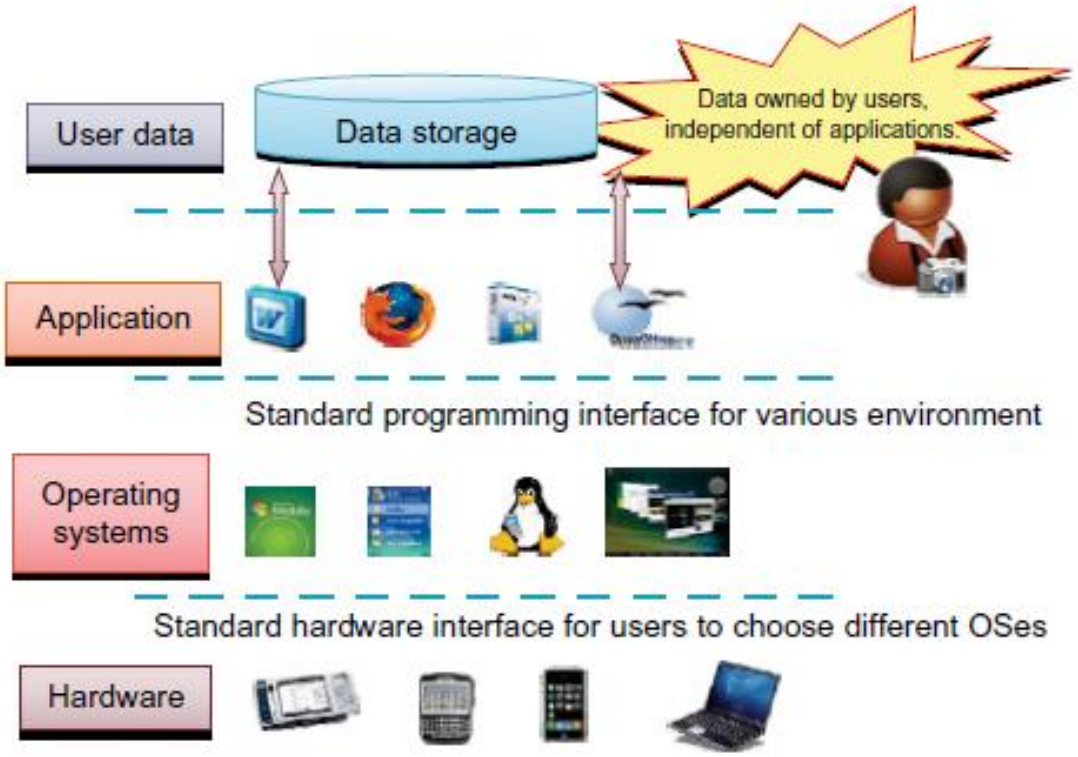


FIGURE 1.22

A transparent computing environment that separates the user data, application, OS, and hardware in time and space – an ideal model for cloud computing.

Distributed Operating Systems

- **Message-Passing Interface (MPI): MPI is a library of sub-programs that can be called from C or FORTRAN to write parallel programs running on a distributed system. The goal here is to represent clusters, grid systems, and P2P systems with upgraded web services and other utility apps. Distributed programming can also be supported by Parallel Virtual Machine (PVM).**
- **MapReduce: it is a web programming model for scalable data processing on large data clusters. It is applied mainly in web-scale search and cloud computing apps. The user specifies a Map function to generate a set of intermediate key/value pairs. Then the user applies a Reduce function to merge all intermediate values with the same (intermediate) key. MapReduce is highly scalable to explore high degrees of parallelism at different job levels and can handle terabytes of data on thousands of client machines. Many MapReduce programs can be executed simultaneously. Ex: Google's clusters.**

Distributed Operating Systems

- **Hadoop Library: Hadoop enables users to write and run apps over vast amounts of distributed data. Users can easily scale Hadoop to store and process Petabytes of data in the web space. The package is economical (open source), efficient (high level of parallelism) and is reliable (keeps multiple data copies).**
- **Open Grid Services Architecture: OGSA is driven by large-scale distributed computing apps. These apps must provide take into account high degree of resource and data sharing. The key features here are: distributed executed environment, public key infrastructure (PKI) services, trust management and security problems in grid computing.**
- **Globus is a middleware library that implements OGSA standards for resource discovery, allocation and security enforcement.**

Distributed Operating Systems

- **Performance Metrics: In a distributed system, system throughput is measured in MIPS, Tflops (Tera Floating point Operations per Second) or Transactions per Second (TPS). Other measures also exist: job response and network latency. An interconnection network with low latency and high bandwidth is preferred. The key factors to be considered for performance are OS boot time, compile time, I/O data rate, and the runtime support system used.**

Dimensions of Scalability

- **Dimensions of Scalability: System scaling can increase or decrease resources depending on different practical factors.**
- **Size Scalability: This targets higher performance or more functionality by increasing the machine size (cache, processors, memory etc.). We can determine the size scalability by counting the number of processors installed. That is more processors => more 'size'.**
- **Software Scalability: Upgrades in OS/compilers, adding mathematical libraries, installing new apps, and using more user friendly environments are the factors considered in determining software scalability.**
- **Application Scalability: This refers to matching problem size scalability (increasing data) with machine size scalability (effectively use the resources to obtain the best result possible).**

Dimensions of Scalability

- **Technology Scalability: Here, systems that can adapt to changes in different aspects of technology like component or network are considered. Three aspects play an important role here: time, space and heterogeneity. Time is concerned with processors, motherboard, power supply packaging and cooling. All these have to be upgraded between 3 to 5 years. Space is related to packaging and energy concerns. Heterogeneity refers to the use of hardware components or software packages from different vendors; this affects scalability the most.**

Threats to networks and systems

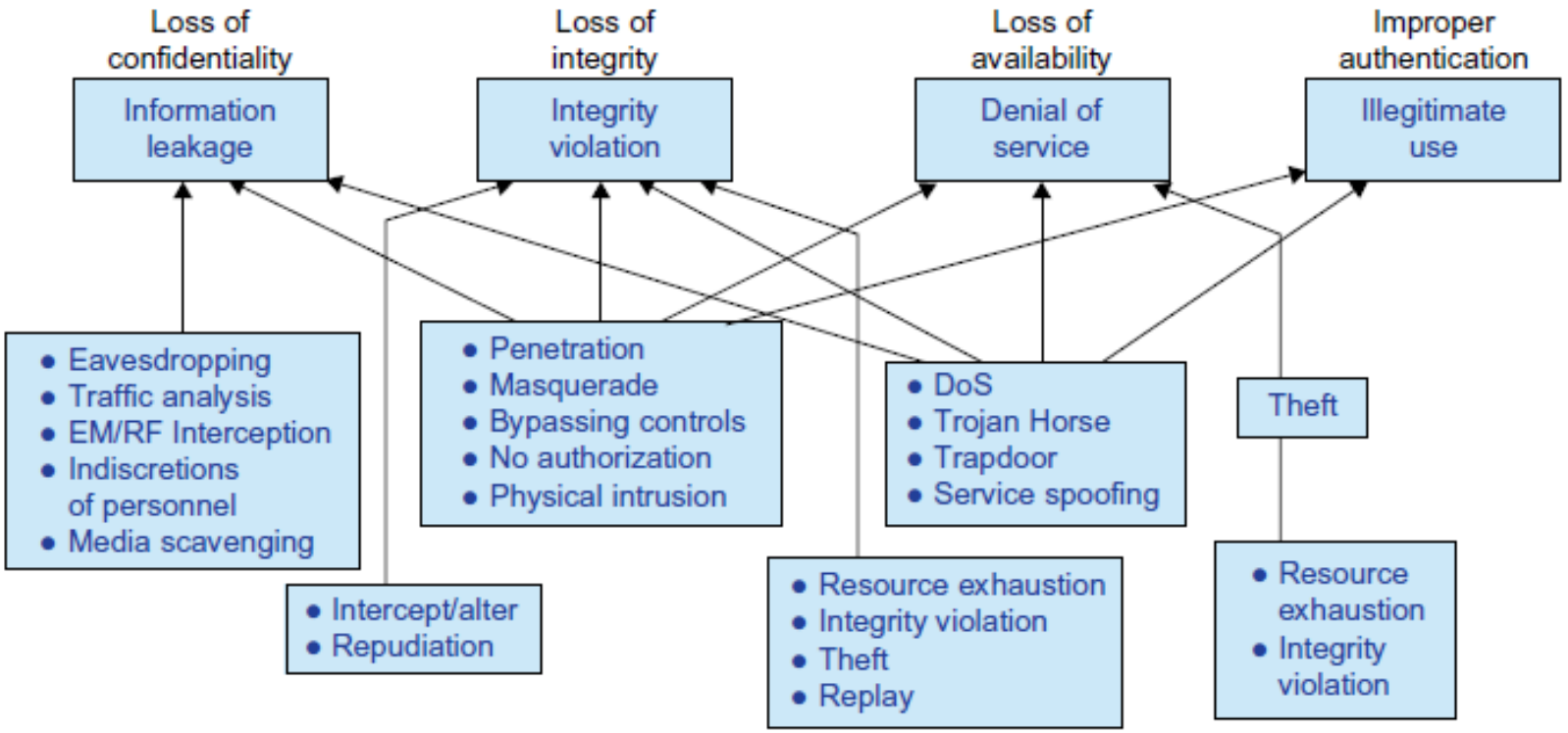


FIGURE 1.25

Various system attacks and network threats to the cyberspace, resulting 4 types of losses.

Dimensions of Scalability

- **Security Responsibilities: The main responsibilities include confidentiality, integrity and availability for most Internet service providers and cloud users. In the order of SaaS, PaaS and IaaS, the providers increase/transfer security control to the users. IN brief, the SaaS model relies on the cloud provider for all the security features. On the other hand, IaaS wants the users to take control of all security functions, but their availability is still decided by the providers. Finally, the PaaS model divides the security aspects in this way: data integrity and availability is with the provider while confidentiality and privacy control is the burden of the users.**
- **Copyright Protection: Collusive (secret agreement) piracy is the main source of copyright violation within the boundary of a P2P network. Clients may illegally share their software, allotted only to them, with others thus triggering piracy. One can develop a proactive (control the situation before damage happens) content poisoning scheme to stop colluders (conspirers) and pirates, detect them and stop them to proceed in their illegal work.**

Dimensions of Scalability

- System Defence Technologies: There exist three generations of network defence. In the first generation, tools were designed to prevent intrusions. These tools established themselves as access control policies, cryptographic systems etc. but an intruder can always slip into the system since there existed a weak link every time. The second generation detected intrusions in a timely manner to enforce remedies. Ex: Firewalls, intrusion detection systems (IDS), public key infrastructure (PKI) services (banking, e-commerce), reputation systems etc. The third generation provides more intelligent responses to intrusions.
- Data Protection Infrastructure: Security infrastructure is required to protect web and cloud services. At the user level, one needs to perform trust negotiation and reputation aggregation over all users. At the app end, we need to establish security precautions and intrusion detection systems to restrain virus, worm, malware, and DDoS attacks. Piracy and copyright violations should also be detected and contained. These can be studied in detail later when the three types of clouds are encountered and the general services offered by the cloud are discussed.

Dimensions of Scalability

- **Energy Efficiency in Distributed Computing: The primary goals in parallel and distributed computing systems are HP and HT and also performance reliability (fault tolerance and security). New challenges encountered in this area (distributed power management-DPM) these days include energy efficiency, workload and resource outsourcing. In the forth-coming topics, the energy consumption issues in servers and HPC systems are discussed.**
- **Energy consumption in parallel and distributed computing raises different issues like monetary (financial), environmental and system performance issues. The megawatts of power needed for PFlops has to be within the budget control and the distributed usage of resources has to be planned accordingly. The rising of temperature due to more usage of the resources (cooling) is also to be addressed.**

Dimensions of Scalability

- **Energy Consumption of Unused Servers: To run a data center, a company has to spend huge amount of money for hardware, software, operational support and energy every year. Hence, the firm should plan accordingly to make maximum utilization of the available resources and yet the financial and cooling issues should not cross their limits. For all the finance spent on a data center, it should also not lie down idle and should be utilized or leased for useful work.**
- **Idle servers can save a lot of money and energy; so the first step in IT departments is to identify the unused or underused servers and plan to utilize their resources in a suitable manner.**

Reducing Energy in Active Servers

- **Reducing Energy in Active Servers: In addition to identifying unused/underused servers for energy savings, we should also apply necessary techniques to decrease energy consumption in active distributed systems. These techniques should not hinder the performance of the concerned system. Power management issues in distributed computing can be classified into four layers, as seen in Figure 1.26 [2].**

Reducing Energy in Active Servers

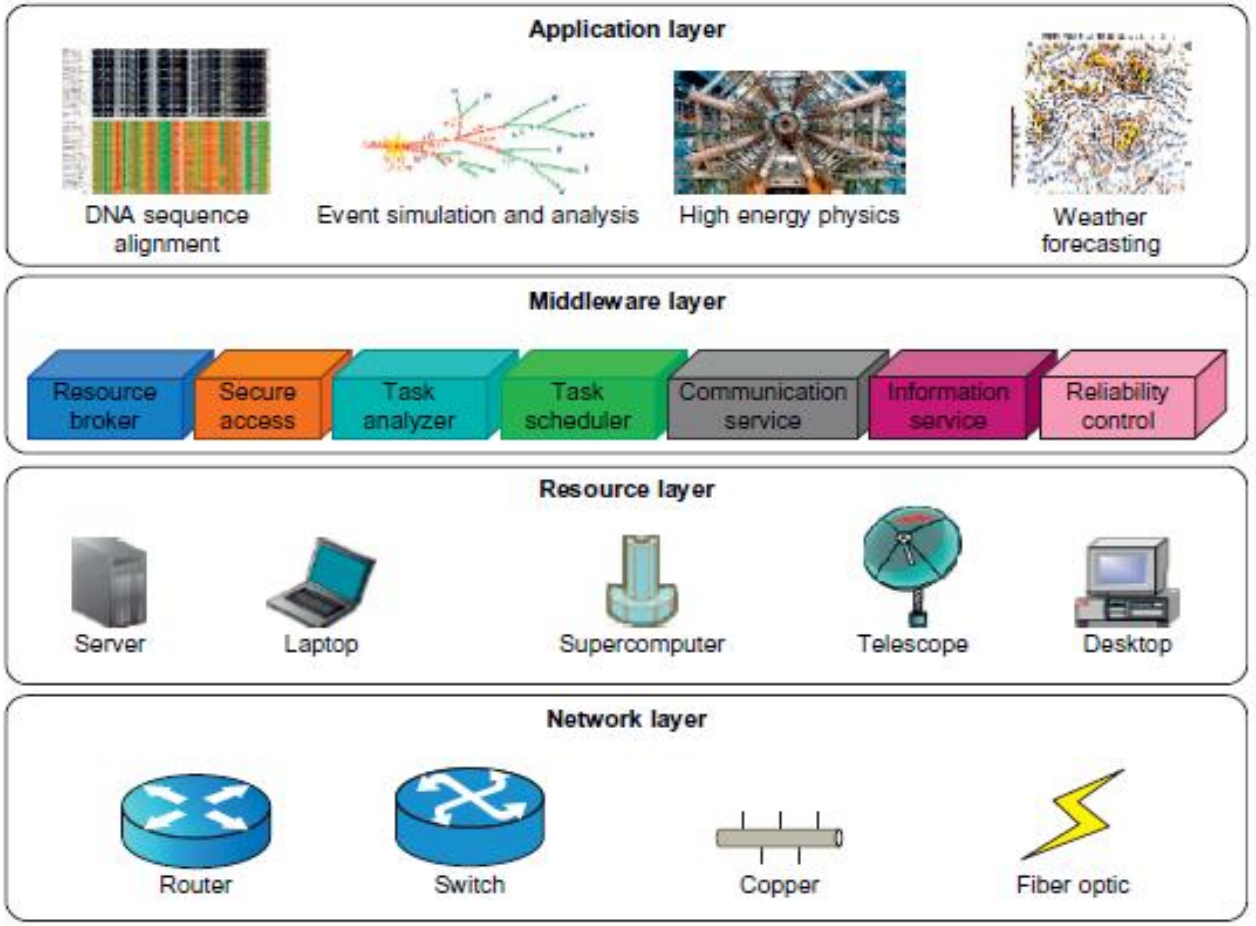


FIGURE 1.26

Four operational layers of distributed computing systems.

Reducing Energy in Active Servers

- **Application Layer: Most apps in different areas like science, engineering, business, financial etc. try to increase the system's speed or quality. By introducing energy-conscious applications, one should try to design the usage and consumption in a planned manner such that the apps manage to use the new multi-level and multi-domain energy management methodologies without reducing the performance. For this goal, we need to identify a relationship between the performance and energy consumption areas (correlation). Note that these two factors (compute and storage) are surely correlated and affect completion time.**

Reducing Energy in Active Servers

- **Middleware layer: The middleware layer is a connection between application layer and resource layer. This layer provides resource broker, communication service, task analyzer & scheduler, security access, reliability control, and information service capabilities. It is also responsible for energy-efficient techniques in task scheduling. In distributed computing system, a balance has to be brought out between efficient resource usage and the available energy.**
- **Resource Layer: This layer consists of different resources including the computing nodes and storage units. Since this layer interacts with hardware devices and the operating systems, it is responsible for controlling all distributed resources. Several methods exist for efficient power management of hardware and OS and majority of them are concerned with the processors.**

Reducing Energy in Active Servers

- **Resource Layer: This layer consists of different resources including the computing nodes and storage units. Since this layer interacts with hardware devices and the operating systems, it is responsible for controlling all distributed resources. Several methods exist for efficient power management of hardware and OS and majority of them are concerned with the processors.**
- **Dynamic power management (DPM) and dynamic voltage frequency scaling (DVFS) are the two popular methods being used recently. In DPM, hardware devices can switch from idle modes to lower power modes. In DVFS, energy savings are obtained based on the fact that power consumption in CMOS [15] (Complementary Metal-Oxide Semiconductor) circuits have a direct relationship with frequency and the square of the voltage supply. [P = 0.5 CV2f] Execution time and power consumption can be controlled by switching among different voltages and frequencies.**

Reducing Energy in Active Servers

- Network Layer: The main responsibilities of the network layer in distributed computing are routing and transferring packets, and enabling network services to the resource layer. Energy consumption and performance are to measured, predicted and balanced in a systematic manner so as to bring out energy-efficient networks. Two challenges exist here:
- The models should represent the networks systematically and should possess a full understanding of interactions among time, space and energy.
- New and energy-efficient algorithms have to be developed to rope in the advantages to the maximum scale and defend against the attacks.
- Data centers are becoming more important in distributed computing since the data is ever-increasing with the advent of social media. They are now another core infrastructure like power grid and transportation systems.

Reducing Energy in Active Servers

- **DVFS Method for Energy Efficiency: This method enables the exploitation of idle time (slack time) encountered by an inter-task relationship. The slack time associated with a task is utilized to the task in a lower voltage frequency. The relationship between energy and voltage frequency in CMOS circuits is calculated by:**

$$E = C_{eff} f v^2 t$$

$$f = K \frac{(v - v_t)^2}{v}$$

where v, Ceff, K and vt are the voltage, circuit switching capacity, a technology dependent factor and threshold voltage; t is the execution time of the task under clock frequency f. By reducing v and f, the energy consumption of the device can also be reduced.

References

- https://en.wikipedia.org/wiki/Scalability
- Kai Hwang et al, Distributed and Cloud Computing – From Parallel Processing to the Internet of Things, Morgan Kaufmann, Elsevier, 2012.
- https://en.wikipedia.org/wiki/High-throughput_computing
- https://en.wikipedia.org/wiki/Radio-frequency_identification
- https://en.wikipedia.org/wiki/Global_Positioning_System
- https://en.wikipedia.org/wiki/Internet_of_things
- https://www.phy.ornl.gov/csep/ca/node10.html
- https://en.wikipedia.org/wiki/Internet_of_things#cite_note-31
- https://en.wikipedia.org/wiki/Multi-core_processor
- https://en.wikipedia.org/wiki/Manycore_processor
- https://en.wikipedia.org/wiki/Peer-to-peer
- https://www.techopedia.com/definition/29016/managed-cloud-hosting
- http://www.interoute.com/cloud-article/what-hybrid-cloud
- https://en.wikipedia.org/wiki/Common_Object_Request_Broker_Architecture
- https://en.wikipedia.org/wiki/CMOS

$$E = C_{eff} f V^2 t$$

$$f = K \frac{(V - V_t)^2}{V}$$

UNIT II
VIRTUAL MACHINES AND VIRTUALIZATION OF CLUSTERS AND DATA CENTERS

Implementation Levels of Virtualization

- Virtualization is a concept by which several VMs are multiplexed into the same hardware machine. The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility. Hardware resources (CPU, memory, I/O devices etc.) or software resources (OS and apps) can be virtualized at various layers of functionality.
- The main idea is to separate hardware from software to obtain greater efficiency from the system. Ex: Users can gain access to more memory by this concept of VMs. With sufficient storage, any computer platform can be installed in another host computer [1], even if processors' usage and operating systems are different.

Implementation Levels of Virtualization

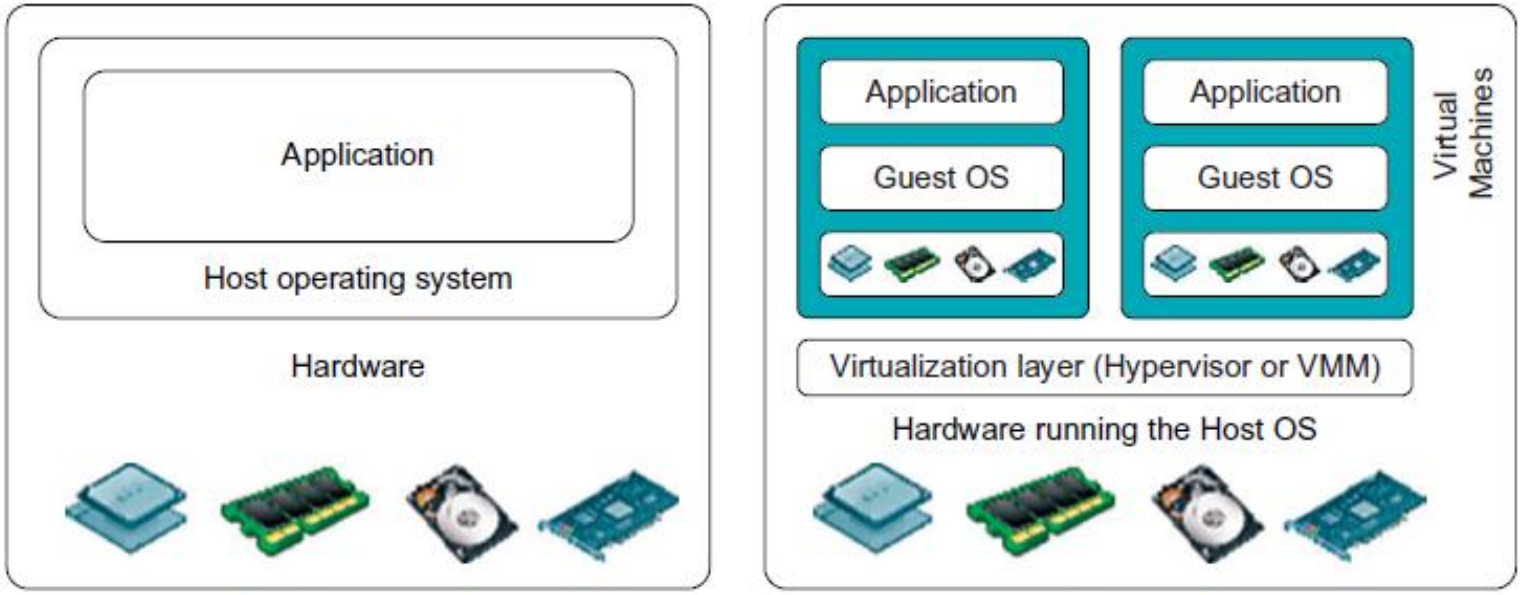


FIGURE 3.1

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

Levels of Virtualization Implementation

Implementation Levels of Virtualization

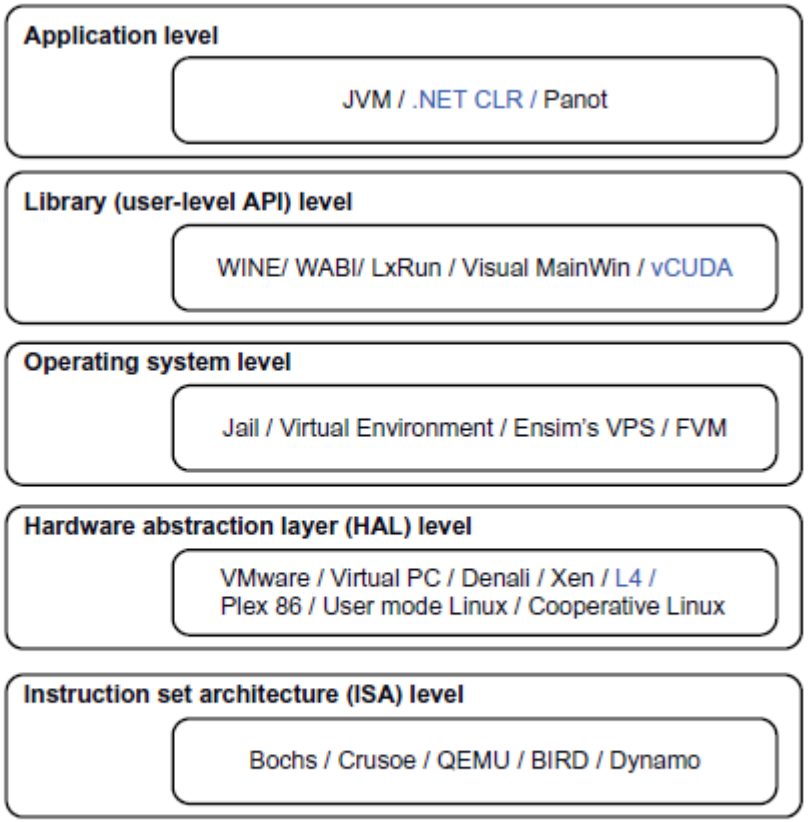


FIGURE 3.2

Virtualization ranging from hardware to applications in five abstraction levels.

Levels of Virtualization Implementation

Implementation Levels of Virtualization

- Instruction Set Architecture Level
- Hardware Abstraction Level
- OS Level
- Library Support Level
- User-App Level

Implementation Levels of Virtualization

Table 3.1 Relative Merits of Virtualization at Various Levels (More "X"'s Means Higher Merit, with a Maximum of 5 X's)

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

Note from Table 3.1 [1] that hardware and OS support will yield the highest performance. At the same time, the hardware and app levels are most expensive to implement. User isolation is difficult to archive and ISA offers best flexibility.

VMM Design Requirement and Providers

- As seen before, hardware-level virtualization inserts a layer between real hardware and traditional OS. This layer (VMM/hypervisor) manages the hardware resources of the computer effectively. By the usage of VMM, different traditional operating systems can be used with the same set of hardware simultaneously.

Requirements for a VMM

- For programs, a VMM should provide an identical environment, same as the original machine.
- Programs running in this environment should show only minor decreases in speed.
- A VMM should be in complete control of the system resources.

Requirements for a VMM

Table 3.2 Comparison of Four VMM and Hypervisor Software Packages

Provider and References	Host CPU	Host OS	Guest OS	Architecture
VMware Workstation [71]	x86, x86-64	Windows, Linux	Windows, Linux, Solaris, FreeBSD, Netware, OS/2, SCO, BeOS, Darwin	Full Virtualization
VMware ESX Server [71]	x86, x86-64	No host OS	The same as VMware Workstation	Para-Virtualization
Xen [7,13,42]	x86, x86-64, IA-64	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, Windows XP and 2003 Server	Hypervisor
KVM [31]	x86, x86-64, IA-64, S390, PowerPC	Linux	Linux, Windows, FreeBSD, Solaris	Para-Virtualization

A VMM should demonstrate efficiency in using the VMs. To guarantee the efficiency of a VMM, a statistically dominant subset of the virtual processor's instructions needs to be executed directly by the real processor with no intervention by the VMM. A comparison can be seen in Table 3.2 [1]:

Virtualization Support at the OS Level

- **Virtualization Support at the OS Level: CC is transforming the computing landscape by shifting the hardware and management costs of a data center to third parties, like banks. The challenges of CC are: (a) the ability to use a variable number of physical machines and VM instances depending on the needs of the problem. Ex: A work may need a single CPU at an instance but multi-CPUs at another instance (b) the slow operation of instantiating new VMs.**
- **As of now, new VMs originate either as fresh boots or as replicates of a VM template – unaware of the current status of the application.**

Why OS Level Virtualization

- **Why OS Level Virtualization (Disadvantages of hardware level virtualization):**
- **It is slow to initiate a hardware level VM since each VM creates its own image from the beginning.**
- **Redundancy content is high in these VMs.**
- **Slow performance and low density**
- **Hardware modifications maybe needed**

Why OS Level Virtualization

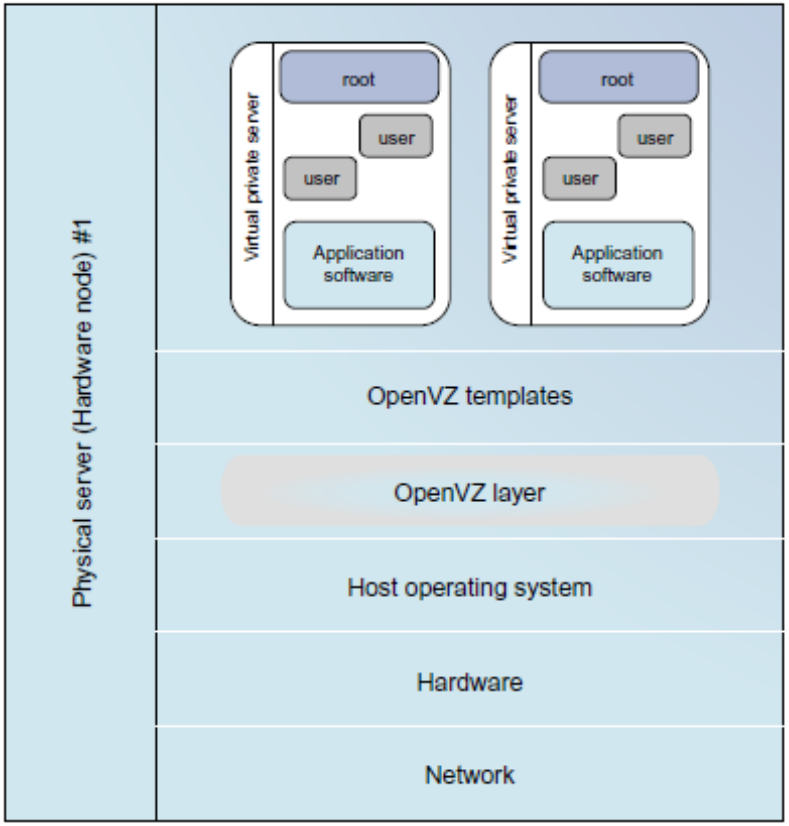


FIGURE 3.3

The OpenVZ virtualization layer inside the host OS, which provides some OS images to create VMs quickly.

Advantages of OS Extensions

- VMs at the OS level have minimal start-up shutdown costs, low resource requirements and high scalability.
- For an OS level VM, the VM and its host environment can synchronise state changes
- These can be achieved through two mechanisms of OS level virtualization:
- All OS level VMs on the same physical machine share a single OS kernel
- The virtualization layer can be designed in way that allows processes in VMs can access as many resources as possible from the host machine, but can never modify them.

Disadvantages of OS Extensions

- **Disadvantages of OS Extension: The main disadvantage of OS extensions is that all VMs at OS level on a single container must have the same kind of guest OS. Though different OS level VMs may have different OS distributions (Win XP, 7, 10), they must be related to the same OS family (Win). A Windows distribution can't run on a Linux based container.**

Virtualization on Linux or Windows Platforms

- **Virtualization on Linux or Windows Platforms: Generally, the OS-level virtualization systems are Linux-based. Windows based virtualization platforms are not much in use. The Linux kernel offers an abstraction layer to allow software processes to with and operate on resources without knowing the hardware details. Different Linux platforms use patched kernels to provide special support for extended functionality.**

Middleware Support for Virtualization

- **Middleware Support for Virtualization: This is the other name for Library-level Virtualization and is also known as user-level Application Binary Interface or API emulation. This type of virtualization can create execution environments for running alien (new/unknown) programs on a platform rather than creating a VM to run the entire OS. The key functions performed here are API call interception and remapping (assign a function to a key).**

Virtualization Structures/Tools and Mechanisms

- **Virtualization Structures/Tools and Mechanisms: It should be noted that there are three classes of VM architecture [Page 1]. Before virtualization, the OS manages the hardware. After virtualization, a virtualization layer is inserted between the hardware and the OS. Here, the virtualization layer is responsible for converting parts of real hardware into virtual hardware. Different operating systems like Windows and Linux can run simultaneously on the same machine in this manner. Depending on the position of the virtualization layer, several classes of VM architectures can be framed out: Hypervisor Architecture, para-virtualization and host-based virtualization.**

Hypervisor and Xen Architecture

- **Hypervisor and Xen Architecture: The hypervisor (VMM) supports hardware level virtualization on bare metal devices like CPU, memory, disk and network interfaces. The hypervisor software exists between the hardware and its OS (platform). The hypervisor provides hypercalls for the guest operating systems and applications. Depending on the functionality, a hypervisor can assume micro-kernel architecture like MS Hyper-V or monolithic hypervisor architecture like the VMware ESX for server virtualization.**
- **Hypercall: A hypercall is a software trap from a domain to the hypervisor, just as a syscall is a software trap from an application to the kernel. Domains will use hypercalls to request privileged operations like updating page tables.**

Hypervisor and Xen Architecture

- **Software Trap: A trap, also known as an exception or a fault, is typically a type of synchronous interrupt caused by an exceptional condition (e.g., breakpoint, division by zero, invalid memory access). A trap usually results in a switch to kernel mode, wherein the OS performs some action before returning control to the originating process. A trap in a system process is more serious than a trap in a user process and might be fatal. The term trap might also refer to an interrupt intended to initiate a context switch to a monitor program or debugger.**
- **Domain: It is a group of computers/devices on a network that are administered as a unit with common rules and procedures. Ex: Within the Internet, all devices sharing a common part of the IP address are said to be in the same domain.**

Hypervisor and Xen Architecture

- **Page Table: A page table is the data structure used by a virtual memory system in an OS to store the mapping between virtual addresses and physical addresses.**
- **Kernel: A kernel is the central part of an OS and manages the tasks of the computer and hardware like memory and CPU time.**
- **Monolithic Kernel: These are commonly used by the OS. When a device is needed, it is added as a part of the kernel and the kernel increases in size. This has disadvantages like faulty programs damaging the kernel and so on. Ex: Memory, processor, device drivers etc.**
- **Micro-kernel: In micro-kernels, only the basic functions are dealt with – nothing else. Ex: Memory management and processor scheduling. It should also be noted that OS can't run only on a micro-kernel, which slows down the OS.**
- **[SIM – Micro SIM]**

Hypervisor and Xen Architecture

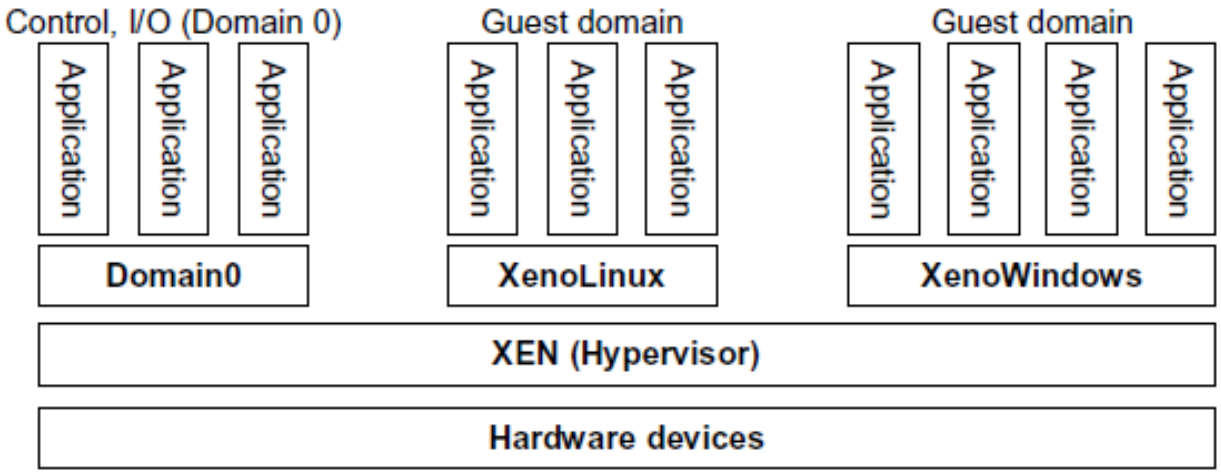


FIGURE 3.5

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

Binary Translation of Guest OS Requests Using a VMI

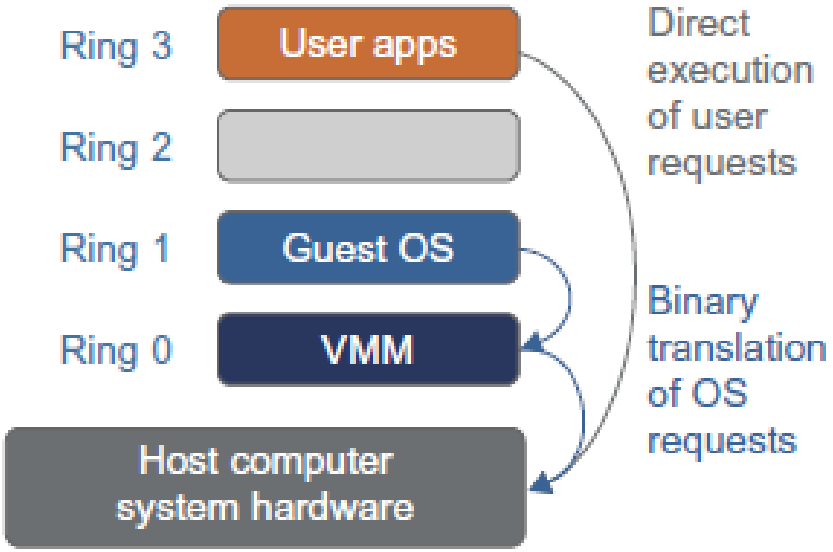


FIGURE 3.6

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.

Para-Virtualization with Compiler Support

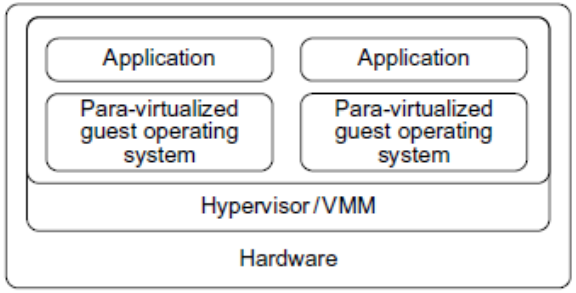


FIGURE 3.7

Para-virtualized VM architecture, which involves modifying the guest OS kernel to replace nonvirtualizable instructions with hypercalls for the hypervisor or the VMM to carry out the virtualization

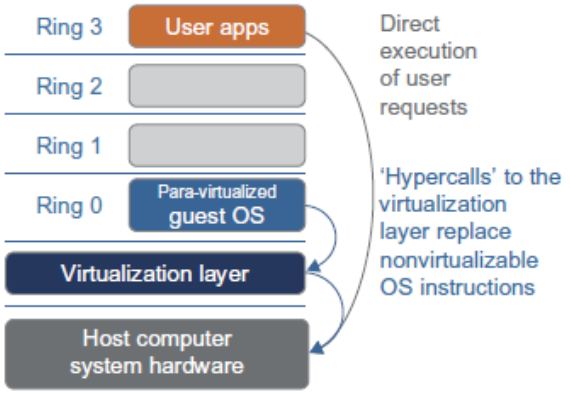


FIGURE 3.8

The use of a para-virtualized guest OS assisted by an intelligent compiler to replace nonvirtualizable OS instructions by hypercalls.

Disadvantages of Para-Virtualization

- **Disadvantages of Para-Virtualization: Although para-virtualization reduces the overhead, it has other problems. Its compatibility (suitability) and portability can be in doubt because it has to support both the modified guest OS and the host OS as per requirements. Also, the maintenance cost of para-virtualization is high since it may require deep kernel modifications. Finally, the performance advantage of para-virtualization is not stable – it varies as per the workload. But compared with full virtualization, para-virtualization is more easy and practical since binary translation is not much considered. Many products utiliza para-virtualization to overcome the less speed of binary translation. Ex: Xen, KVM, VMware ESX.**

Virtualization of CPU, Memory and I/O Devices

- **Virtualization of CPU, Memory and I/O Devices: Processors employ a special running mode and instructions, known as hardware-assisted virtualization. Through this, the VMM and guest OS run in different modes; all sensitive instructions of the guest OS and its apps are caught by the 'trap' in the VMM.**
- **H/W Support for Virtualization: Modern operating systems and processors permit multiple processes to run simultaneously. A protection mechanism should exist in the processor so that all instructions from different processes will not access the hardware directly – this will lead to a system crash.**

Hardware Assisted CPU VZ

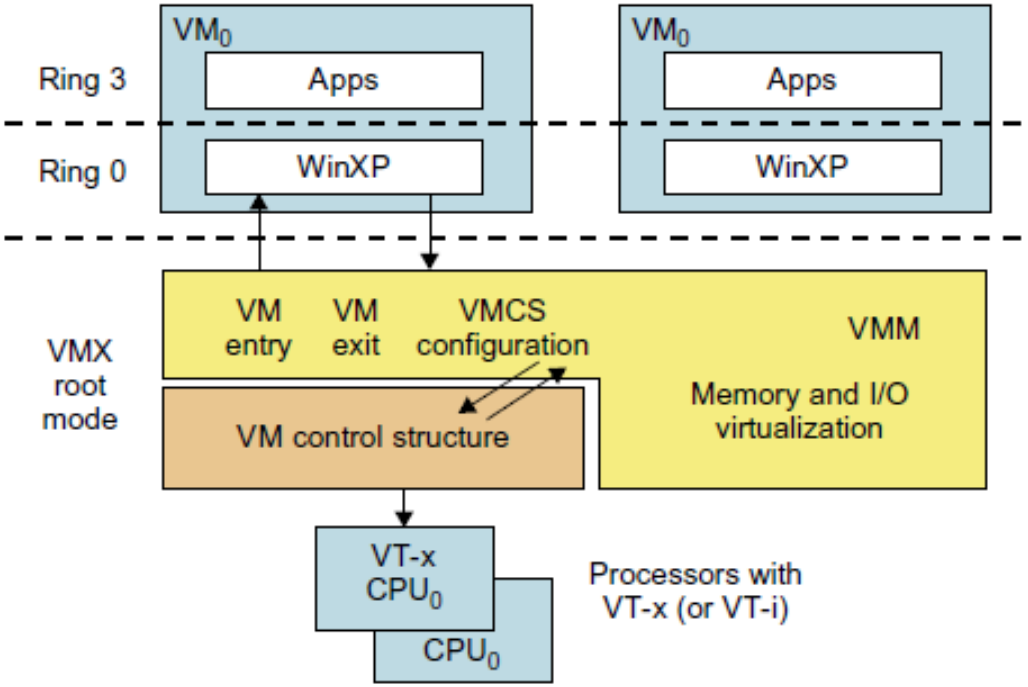


FIGURE 3.11

Intel hardware-assisted CPU virtualization.

VMCS=> VM Control System
VMX=> A virtual router

Virtualization of CPU, Memory and I/O Devices

- **Memory Virtualization: In the traditional methodology, the OS maintains mappings between virtual memory to machine memory (MM) using page tables, which is a one-stage mapping from virtual memory to MM.**
- **Virtual memory is a feature of an operating system (OS) that allows a computer to compensate for shortages of physical memory by temporarily transferring pages of data from random access memory (RAM) to disk storage**
- **Machine Memory [6] is the upper bound (threshold) of the physical memory that a host can allocate to the VM. All modern x86 processors contain memory management unit (MMU) and a translation look-aside buffer (TLB) to optimize (use in the best way) the virtual memory performance.**

Virtualization of CPU, Memory and I/O Devices

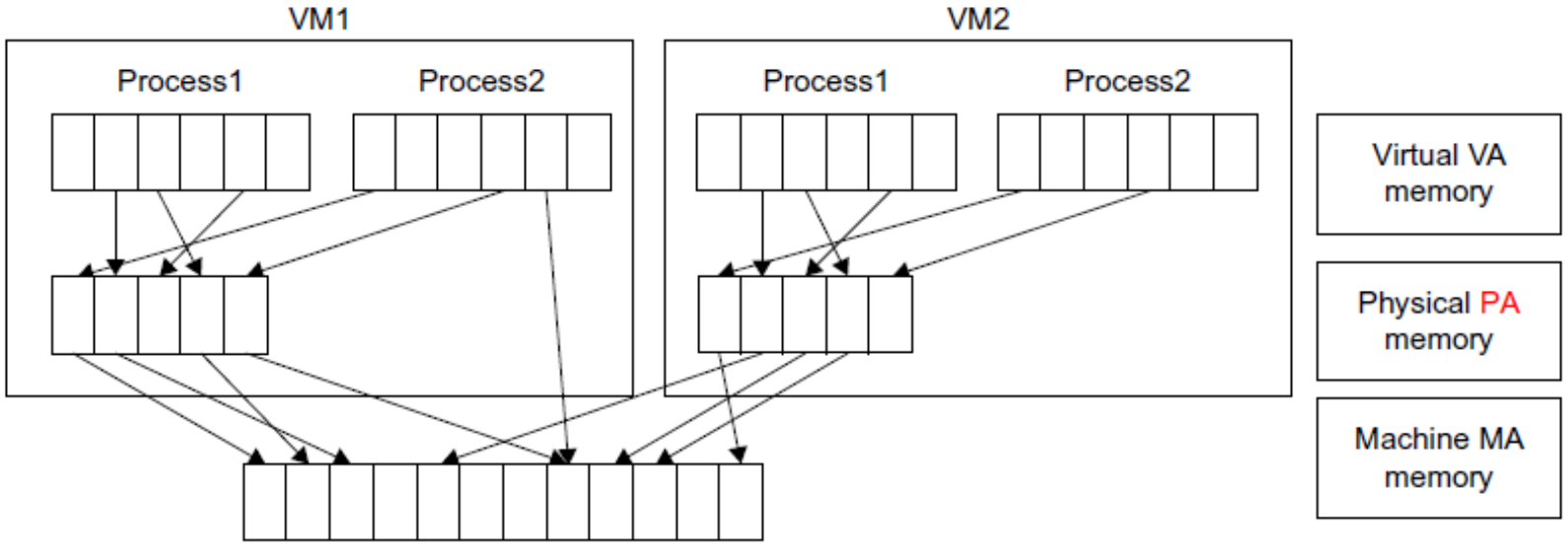


FIGURE 3.12

Two-level memory mapping procedure.

VA-Virtual Address; PA-Physical Address;
MA-Machine Address

UNIT – 3
CLOUD PLATFORM ARCHITECTURE

Cloud Computing and Service Models

- **Cloud Computing and Service Models: In recent days, the IT industry has moved from manufacturing to offering more services (service-oriented). As of now, 80% of the industry is 'service-industry'. It should be realized that services are not manufactured/invented from time-to-time; they are only rented and improved as per the requirements.**
- **Public, Private and Hybrid Clouds: Cloud computing has evolved from the concepts of clusters, grids and distributed computing. Different resources (hardware, finance, time) are leveraged (use to maximum advantage) to bring out the maximum HTC. A CC model enables the users to share resources from anywhere at any time through their connected devices.**

Cloud Computing and Service Models

- **Public, Private and Hybrid Clouds: Cloud computing has evolved from the concepts of clusters, grids and distributed computing. Different resources (hardware, finance, time) are leveraged (use to maximum advantage) to bring out the maximum HTC. A CC model enables the users to share resources from anywhere at any time through their connected devices.**

The Basic Architecture

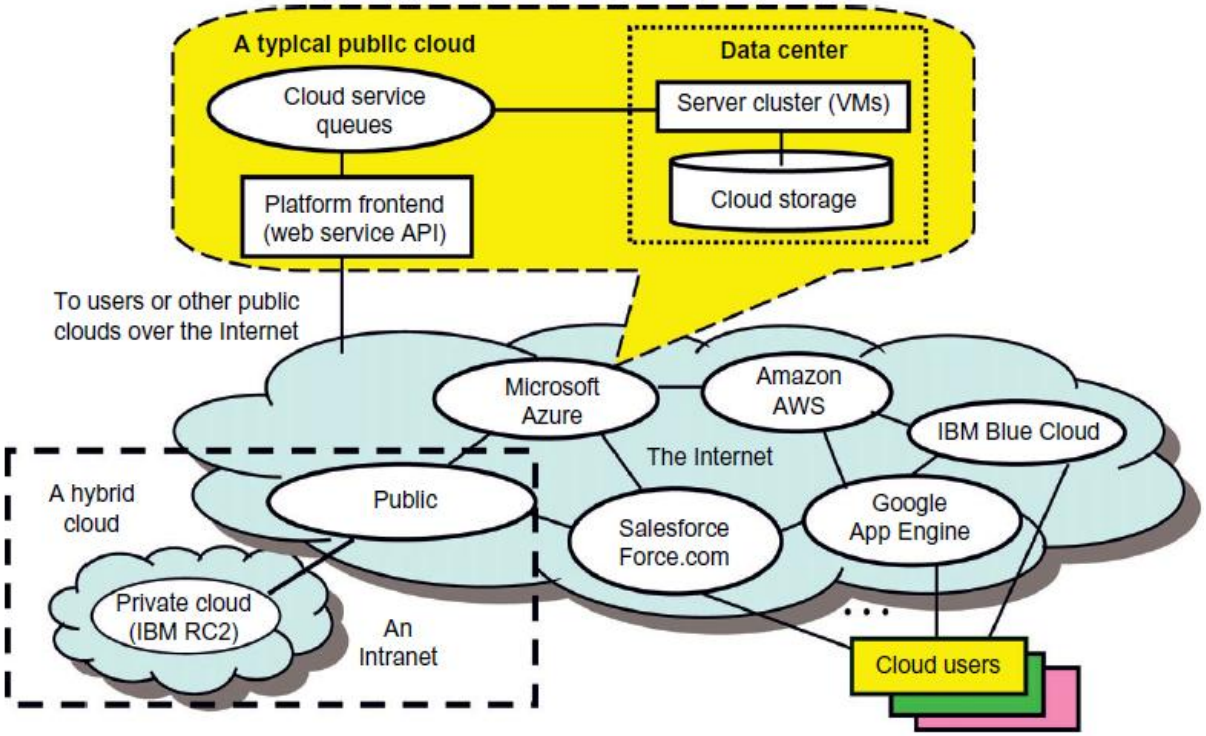


FIGURE 4.1

Public, private, and hybrid clouds illustrated by functional architecture and connectivity of representative clouds available by 2011.

Data Center Networking Architecture

- **Data Center Networking Architecture: The core of a cloud is the server cluster and the cluster nodes are used as compute nodes. The scheduling of user jobs requires that virtual clusters are to be created for the users and should be granted control over the required resources. Gateway nodes are used to provide the access points of the concerned service from the outside world. They can also be used for security control of the entire cloud platform. It is to be noted that in physical clusters/grids, the workload is static; in clouds, the workload is dynamic and the cloud should be able to handle any level of workload on demand.**

Differences between Data Centers and Super Compute

- **Differences between Data Centers and Super Computers: In data centers, scalability is a fundamental requirement. Note that data centers have multiple servers. Ex: MS-Chicago Data Center has 100,000 eight-core servers housed in 50 containers (2000 in each). In supercomputers, a separate data farm [3] is used; a data center uses disks on server nodes plus memory cache and DBs.**
- **Data Centers and Supercomputers also possess different networking requirements. (bandwidth, routers used etc.)**
- **NOTE: Data Farm => Data farming is the process of using computational experiments to 'grow' or increase data which can be utilized for statistical analyzing.**

Cloud Development Trends :

- **Cloud Development Trends: There is a good chance that private clouds will grow in the future since private clouds are more secure, and adjustable within an organization. Once they are matured and more scalable, they might be converted into public clouds. In another angle, hybrid clouds might also grow in the future.**

Cloud Ecosystem and Enabling Technologies :

- **Cloud Ecosystem and Enabling Technologies: The differences between classical computing and cloud computing can be seen in the table [1] below. In traditional computing, a user has to buy the hardware, acquire the software, install the system, test the configuration and execute the app code. The management of the available resources is also a part of this. Finally, all this process has to be revised for every 1.5 or 2 years since the used methodologies will become obsolete.**

Cloud Design Objectives:

- Shifting computing from desktops to data centers
- Service provisioning and cloud economics
- Scalability in performance (as the no. of users increases)
- Data Privacy Protection
- High quality of cloud services (QoS must be standardized to achieve this)
- New standards and interfaces

Cost Model:

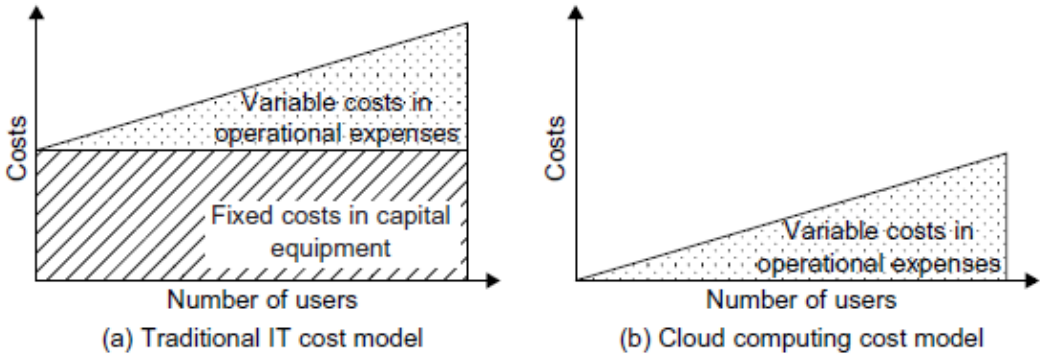


FIGURE 4.3

Computing economics between traditional IT users and cloud users.

Cloud Ecosystems

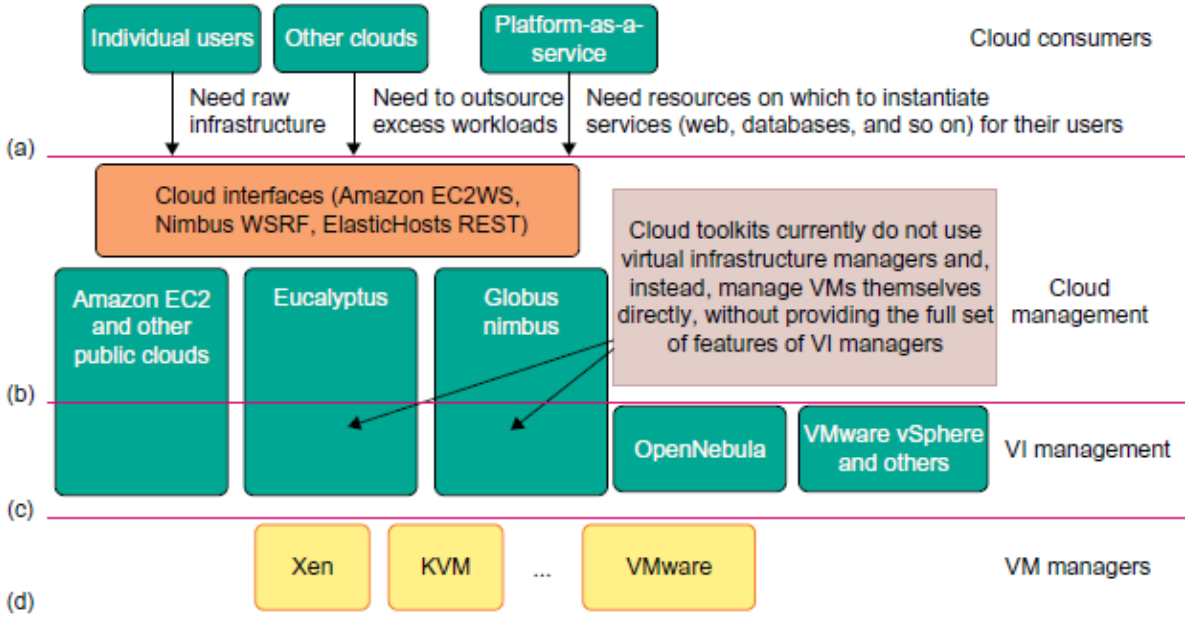


FIGURE 4.4

Cloud ecosystem for building private clouds: (a) Consumers demand a flexible platform; (b) Cloud manager provides virtualized resources over an IaaS platform; (c) VI manager allocates VMs; (d) VM managers handle VMs installed on servers.

Infrastructure-as-a-Service (IaaS):

- **Infrastructure-as-a-Service (IaaS): A model for different services is shown in Figure 4.5 [1], as shown below. The required service is performed by the rented cloud infrastructure. On this environment, the user can deploy and run his apps. Note that user doesn't have any control over the cloud infrastructure but can choose his OS, storage, apps and network components.**
- **Ex: Amazon EC2.**

Amazon EC2.

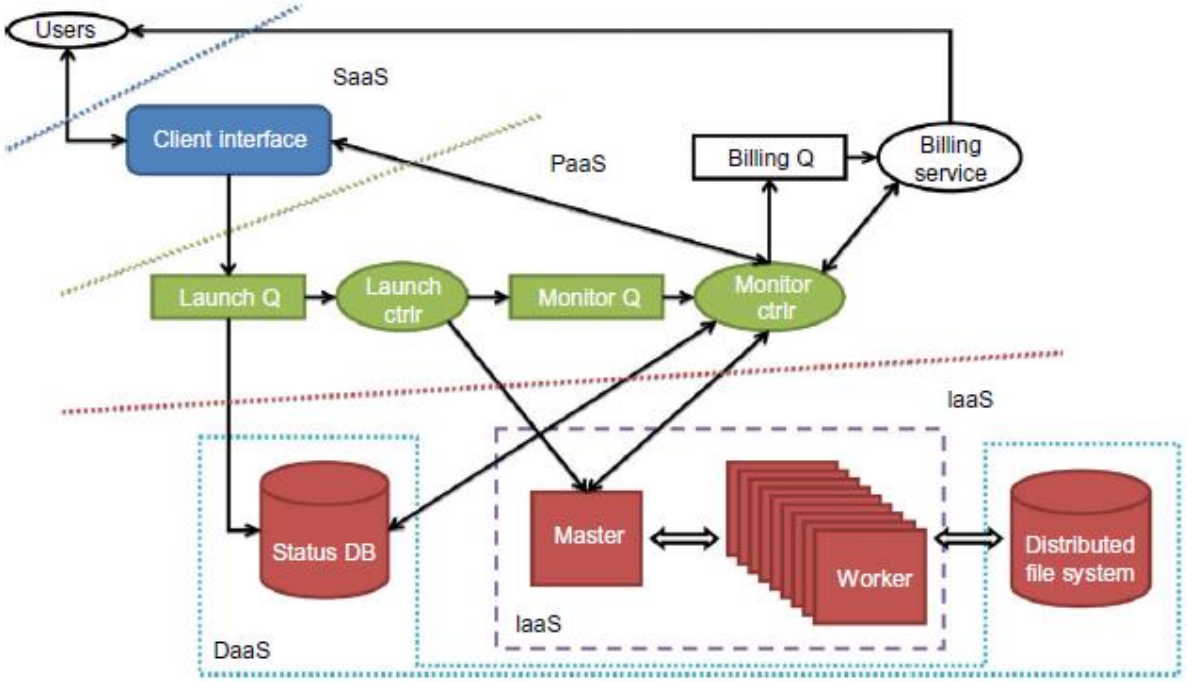


FIGURE 4.5

The IaaS, PaaS, and SaaS cloud service models at different service levels..

Platform-as-a-Service (PaaS)

- **Platform-as-a-Service (PaaS): To develop, deploy and manage apps with provisioned resources, an able platform is needed by the users. Such a platform includes OS and runtime library support. Different PaaS offered in the current market and other details are highlighted in the Table 4.2 [1] below:**

Amazon EC2.

Table 4.2 Five Public Cloud Offerings of PaaS [10,18]

Cloud Name	Languages and Developer Tools	Programming Models Supported by Provider	Target Applications and Storage Option
Google App Engine	Python, Java, and Eclipse-based IDE	MapReduce, web programming on demand	Web applications and BigTable storage
Salesforce.com's Force.com	Apex, Eclipse-based IDE, web-based Wizard	Workflow, Excel-like formula, Web programming on demand	Business applications such as CRM
Microsoft Azure	.NET, Azure tools for MS Visual Studio	Unrestricted model	Enterprise and web applications
Amazon Elastic MapReduce	Hive, Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, C++	MapReduce	Data processing and e-commerce
Aneka	.NET, stand-alone SDK	Threads, task, MapReduce	.NET enterprise applications, HPC

Software as a Service (SaaS):

- **Software as a Service (SaaS): This is about a browser-initiated app s/w over thousands of cloud customers. Services & tools offered by PaaS are utilized in construction and deployment of apps and management of their resources. The customer needs no investment and the provider can keep the costs low. Customer data is also stored in a cloud and is accessible through different other services. Ex: Gmail, Google docs, Salesforce.com etc.**

Enabling Technologies for Clouds

Table 4.3 Cloud-Enabling Technologies in Hardware, Software, and Networking

Technology	Requirements and Benefits
Fast platform deployment	Fast, efficient, and flexible deployment of cloud resources to provide dynamic computing environment to users
Virtual clusters on demand	Virtualized cluster of VMs provisioned to satisfy user demand and virtual cluster reconfigured as workload changes
Multitenant techniques	SaaS for distributing software to a large number of users for their simultaneous use and resource sharing if so desired
Massive data processing	Internet search and web services which often require massive data processing, especially to support personalized services
Web-scale communication	Support for e-commerce, distance education, telemedicine, social networking, digital government, and digital entertainment applications
Distributed storage	Large-scale storage of personal records and public archive information which demands distributed storage over the clouds
Licensing and billing services	License management and billing services which greatly benefit all types of cloud services in utility computing

Cloud Architecture

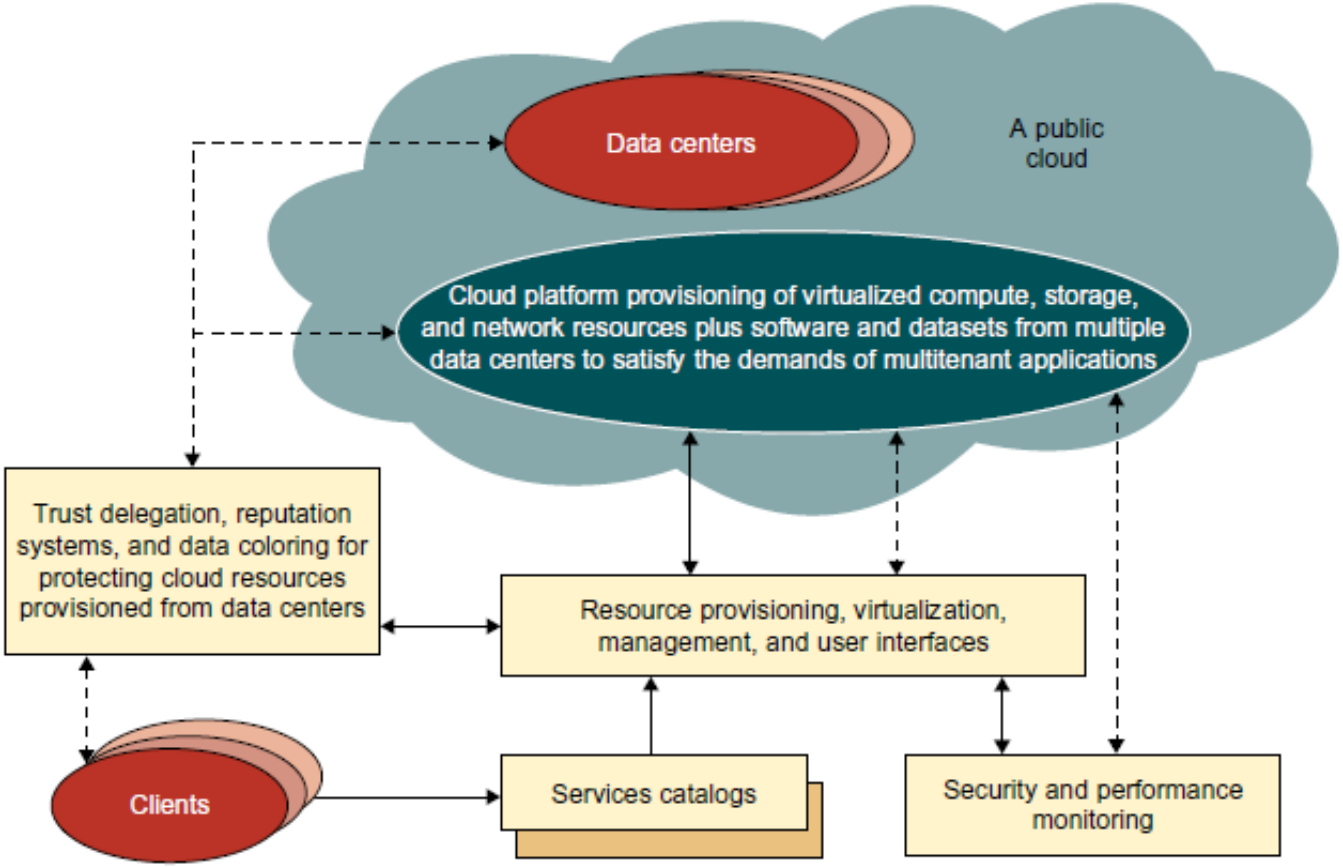


FIGURE 4.14

A security-aware cloud platform built with a virtual cluster of VMs, storage, and networking resources over the data-center servers operated by providers.

Cloud Architecture in Layers

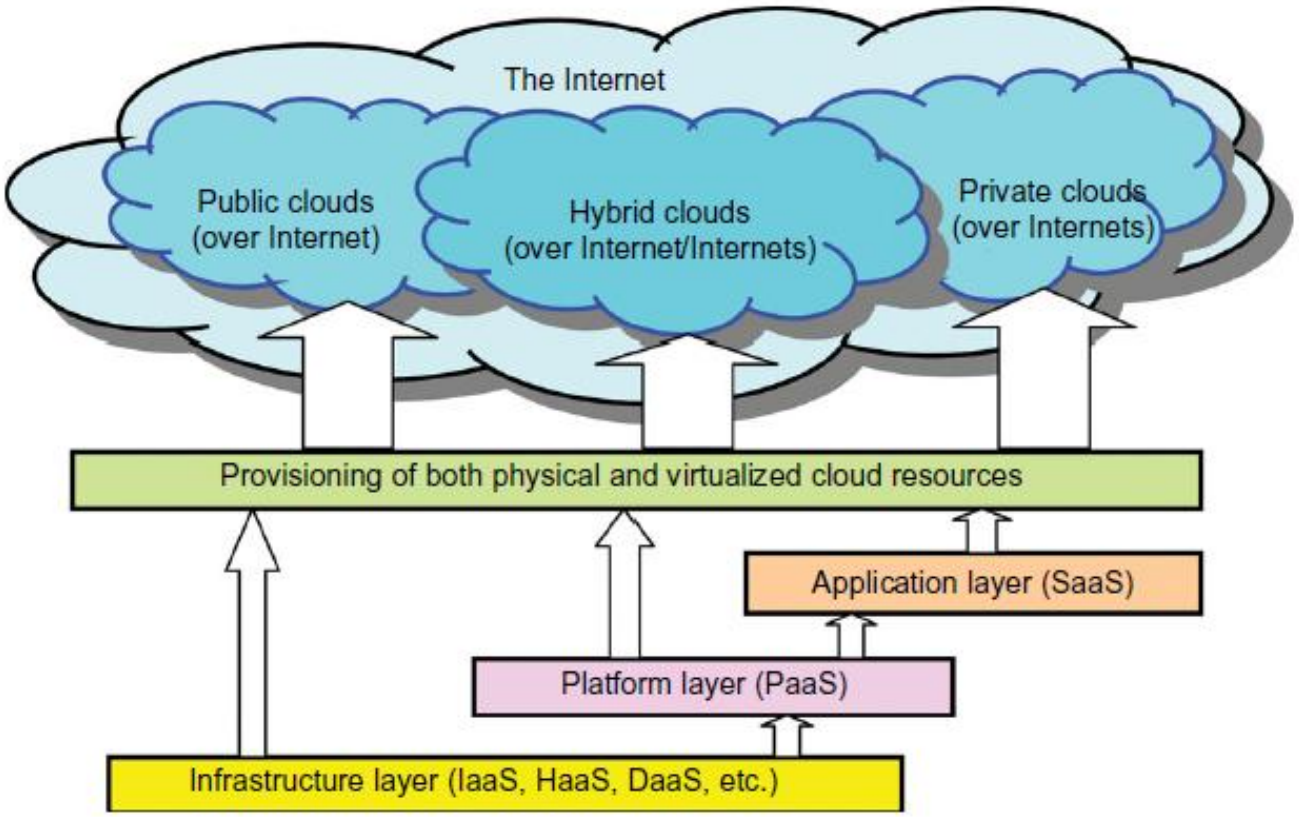


FIGURE 4.15

Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

Market-Oriented Cloud Architecture

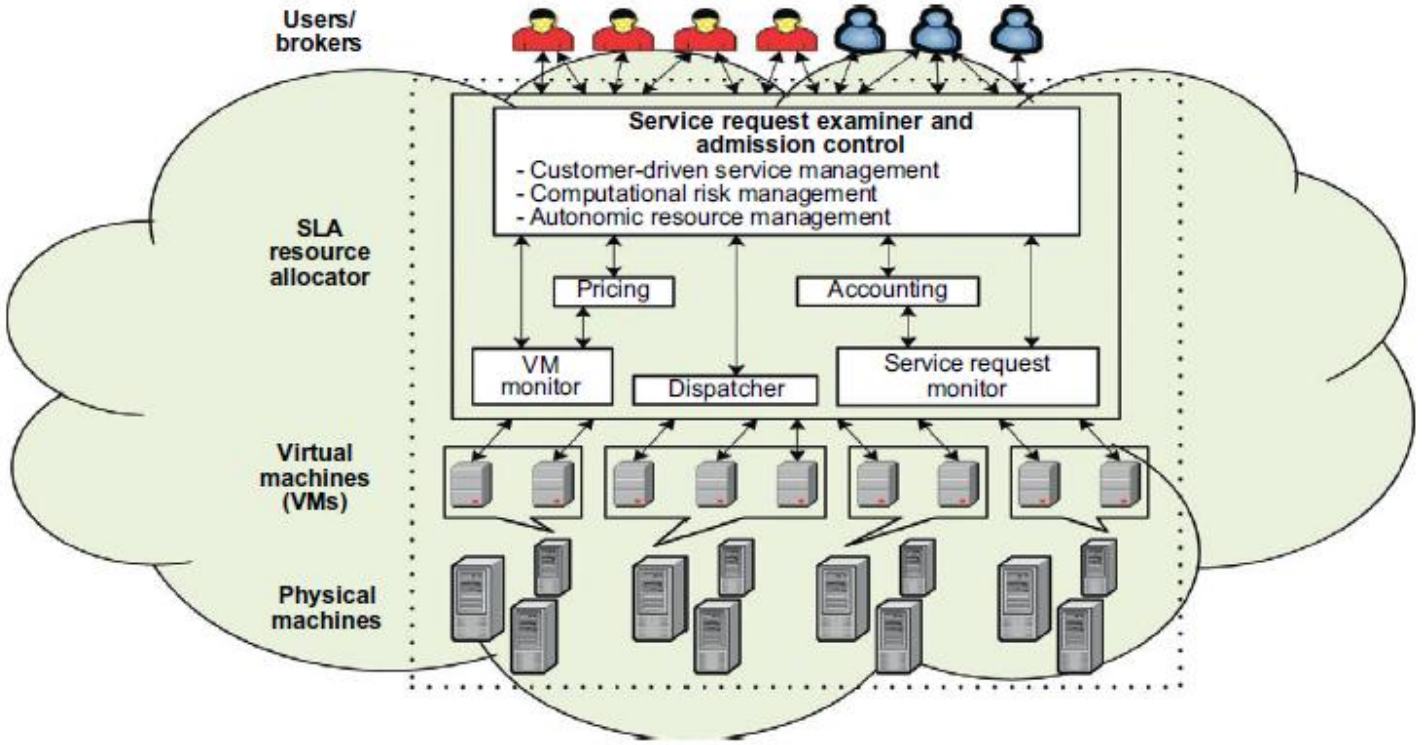


FIGURE 4.16

Market-oriented cloud architecture to expand/shrink leasing of resources with variation in QoS/demand from users.

QoS factors

- **QoS factors: In CC, different services being offered as commercial options in the market should take into account diverse factors for every service request: time, cost, reliability, and security. The QoS requirements can't be static and might from time to time on demand. Importance must be given to the customer, his requests and requirements – he is paying for all these. For achieving all these accomplishments in the CC market, the CRM steps into picture and plays a crucial role to satisfy each and every customer.**

Hardware VZ

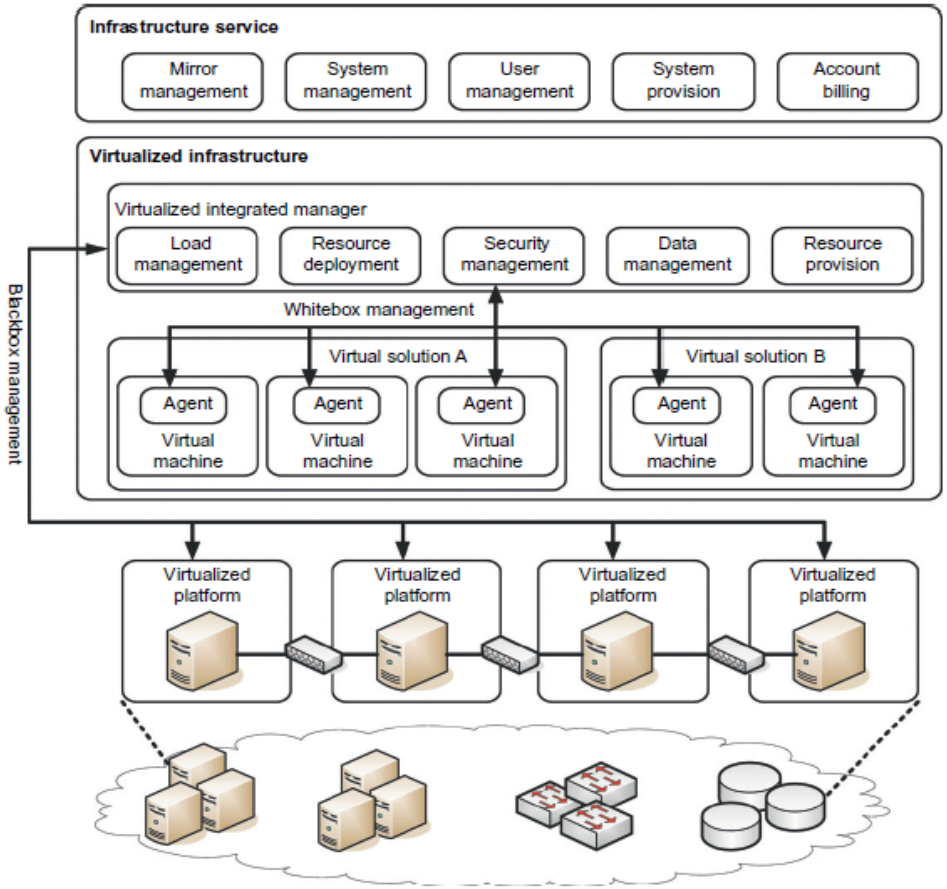


FIGURE 4.17

Virtualized servers, storage, and network for cloud platform construction.

(*Courtesy of Zhong-Yuan Qin, SouthEast University, China*)

Architectural Design Challenges

- Service Availability and Data Lock-in Problem
- Data Privacy and Security
- Unpredictable Performance and Bottlenecks
- Distributed Storage and Widespread Bugs
- Cloud Scalability, Interoperability and Standardization
- Software Licensing

Public Cloud Platforms

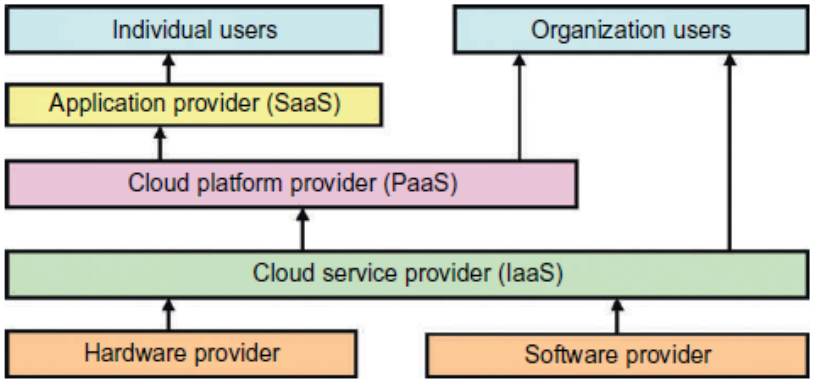


FIGURE 4.19

Roles of individual and organizational users and their interaction with cloud providers under various cloud service models.

Five Major service providers.

Table 4.5 Five Major Cloud Platforms and Their Service Offerings [36]

Model	IBM	Amazon	Google	Microsoft	Salesforce
PaaS	BlueCloud, WCA, RC2		App Engine (GAE)	Windows Azure	Force.com
IaaS	Ensembles	AWS		Windows Azure	
SaaS	Lotus Live		Gmail, Docs	.NET service, Dynamic CRM	Online CRM, Gifttag
Virtualization		OS and Xen	Application Container	OS level/ Hypel-V	
Service Offerings	SOA, B2, TSAM, RAD, Web 2.0	EC2, S3, SQS, SimpleDB	GFS, Chubby, BigTable, MapReduce	Live, SQL Hotmail	Apex, visual force, record security
Security Features	WebSphere2 and PowerVM tuned for protection	PKI, VPN, EBS to recover from failure	Chubby locks for security enforcement	Replicated data, rule-based access control	Admin./record security, uses metadata API
User Interfaces		EC2 command-line tools	Web-based admin. console	Windows Azure portal	
Web API	Yes	Yes	Yes	Yes	Yes
Programming Support	AMI		Python	.NET Framework	

Note: *WCA: WebSphere CloudBurst Appliance; RC2: Research Compute Cloud; RAD: Rational Application Developer; SOA: Service-Oriented Architecture; TSAM: Tivoli Service Automation Manager; EC2: Elastic Compute Cloud; S3: Simple Storage Service; SQS: Simple Queue Service; GAE: Google App Engine; AWS: Amazon Web Services; SQL: Structured Query Language; EBS: Elastic Block Store; CRM: Consumer Relationship Management.*

PKI=> Public Key Infrastructure; VPN=> Virtual Private Network

Google App Engine (GAE)

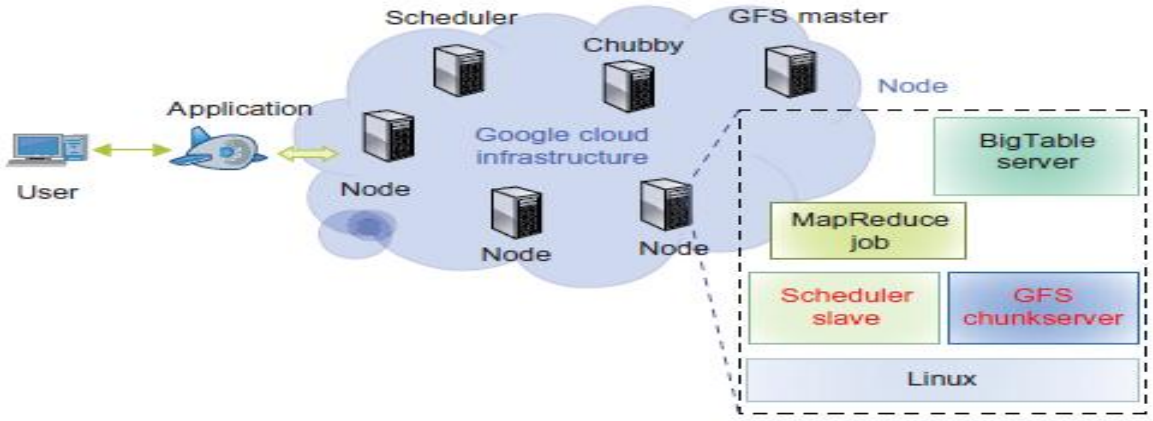


FIGURE 4.20

Google cloud platform and major building blocks, the blocks shown are large clusters of low-cost servers.

- Datastore offers OO, distributed and structured data storage services based on BigTable techniques. This secures data management operations.
- Application Runtime Environment: It is a platform for scalable web programming and execution. (Supports the languages of Java and Python)
- Software Development Kit: It is used for local app development and test runs of the new apps.
- Administration Console: Used for easy management of user app development cycles instead of physical resource management.
- Web Service Infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources.

Amazon Web Services (AWS)

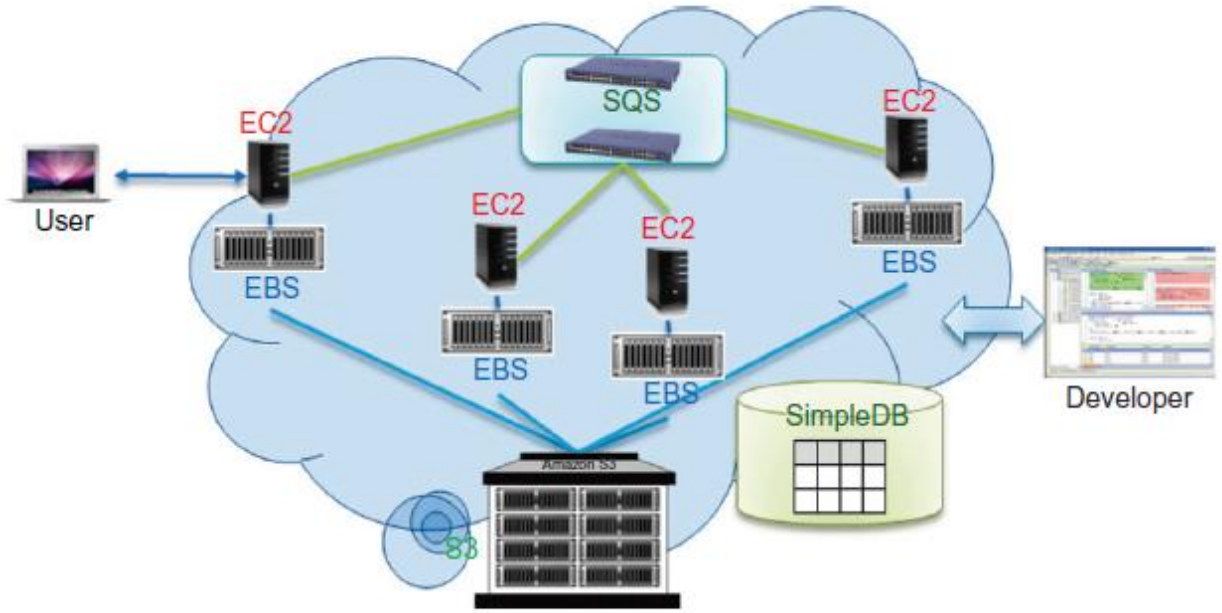


FIGURE 4.21

Amazon cloud computing infrastructure (Key services are identified here; many more are listed in Table 4.6).

EC2 provides the virtualized platforms to host the VMs where the cloud app can run.
S3 (Simple Storage Service) provides the OO storage service for the users.
EBS (Elastic Block Service) provides the block storage interface which can be used to support traditional apps.
SQS (Simple Queue Service) ensures a reliable message service between two processes.

Amazon Web Services (AWS)

Table 4.6 AWS Offerings in 2011

Service Area	Service Modules and Abbreviated Names
Compute	Elastic Compute Cloud (EC2), Elastic MapReduce, Auto Scaling
Messaging	Simple Queue Service (SQS), Simple Notification Service (SNS)
Storage	Simple Storage Service (S3), Elastic Block Storage (EBS), AWS Import/Export
Content Delivery	Amazon CloudFront
Monitoring	Amazon CloudWatch
Support	AWS Premium Support
Database	Amazon SimpleDB, Relational Database Service (RDS)
Networking	Virtual Private Cloud (VPC) (Example 4.1, Figure 4.6), Elastic Load Balancing
Web Traffic	Alexa Web Information Service, Alexa Web Sites
E-Commerce	Fulfillment Web Service (FWS)
Payments and Billing	Flexible Payments Service (FPS), Amazon DevPay
Workforce	Amazon Mechanical Turk

Amazon offers a RDS (relational database service) with a messaging interface. The AWS offerings are given below in Table 4.6 [1].

MS-Azure

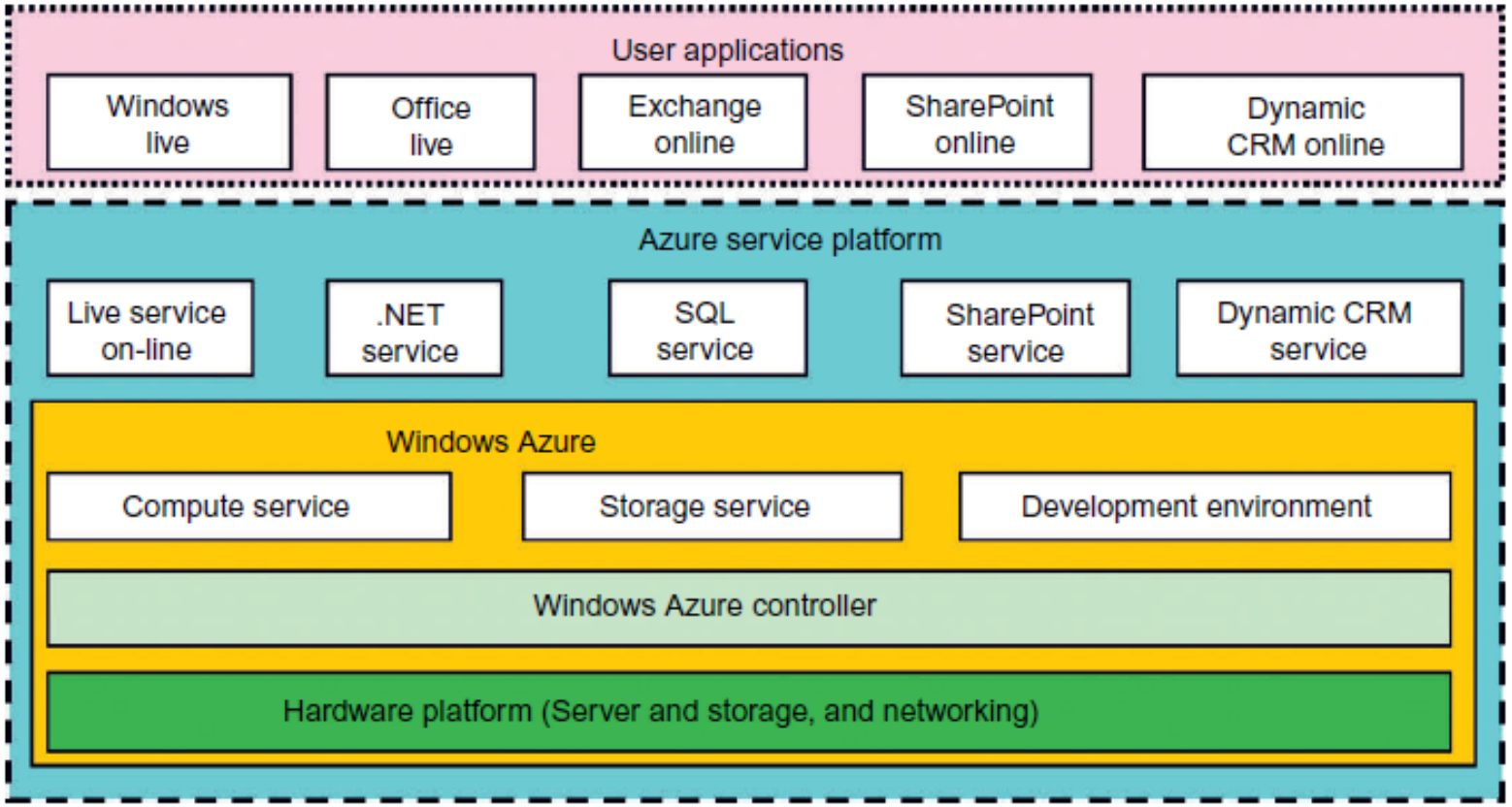


FIGURE 4.22

Microsoft Windows Azure platform for cloud computing.

MS-Azure

- Live Service: Through this, the users can apply MS live apps and data across multiple machines concurrently.
- .NET Service: This package supports app development on local hosts and execution on cloud machines.
- SQL Azure: Users can visit and utilized the relational database associated with a SQL server in the cloud.
- SharePoint Service: A scalable platform to develop special business apps.
- Dynamic CRM Service: This provides a business platform for the developers to manage the CRM apps in financing, marketing, sales and promotions.

UNIT-4
CLOUD PROGRAMMING AND SOFTWARE ENVIRONMENTS

Inter-Cloud Resource Management

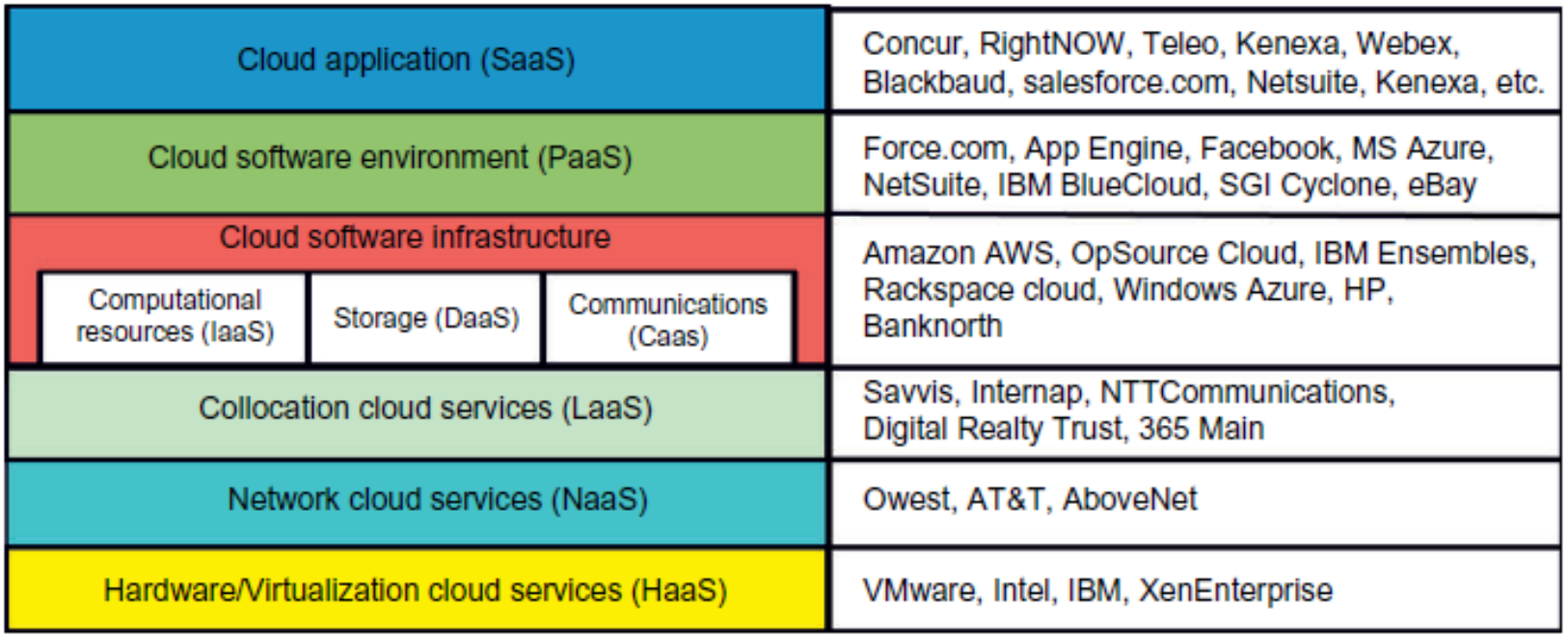


FIGURE 4.23

A stack of six layers of cloud services and their providers.

The top three service layers are SaaS, PaaS and IaaS. The bottom three layers are related to physical requirements and are as Hardware as a Service (HaaS), Network as a Service (NaaS), Location as a Service (LaaS), and Security as a Service (SaaS).

Inter-Cloud Resource Management

Table 4.7 Cloud Differences in Perspectives of Providers, Vendors, and Users

Cloud Players	IaaS	PaaS	SaaS
IT administrators/cloud providers	Monitor SLAs	Monitor SLAs and enable service platforms	Monitor SLAs and deploy software
Software developers (vendors)	To deploy and store data	Enabling platforms via configurators and APIs	Develop and deploy software
End users or business users	To deploy and store data	To develop and test web software	Use business software

Software Stack for CC

Software Stack for CC: A software stack [7] is a group of programs that work in tandem (in order) to produce a common goal. It may also refer to any set of apps that works in a specific order toward a common goal. Ex: Like a set in maths or a cluster in DM. The system has to be designed to meet goals like HT, HA, and fault tolerance. Physical or virtual servers can be used making the platform more flexible and be able to store and utilize large amount of data.

Resource Provisioning and Platform Deployme

Provisioning of Compute Resources (VMs): The provisioning of resources like CPU, memory, and bandwidth are distributed among the users as per the service level agreements (SLAs) signed before the start of the work. The problem here is the ever-changing levels of requests from the user, power management and conflicts in the SLAs.

Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures. Resource provisioning also demands fast discovery of services and data in the provided infrastructure. Ex: Efficient installation of VMs, live VM migration, and fast recovery from failures. Providers like Amazon, IBM and MS-Azure use VM templates, automation of provisioning and power-efficient schemes.

Resource Provisioning and Platform Deployme

Provisioning of Compute Resources (VMs): The provisioning of resources like CPU, memory, and bandwidth are distributed among the users as per the service level agreements (SLAs) signed before the start of the work. The problem here is the ever-changing levels of requests from the user, power management and conflicts in the SLAs.

Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures. Resource provisioning also demands fast discovery of services and data in the provided infrastructure. Ex: Efficient installation of VMs, live VM migration, and fast recovery from failures. Providers like Amazon, IBM and MS-Azure use VM templates, automation of provisioning and power-efficient schemes.

Resource Provisioning Methods

Demand-Driven Resource Provisioning: This method adds or removes computing instances based on the current utilization level for the allocated resources. This method automatically allocates two processors for the user app, if the user utilizes more than 60% of time for an extended period. That is, if the resource utilization has crossed a threshold of the concerned resource, extra resources will be allocated. This methodology is implemented by Amazon in EC2.

Event-Driven Resource Provisioning: This scheme adds or removes machine instances based on an event like festival season. At this time, the no. of users peaks and so does the traffic. This anticipation results in good QoS and customer satisfaction.

Resource Provisioning Methods

Popularity-Driven Resource Provisioning: In this method, The Internet searches for popularity of certain apps and creates extra instances if the popularity has risen.

Dynamic Resource Deployment: This can be implemented to achieve scalability in performance through efficient allocation of resources at every place in the grid as the situation demands. To achieve this, we need an **inter-grid gateway** (IGG) between different grids that allocates the resources from a local cluster to deploy apps by **requesting the VMs**, **enacting** (endorse) **the leases**, and **deploying the VMs as per requests**.

The Inter-Grid provides and allocates a **distributed virtual environment (DVE)**. It is a virtual cluster of VMs that runs in isolation from other virtual clusters. This process is carried out by a component called DVE manager. Received massages are handled in parallel in a thread pool.

Resource Provisioning Methods

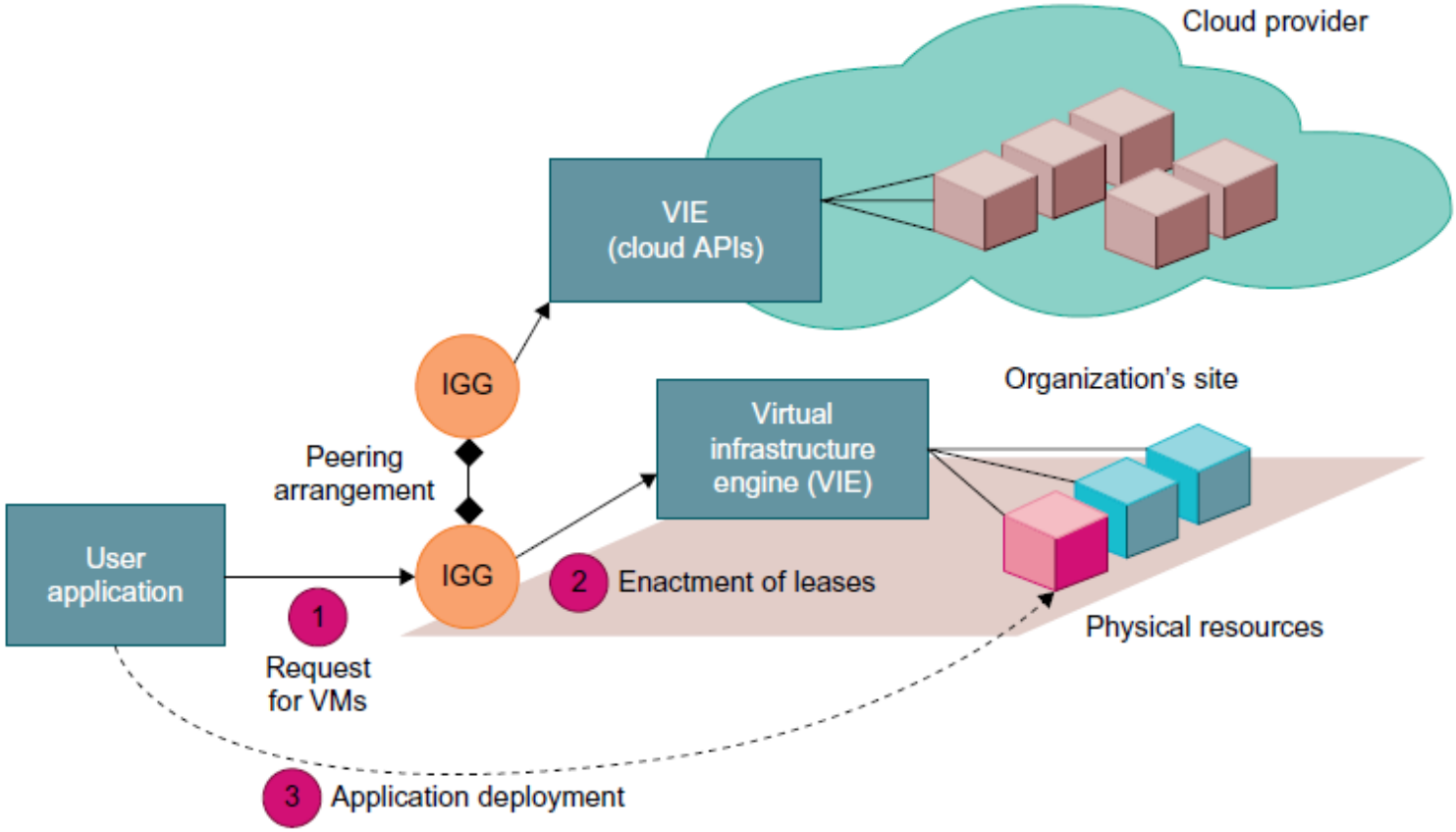


FIGURE 4.26

Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

Provisioning of Storage Resources

Provisioning of Storage Resources: The data in CC is stored in the clusters of the cloud provider and can be accessed anywhere in the world. Ex: email. For data storage, distributed file system, tree structure file system, and others can be used. Ex: GFS, HDFS, MS-Cosmos. This method provides a convenient coding platform for the developers. The storage methodologies and their features can be found in Table 4.8 [1].

Provisioning of Storage Resources

Table 4.8 Storage Services in Three Cloud Computing Systems

Storage System	Features
GFS: Google File System	Very large sustainable reading and writing bandwidth, mostly continuous accessing instead of random accessing. The programming interface is similar to that of the POSIX file system accessing interface.
HDFS: Hadoop Distributed File System	The open source clone of GFS. Written in Java. The programming interfaces are similar to POSIX but not identical.
Amazon S3 and EBS	S3 is used for retrieving and storing data from/to remote servers. EBS is built on top of S3 for using virtual disks in running EC2 instances.

Virtual Machine Creation & Management

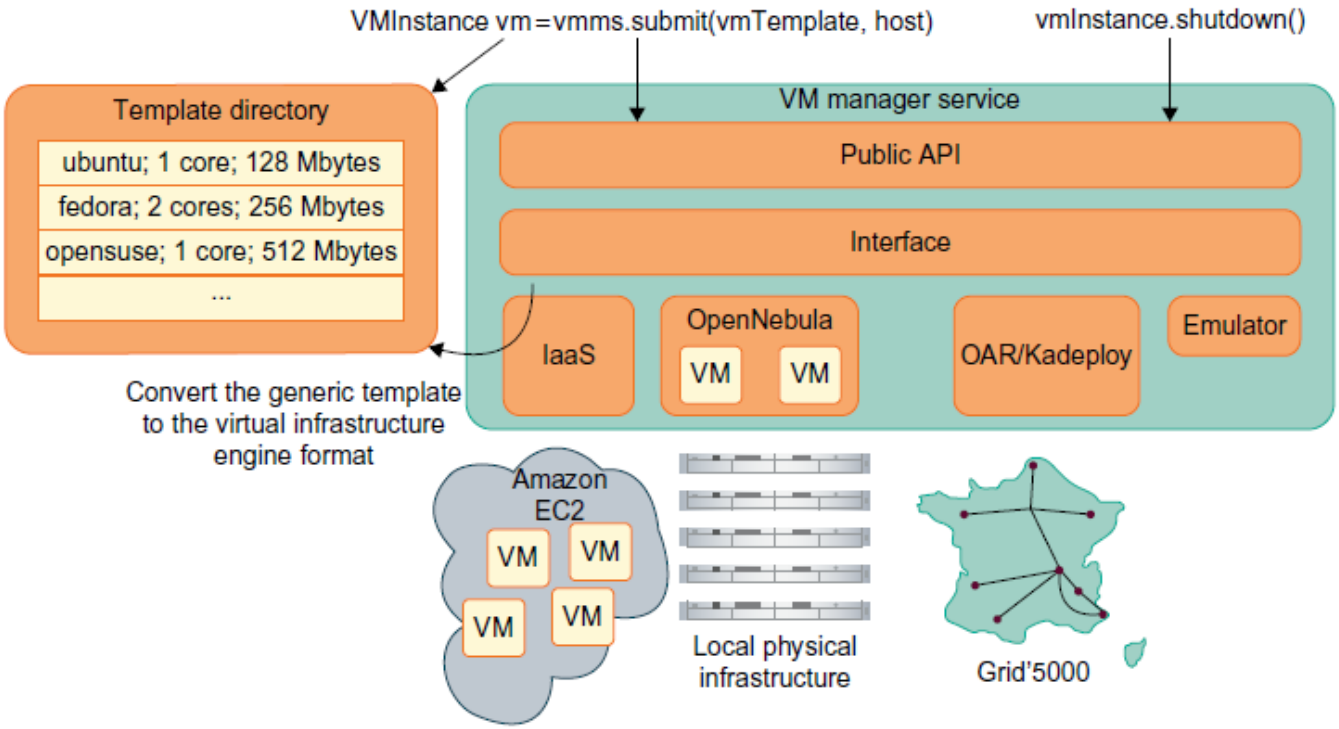


FIGURE 4.27

Interactions among VM managers for cloud creation and management; the manager provides a public API for users to submit and control the VMs.

Virtual Machine Creation & Management

- **Independent Service Management**: By using independent service providers, the cloud apps can run different services at the same time. Some other services are used for providing data other than the compute or storage services.
- **Running Third Party Apps**: IN this case, the cloud platforms have to provide support for apps constructed by third-party app providers. The concerned APIs are in the form of services provided by another company. (Ex: Dropbox + Gmail + User).
- **VM Manager**: It is a link between the gateway and resources. The physical resources aren't shared directly, but in a virtualized method. The VMs themselves become the actual resources. Ex: OpenNebula (an OS). Users submit VMs on physical machines using hypervisors, which enables the running of several operating systems on the same host concurrently.

Virtual Machine Creation & Management

VM Templates: A VM template is analogous (similar) to the configuration of a computer and contains the description for a VM. Information provided is:

The no. of processors allocated to the VM
Memory required by a VM
The kernel used by the VM's OS
The disk image containing the VM's file system
The price per hour
The gateway administrator provides the VM template information and can add, update and delete the templates at any time. Before starting an instance, scheduler gives the network configuration and address of the host. The MAC and IP addresses are also allocated. It also contains the path to the disk image storage.

Virtual Machine Creation & Management

Distributed VM Management: A distributed VM manager requests for VMs and gets their status and obtains a list containing the IP addresses of the VMs with secure shell (SSH) tunnels. The managers also obtains the template to be used by the VM, schedules the task for the VM, sets up the tunnel, and executes the tasks for each of the VM.

Cloud Security and Trust Management

Cloud Security and Trust Management: Lacking of trust between service providers and clients has been a major problem in the field and much more since the advent of ecommerce. Cloud platforms are a concern for some users for lack of privacy protection, security assurance, and so on. All these can be solved with a technical approach.

Security measures at various levels:

Basic Cloud Security: The basic cloud security enforcements are: security measures in data centers (like biometric readers, CCTV, man-traps etc.), fault-tolerant firewalls, IDS Intrusion Detection System), data encryption, strict password policies, and so on. The Figure 4.31 [1] shows the security measures at various levels:

Virtual Machine Creation & Management

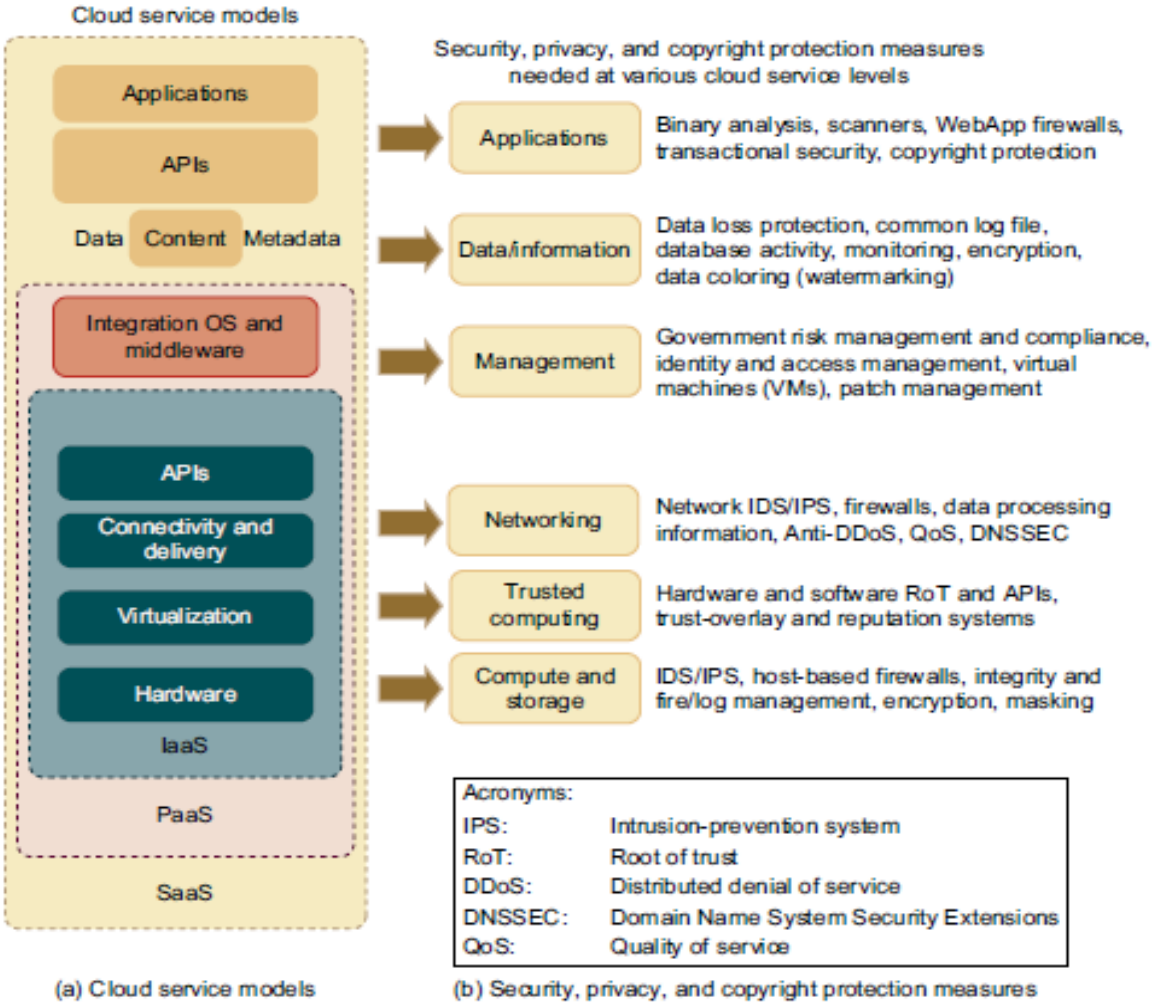


FIGURE 4.31

Cloud service models on the left (a) and corresponding security measures on the right (b); the IaaS is at the innermost level, PaaS is at the middle level, and SaaS is at the outermost level, including all hardware, software, datasets, and networking resources.

Cloud Defence Methods

Cloud Defence Methods: Virtualization enhances cloud security, but VMs add an additional layer of software that might lead to a single point of failure. So the VMs should be isolated in their deployment and work – the failure of one VM will not affect another. The Table 4.9 [1] below lists the protection schemes to secure public clouds and data centers.

Protection schemes to secure public clouds ar data centers.

Table 4.9 Physical and Cyber Security Protection at Cloud/Data Centers

Protection Schemes	Brief Description and Deployment Suggestions
Secure data centers and computer buildings	Choose hazard-free location, enforce building safety. Avoid windows, keep buffer zone around the site, bomb detection, camera surveillance, earthquake-proof, etc.
Use redundant utilities at multiple sites	Multiple power and supplies, alternate network connections, multiple databases at separate sites, data consistency, data watermarking, user authentication, etc.
Trust delegation and negotiation	Cross certificates to delegate trust across PKI domains for various data centers, trust negotiation among certificate authorities (CAs) to resolve policy conflicts
Worm containment and DDoS defense	Internet worm containment and distributed defense against DDoS attacks to secure all data centers and cloud platforms
Reputation system for data centers	Reputation system could be built with P2P technology; one can build a hierarchy of reputation systems from data centers to distributed file systems
Fine-grained file access control	Fine-grained access control at the file or object level; this adds to security protection beyond firewalls and IDSes
Copyright protection and piracy prevention	Piracy prevention achieved with peer collusion prevention, filtering of poisoned content, nondestructive read, alteration detection, etc.
Privacy protection	Uses double authentication, biometric identification, intrusion detection and disaster recovery, privacy enforcement by data watermarking, data classification, etc.

Defence against DDoS Flooding attacks

Defence against DDoS Flooding attacks: A DDoS defence system must be designed to cover multiple network domains in a cloud platform. The DDoS causes an abnormal surge in the network traffic by a hidden attacker which leads of the crash of the service/website or disk exhaustion or connection saturation.

Data and Software Protection Techniques:

Data and Software Protection Techniques:

Data Integrity and Privacy Protection

Data Colouring and Cloud Watermarking

Data Lock-in Problems and Solutions: Once the data is moved into the cloud, users cannot easily extract their data and programs from the cloud servers to run on another platform. This is known as data lock-in. The solution possible here is to build platform-independent APIs where migration from one platform to another is easier.

Service-Oriented Architecture

Service-Oriented Architecture: SOA is concerned about how to design a software system that makes use of services or apps through their interfaces. These apps are distributed over the networks. The World Wide Web Consortium (W3C) defines SOA as a form of distributed architecture characterized by:

Logical View: The SOA is an abstracted, logical view of actual programs, DBs etc. defined in terms of the operations it carries out. The service is formally defined in terms of messages exchanged between providers and requests.

Message Orientation

Description Orientation

Services and Web Services

Services and Web Services: In an SOA concept, the s/w capabilities are delivered & consumed through loosely coupled and reusable services using messages. 'Web Service' is a self-contained modular application designed to be used by other apps across the web. This can be seen in Figure 5.2 [1].

WSDL => Web Services Description Language
UDDI => Universal Description, Discovery and Integration
SOAP => Simple Object Access Protocol

Services and Web Services

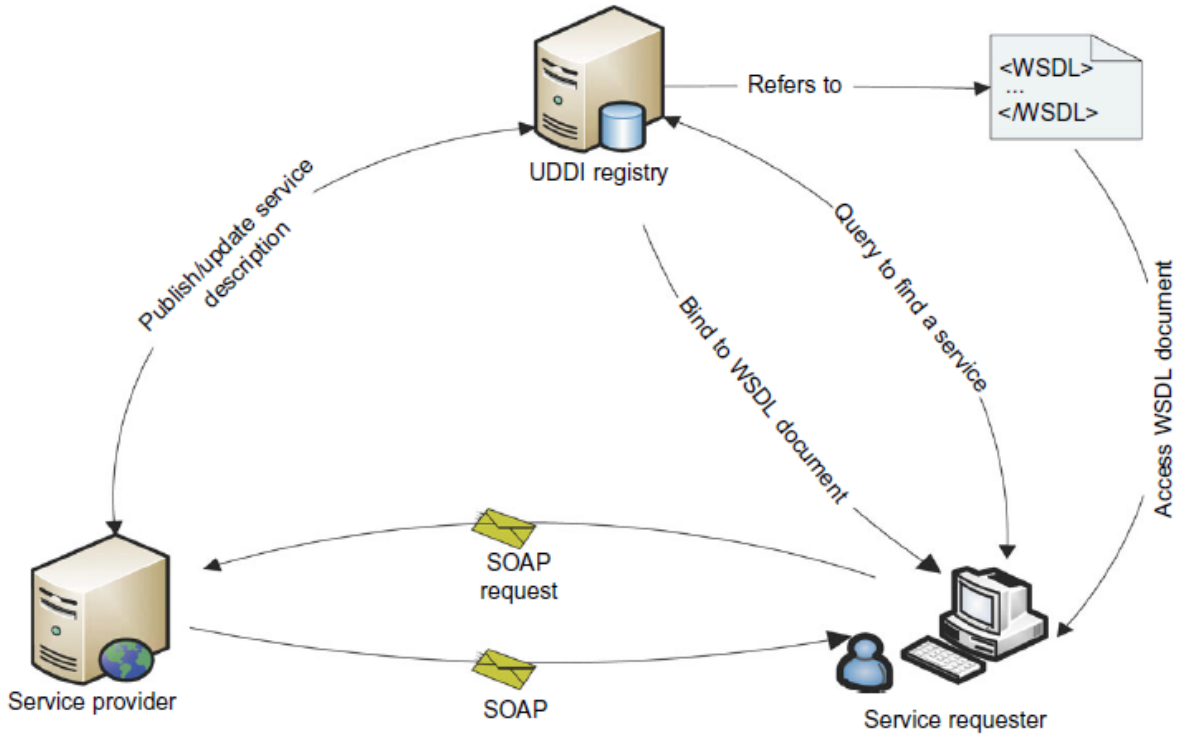


FIGURE 5.2

A simple web service interaction among provider, user, and the UDDI registry.

Services and Web Services

- **SOAP**: This provides a standard packaging structure for transmission of XML documents over various IPs. (HTTP, SMTP, FTP). A SOAP message consists of an **envelope** (root element), which itself contains a **header**. It also had a **body** that carries the payload of the message.
- **WSDL**: It describes the interface and a set of operations supported by a web service in a standard format.
- **UDDI**: This provides a global registry for advertising and discovery of web services by searching for names, identifiers, categories.

Since SOAP can combine the strengths of XML and HTTP, it is useful for heterogeneous distributed computing environments like grids and clouds

Enterprise Multitier Architecture

- **Enterprise Multitier Architecture**: This is a kind of client/server architecture application processing and data management are logically separate processes. As seen below in Figure 5.4 [1], it is a three-tier information system where each layer has its own important responsibilities.
- **Presentation Layer**: Presents information to external entities and allows them to interact with the system by submitting operations and getting responses.
- **Application Logic (Middleware)**: These consist of programs that implement actual operations requested by the client. The middle tier can also be used for user authentication and granting of resources, thus removing some load from the servers.
- **Resource Management Layer (Data Layer):** It deals with the data sources of an information system.

Enterprise Multitier Architecture

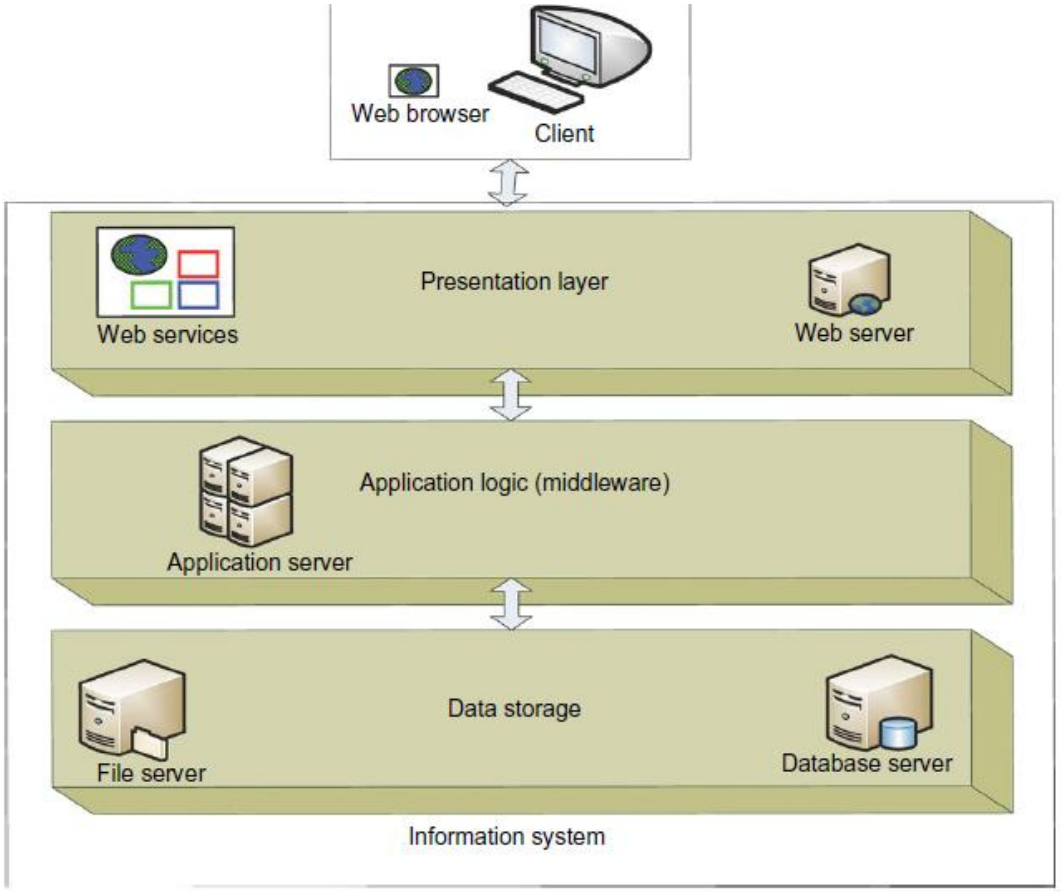


FIGURE 5.4

Three-tier system architecture.

Open Grid Services Architecture

OGSA Grid: Open Grid Services Architecture is intended to

- Facilitate the usage of resources across heterogeneous environments
- Deliver best QoS
- Define open interfaces between diverse resources
- Develop inter-operable standards

OGSA architecture falls into seven broad areas, as shown in Figure 5.5 [1].

Infrastructure Services, Execution Management Services, Data Management Services, Resource Management Services, Security Services, Security Services, Information Services and Self-management Services (automation).

Open Grid Services Architecture

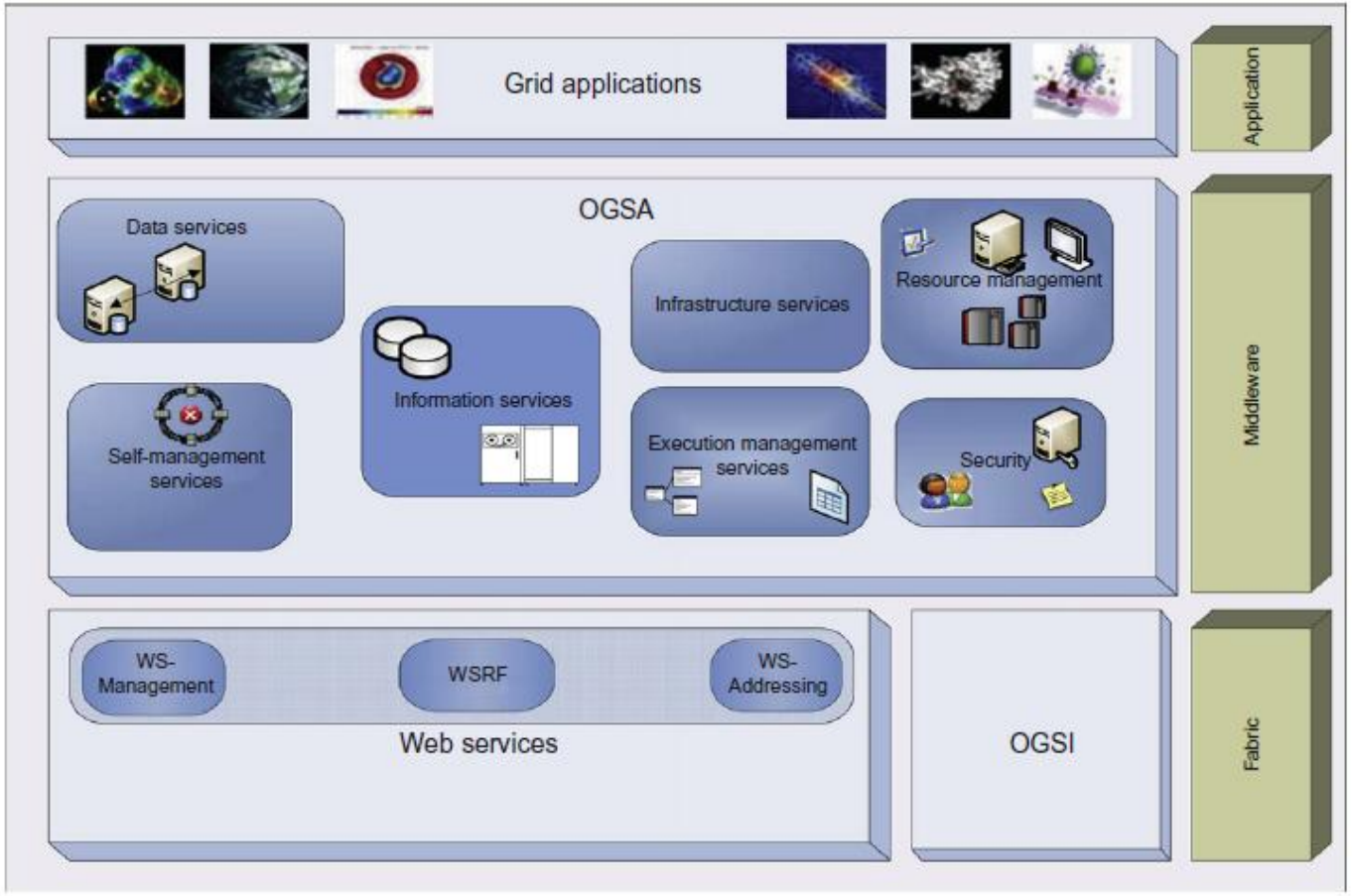
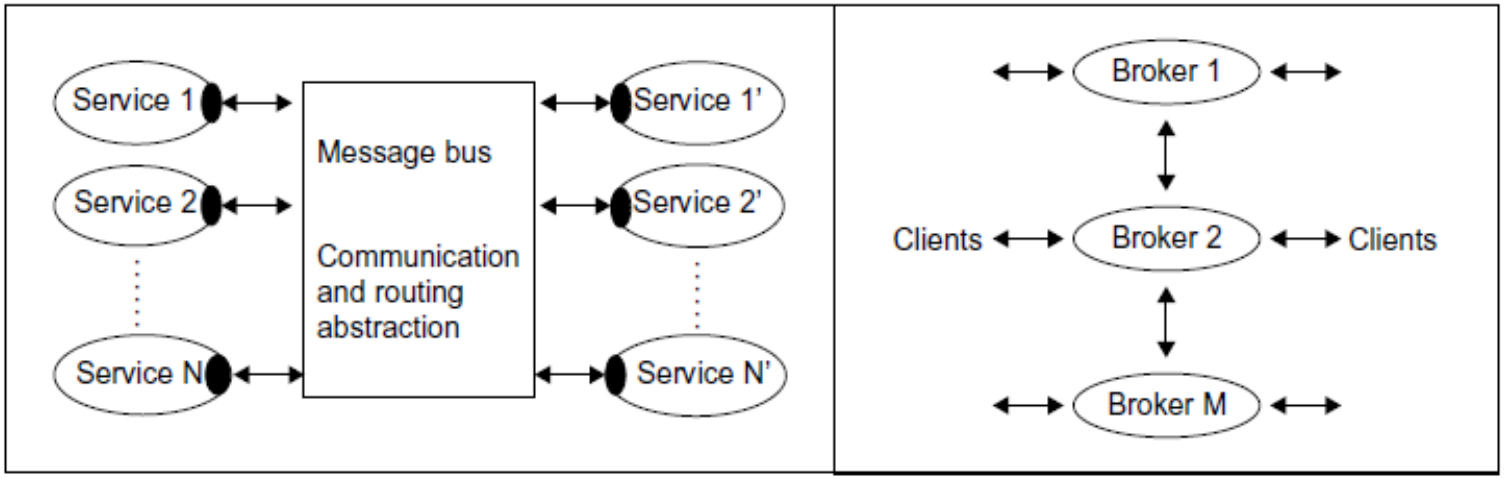


FIGURE 5.5

The OGSA architecture.

Message-Oriented Middleware



(a) Implemented between services

(b) As a network of distributed brokers

FIGURE 5.6

Two message bus implementations between services or using a broker network.

Enterprise Service Bus (ESB) refers to the case where the bus supports the integration of many components in different styles as shown above. No source and destination channel is opened but only messages are induced from different services. A message bus is shown linking the services by receiving and sending messages but this methodology can work with any software or hardware.

Queuing and Message Systems

Queuing and Message Systems: The best known standard in this field is the Java Message Service (JMS) which specifies a set of interfaces utilized in communication queuing systems. Advanced Message Queuing Protocol (AMQP) specifies a set of wire formats for communications.

UNIT V
CLOUD RESOURCE MANAGEMENT AND SCHEDULING

Virtual Clusters and Resource Management

- **Virtual Clusters and Resource Management: A physical cluster is a collection of physical servers that are interconnected. The issues that are to be dealt with here are: live migration of VMs, memory and file migrations and dynamic deployment of virtual clusters.**
- **When a general VM is initialized, the administrator has to manually write configuration information; this increases his workload, particularly when more and more VMs join the clusters. As a solution to this, a service is needed that takes care of the configuration information (capacity, speed etc.) of the VMs. The best example is Amazon's Elastic Compute Cloud (EC2), which provides elastic computing power in a cloud.**

Virtual Clusters and Resource Management

- **Most VZ platforms like VMware ESX Server, and XenServer support a bridging mode which allows all domains to appear on the network as individual hosts. Through this mode, VMs can communicate with each other freely through the virtual network and configure automatically.**

Physical versus Virtual Clusters

- **Physical versus Virtual Clusters: Virtual Clusters are built with VMs installed at one or more physical clusters. The VMs in a virtual cluster are interconnected by a virtual network across several physical networks. The concept can be observed in Figure 3.18 [1].**

Physical versus Virtual Clusters

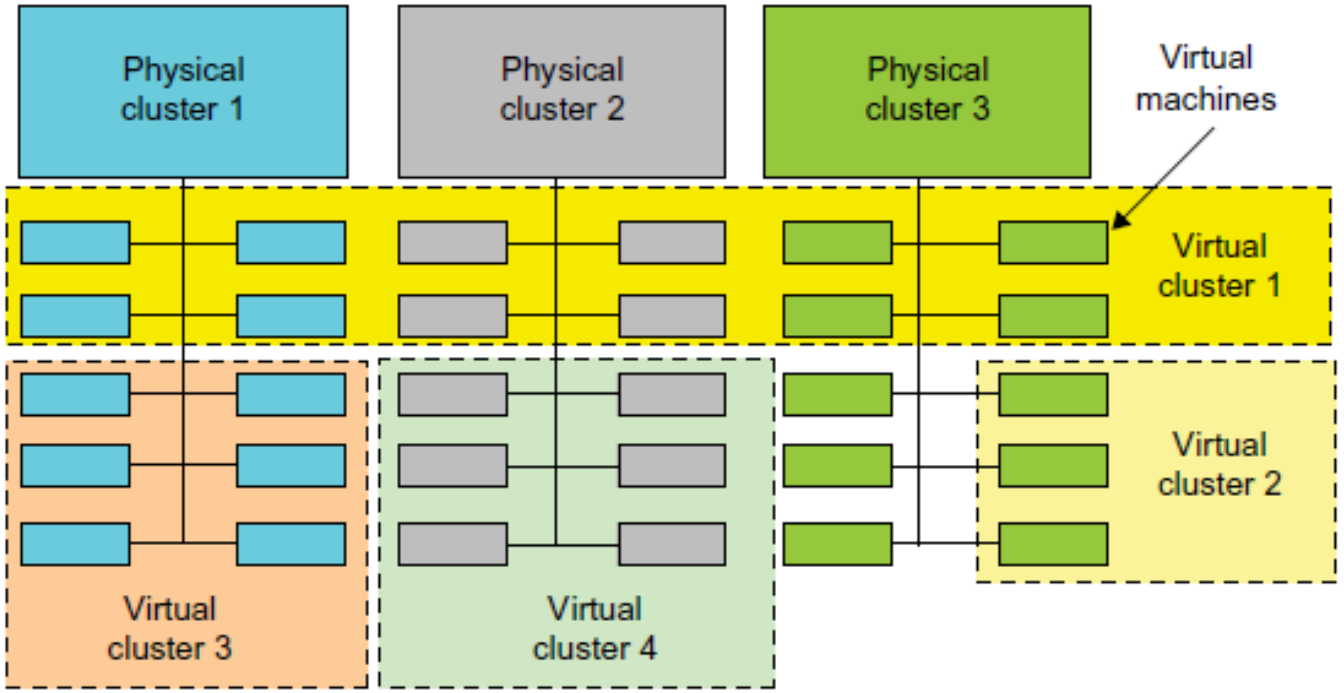


FIGURE 3.18

A cloud platform with four virtual clusters over three physical clusters shaded differently.

Physical versus Virtual Clusters

- The provisioning of VMs to a virtual cluster is done dynamically to have the following properties:
- The virtual cluster nodes can be either physical or virtual (VMs) with different operating systems.
- A VM runs with a guest OS that manages the resources in the physical machine.
- The purpose of using VMs is to consolidate multiple functionalities on the same server.
- VMs can be replicated in multiple servers to promote parallelism, fault tolerance and disaster discovery.
- The no. of nodes in a virtual cluster can grow or shrink dynamically.
- The failure of some physical nodes will slow the work but the failure of VMs will cause no harm (fault tolerance is high).

Physical versus Virtual Clusters

- **NOTE: Since system virtualization is widely used, the VMs on virtual clusters have to be effectively managed. The virtual computing environment should provide high performance in virtual cluster deployment, monitoring large clusters, scheduling of the resources, fault tolerance and so on.**

Virtual Cluster based on application partitioning

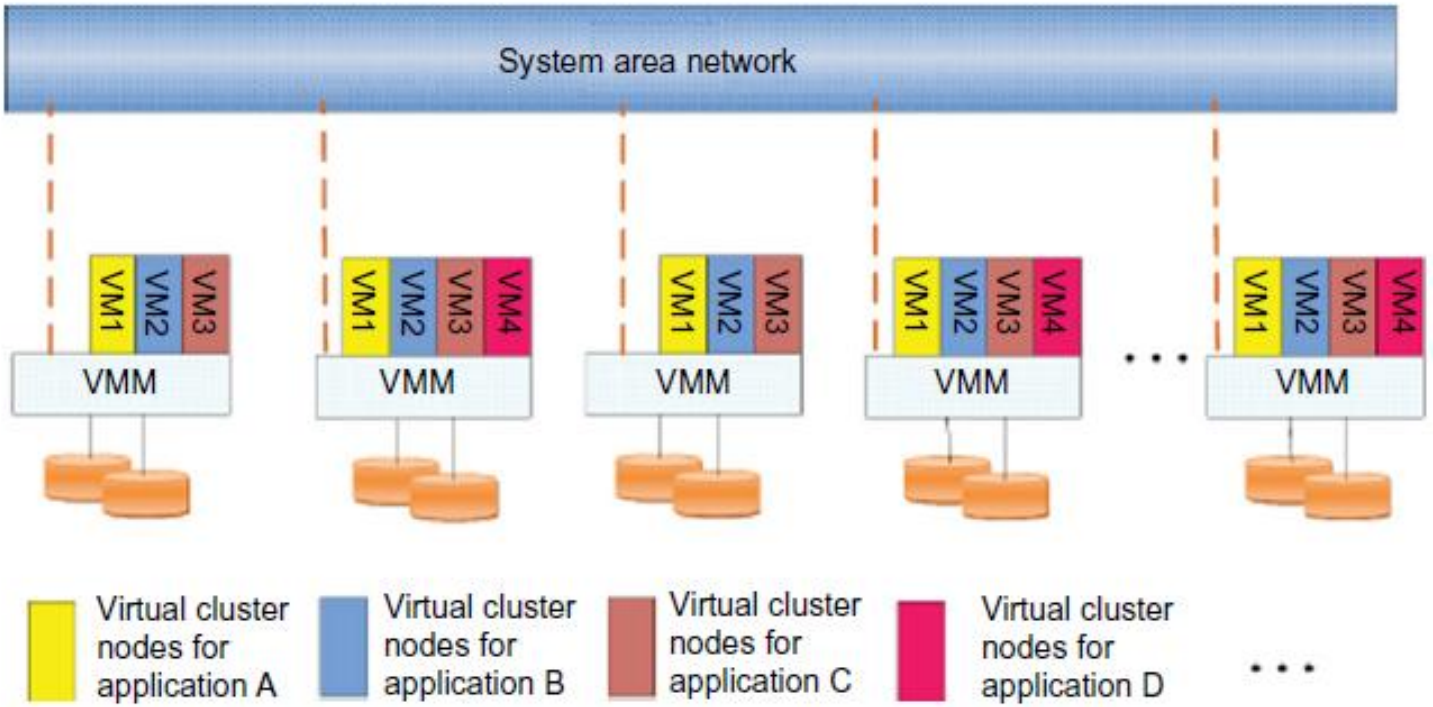


FIGURE 3.19

The concept of a virtual cluster based on application partitioning.

Physical versus Virtual Clusters

- **Figure 3.19 [1] shows the concept of a virtual cluster based on app partitioning. The different colours represent nodes in different virtual clusters. The storage images (SSI) from different VMs from different clusters is the most important concept here. Software packages can be pre-installed as templates and the users can build their own software stacks. Note that the boundary of the virtual cluster might change since VM nodes are added, removed, or migrated dynamically.**

Fast Deployment and Effective Scheduling

- **Fast Deployment and Effective Scheduling: The concerned system should be able to**
- **Construct and distribute software stacks (OS, libraries, apps) to a physical node inside the cluster as fast as possible**
- **Quickly switch runtime environments from one virtual cluster to another.**
- **NOTE: Green Computing: It is a methodology that is environmentally responsible and an eco-friendly usage of computers and their resources. It is also defined as the study of designing, manufacturing, using and disposing of computing devices in a way that reduces their environmental impact.**

Fast Deployment and Effective Scheduling

- **Engineers must concentrate upon the point the available resources are utilized in a cost and energy-reducing manner to optimize the performance and throughput. Parallelism must be put in place wherever needed and virtual machines/clusters should be used for attaining this goal. Through this, we can reduce the overhead, attain load balancing and achieve scale-up and scale-down mechanisms on the virtual clusters. Finally, the virtual clusters must be clustered among themselves again by mapping methods in a dynamical manner.**

High Performance Virtual Storage

- High Performance Virtual Storage: A template must be prepared for the VM construction and usage and distributed to the physical hosts. Software packages that reduce the time for customization (getting used to) and switching of environment. Users should be identified by their profiles that are stored in data blocks. All these methods increase the performance in virtual storage. Ex: Dropbox
- Steps to deploy (arrange/install) a group of VMs onto a target cluster:
- Preparing the disk image (SSI)
- Configuring the virtual machines
- Choosing the destination nodes
- Executing the VM deployment commands at every host

High Performance Virtual Storage

- **NOTE: A template is a disk image/SSI that hides the distributed environment from the user. It may consist of an OS and some apps. Templates are chosen by the users as per their requirements and can implement COW (Copy on Write) format. A new COW backup file is small and easy to create and transfer, thus reducing space consumption.**
- **It should be noted that every VM is configured with a name, disk image, network settings, and is allocated a CPU and memory. But this might be cumbersome if the VMs are many in number. The process can be simplified by configuring similar VMs with pre-edited profiles. Finally, the deployment principle should be able to fulfil the VM requirement to balance the workloads.**

Live VM Migration Steps

- **Live VM Migration Steps: Normally in a cluster built with mixed modes of host and guest systems, the procedure is to run everything on the physical machine. When a VM fails, it can be replaced by another VM on a different node, as long as they both run the same guest OS. This is called a failover (a procedure by which a system automatically transfers control to a duplicate system when it detects a fault or failure) of a physical system to a VM. Compared to a physical-physical failover, this methodology has more flexibility. It also has a drawback – a VM must stop working if its host node fails. This can be lessened by migrating from one node to another for a similar VM. The live migration process is depicted in Figure 3.20 [1].**

Managing a Virtual Cluster

- Managing a Virtual Cluster: There exist four ways.
- We can use a guest-based manager, by which the cluster manager resides inside a guest OS. Ex: A Linux cluster can run different guest operating systems on top of the Xen hypervisor.
- We can bring out a host-based manager which itself is a cluster manager on the host systems. Ex: VMware HA (High Availability) system that can restart a guest system after failure.
- An independent cluster manager, which can be used on both the host and the guest – making the infrastructure complex.
- Finally, we might also use an integrated cluster (manager), on the guest and host operating systems; here the manager must clearly distinguish between physical and virtual resources.
- NOTE: The virtual cluster management schemes are greatly enhanced if the VM life migration is enabled with minimum overhead.

Managing a Virtual Cluster

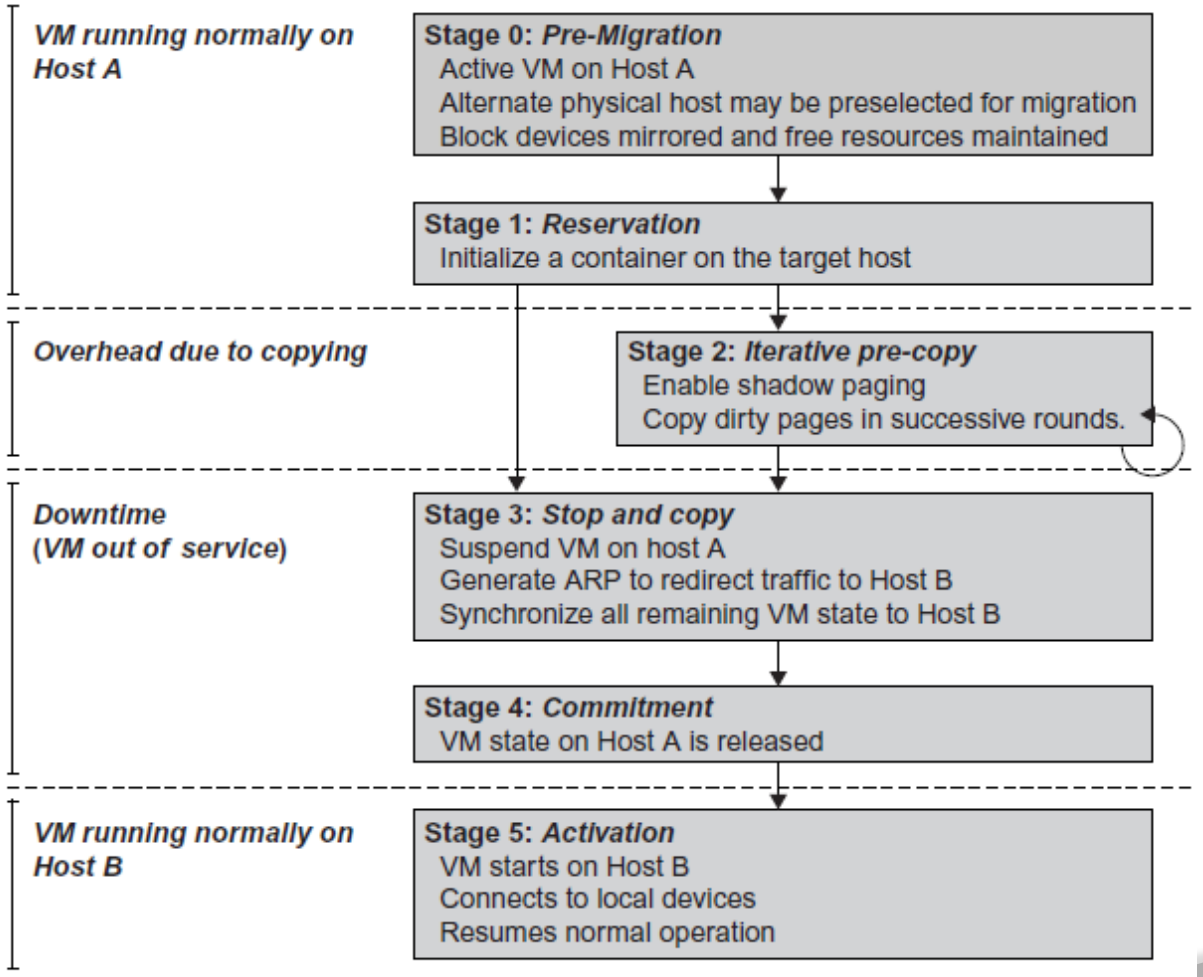


FIGURE 3.20

Live migration process of a VM from one host to another.

Managing a Virtual Cluster

- **Virtual clusters are generally used where fault tolerance of VMs on the host plays an important role in the total cluster strategy. These clusters can be applied in grids, clouds and HPC platforms. The HPC is obtained by dynamical finding and usage of resources as per requirement, and less migration time & bandwidth that is used.**

VM States

- A VM can be in one of the following states:
- Inactive State: This is defined by the VZ platform, under which the VM is not enabled.
- Active State: This refers to a VM that has been instantiated at the VZ platform to perform a task.
- Paused State: A VM has been instantiated but disabled temporarily to process a task or is in a waiting state itself.
- Suspended State: A VM enters this state if its machine file and virtual resources are stored back to the disk.

Live Migration Steps

- Live Migration Steps: This consists of 6 steps.
- Steps 0 and 1: Start migration automatically and checkout load balances and server consolidation.
- Step 2: Transfer memory (transfer the memory data + recopy any data that is changed during the process). This goes on iteratively till changed memory is small enough to be handled directly.
- Step 3: Suspend the VM and copy the last portion of the data.
- Steps 4 and 5: Commit and activate the new host. Here, all the data is recovered, and the VM is started from exactly the place where it was suspended, but on the new host.
- Virtual Clusters are being widely used to use the computing resources effectively, generate HP, overcome the burden of interaction between different OSs and make different configurations to coexist.

Memory Migration

- **Memory Migration: This is done between the physical host and any other physical/virtual machine. The techniques used here depend upon the guest OS. MM can be in a range of megabytes to gigabytes. The Internet Suspend-Resume (ISR) technique exploits temporal locality since the memory states are may have overlaps in the suspended/resumed instances of a VM. Temporal locality (TL) refers to the fact that the memory states differ only by the amount of work done since a VM was last suspended.**
- **To utilize the TL, each file is represented as a tree of small sub-files. A copy of this tree exists in both the running and suspended instances of the VM. The advantage here is usage of tree representation of a file and caching ensures that the changed files are only utilized for transmission.**

File System Migration

- **File System Migration: To support VM migration from one cluster to another, a consistent and location-dependent view of the file system is available on all hosts. Each VM is provided with its own virtual disk to which the file system is mapped to. The contents of the VM can be transmitted across the cluster by inter-connections (mapping) between the hosts. But migration of an entire host (if required) is not advisable due to cost and security problems. We can also provide a global file system across all host machines where a VM can be located. This methodology removes the need of copying files from one machine to another – all files on all machines can be accessed through network.**

Smart Copying

- **Smart Copying ensures that after being resumed from suspension state, a VM doesn't get a whole file as a backup. It receives only the changes that were made. This technique reduces the amount of data that has to be moved between two locations.**

Network Migration

- **Network Migration: A migrating should maintain open network connections. It should not depend upon forwarding mechanisms (mediators) or mobile mechanisms. Each VM should be assigned a unique IP or MAC (Media Access Control) [7] addresses which is different from that of the host machine. The mapping of the IP and MAC addresses to their respective VMs is done by the VMM.**
- **If the destination of the VM is also on the same LAN, special messages are sent using MAC address that the IP address of the VM has moved to a new location. If the destination is on another network, the migrating OS can keep its original Ethernet MAC address and depend on the network switch [9] to detect its move to a new port [8].**

Two approaches in live migration

- There are two approaches in live migration: pre copy and post copy.
- In pre copy, which is manly used in live migration, all memory pages are first transferred; it then copies the modified pages in the last round iteratively. Here, performance 'degradation' will occur because migration will be encountering dirty pages (pages that change during networking) [10] all around in the network before getting to the right destination. The iterations could also increase, causing another problem. To encounter these problems, check-pointing/recovery process is used at different positions to take care of the above problems and increase the performance.

Two approaches in live migration

- In post-copy, all memory pages are transferred only once during the migration process. The threshold time allocated for migration is reduced. But the downtime is higher than that in pre-copy.
- NOTE: Downtime means the time in which a system is out of action or can't handle other works.
- Ex: Live migration between two Xen-enabled hosts: Figure 3.22 [1]
- CBC Compression=> Context Based Compression
- RDMA=> Remote Direct memory Access

Two **approaches in live migration**

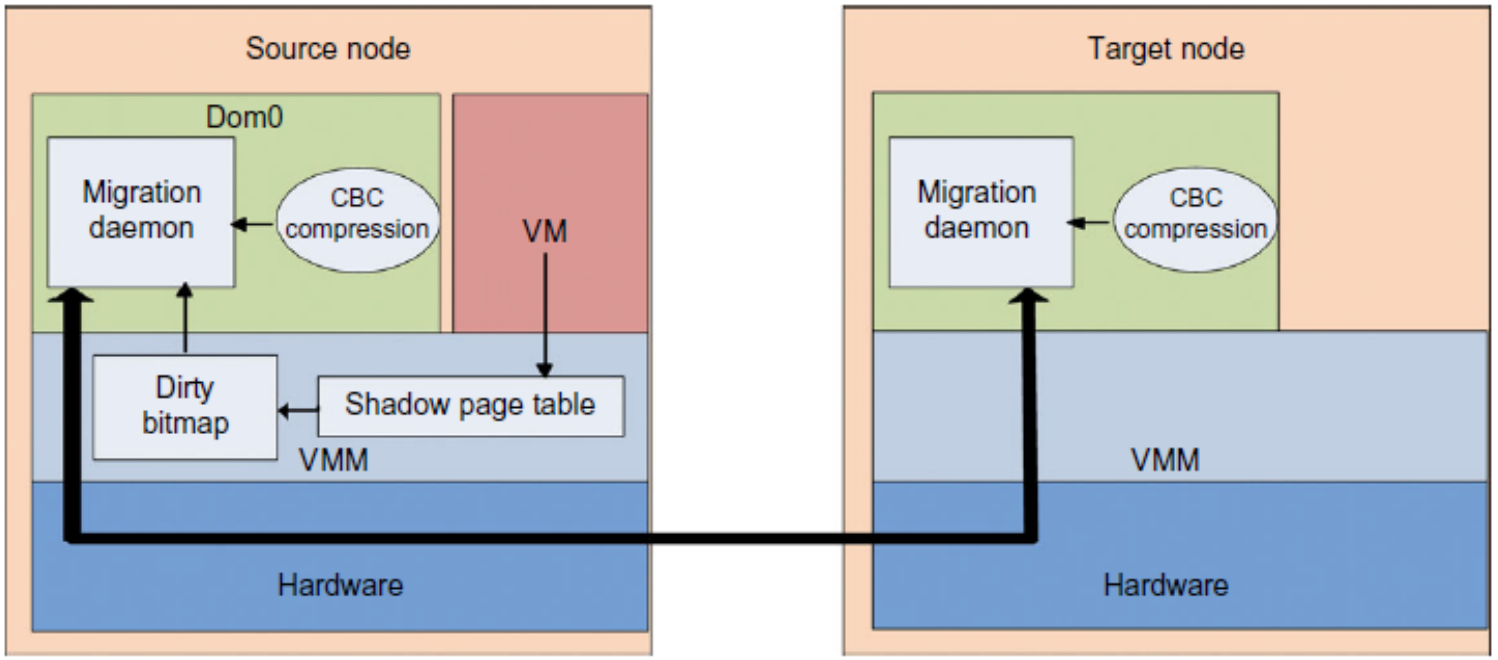


FIGURE 3.22

Live migration of VM from the Dom0 domain to a Xen-enabled target host.

Virtual Storage Management

- **Virtual Storage Management: VZ is mainly lagging behind the modernisation of data centers and is the bottleneck of VM deployment. The CPUs are rarely updates, the chips are not replaced and the host/guest operating systems are not adjusted as per the demands of situation.**
- **EC2 => Amazon Elastic Compute Cloud**
- **WS => Web Service**
- **CLI => Command Line Interface**
- **WSRF => Web Services Resource Framework**
- **KVM => Kernel-based VM**
- **VMFS => VM File System**
- **HA => High Availability**

Cloud OS for VZ Data Centers

Table 3.6 VI Managers and Operating Systems for Virtualizing Data Centers [9]

Manager/ OS, Platforms, License	Resources Being Virtualized, Web Link	Client API, Language	Hypervisors Used	Public Cloud Interface	Special Features
Nimbus Linux, Apache v2	VM creation, virtual cluster, www .nimbusproject.org/	EC2 WS, WSRF, CLI	Xen, KVM	EC2	Virtual networks
Eucalyptus Linux, BSD	Virtual networking (Example 3.12 and [41]), www .eucalyptus.com/	EC2 WS, CLI	Xen, KVM	EC2	Virtual networks
OpenNebula Linux, Apache v2	Management of VM, host, virtual network, and scheduling tools, www.opennebula.org/	XML-RPC, CLI, Java	Xen, KVM	EC2, Elastic Host	Virtual networks, dynamic provisioning
vSphere 4 Linux, Windows, proprietary	Virtualizing OS for data centers (Example 3.13), www .vmware.com/ products/vsphere/ [66]	CLI, GUI, Portal, WS	VMware ESX, ESXi	VMware vCloud partners	Data protection, vStorage, VMFS, DRM, HA

Example of Eucalyptus for Virtual Networking of Pri Cloud

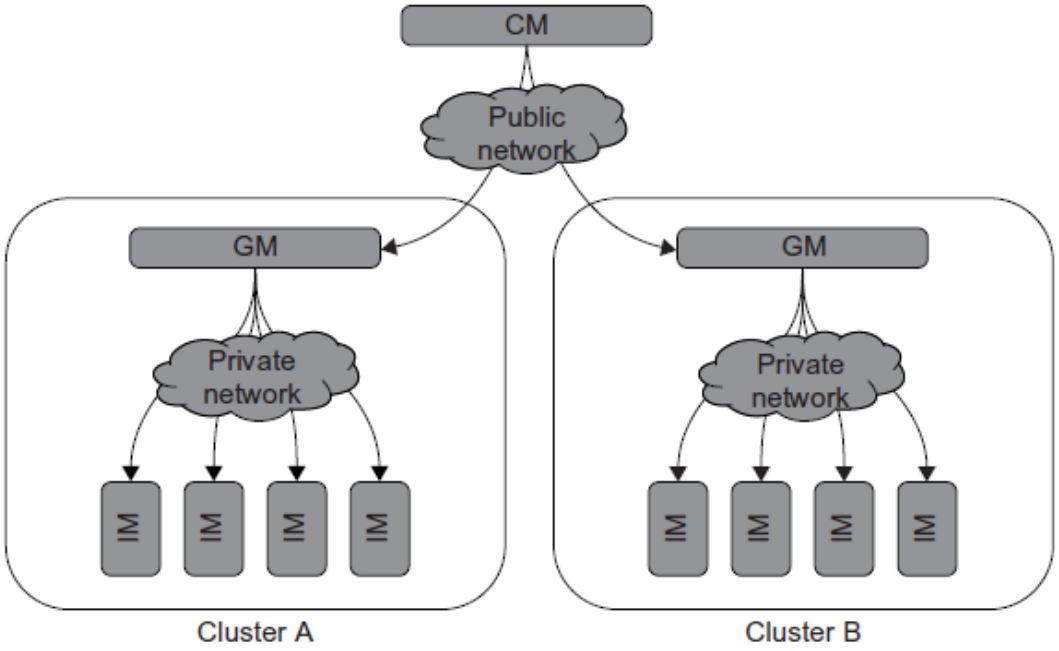


FIGURE 3.27

Eucalyptus for building private clouds by establishing virtual networks over the VMs linking through Ethernet and the Internet.

Virtual Storage Management

- **Instance Manager (IM): It controls execution, inspection and terminating of VM instances on the host machines where it runs.**
- **Group Manager (GM): It gathers information about VM execution and schedules them on specific IMs; it also manages virtual instance network.**
- **Cloud Manager (CM): It is an entry-point into the cloud for both users and administrators. It gathers information about the resources, allocates them by proper scheduling, and implements them through the GMs.**

VM-based Intrusion Detection

- **VM-based Intrusion Detection: Intrusions are unauthorized access to a computer from other network users. An intrusion detection system (IDS), which is built on the host OS can be divided into two types: host-based IDS (HIDS) and a network-based IDS (NIDS).**

Architecture of VM-based Intrusion Detection

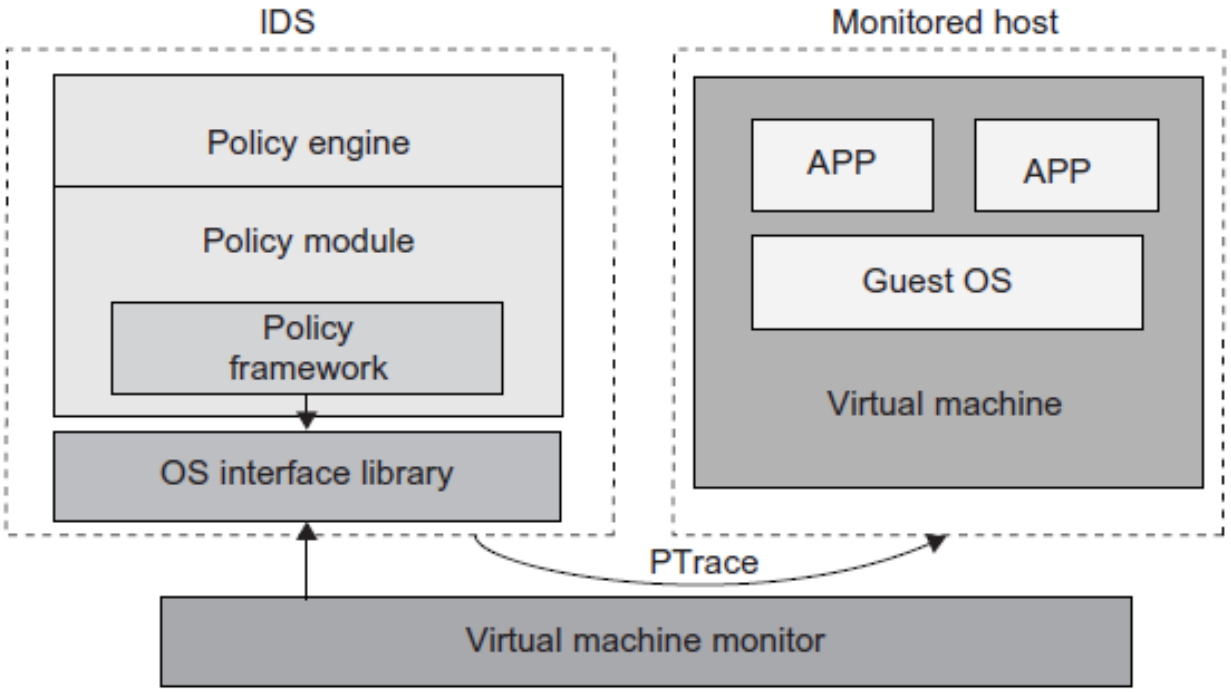


FIGURE 3.29

The architecture of livewire for intrusion detection using a dedicated VM.

VM-based Intrusion Detection

- **The above figure proposes the concept of granting IDS runs only on a highly-privileged VM. Notice that policies play an important role here. A policy framework can monitor the events in different guest operating systems of different VMs by using an OS interface library to determine which grant is secure and which is not.**
- **It is difficult to determine which access is intrusion and which is not without some time delay. Systems also may use access 'logs' to analyze which is an intrusion and which is secure. The IDS log service is based on the OS kernel and the UNIX kernel is hard to break; so even if a host machine is taken over by the hackers, the IDS log book remains unaffected.**
- **The security problems of the cloud mainly arise in the transport of the images through the network from one location to another. The VMM must be used more effectively and efficiently to deny any chances for the hackers.**