ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

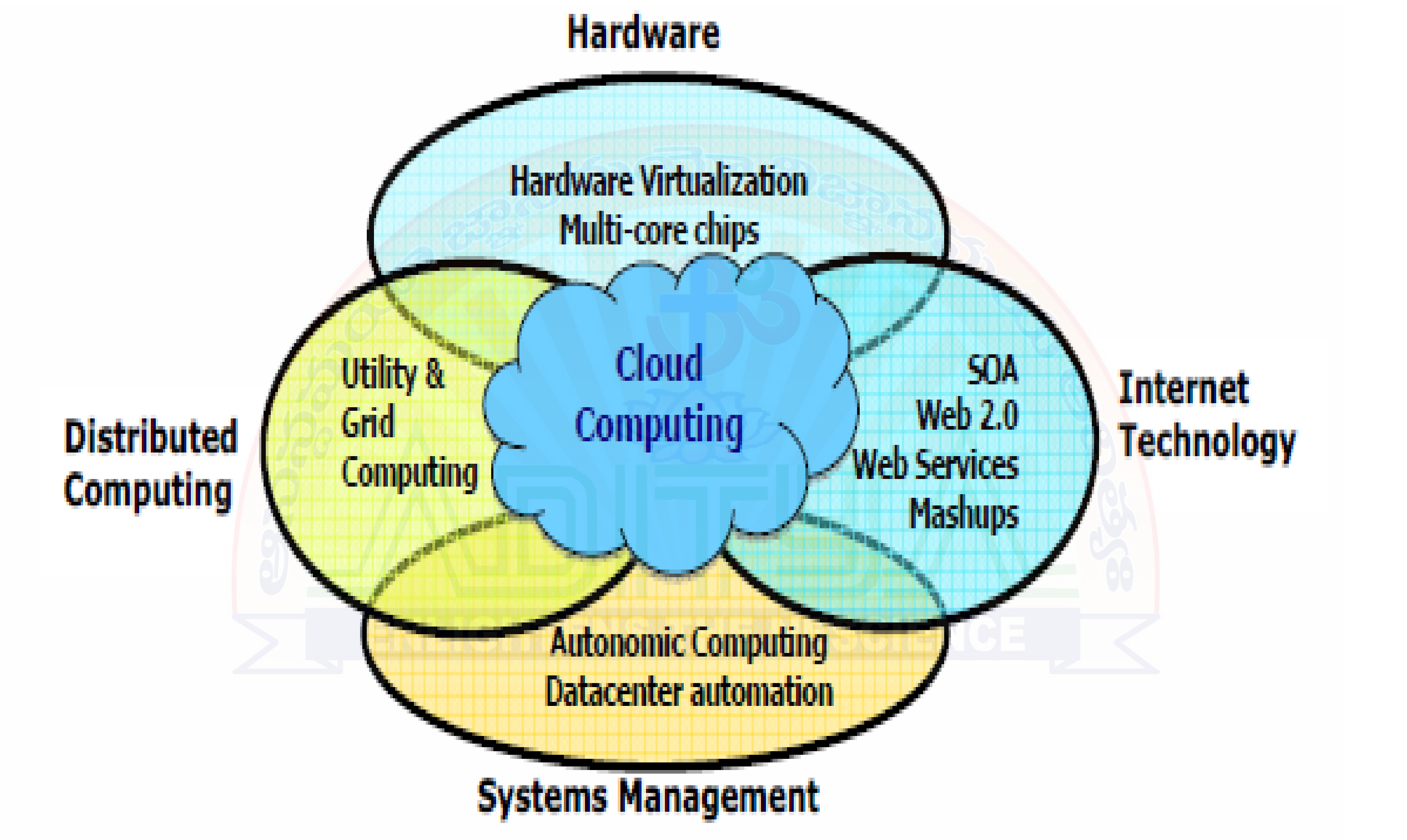
CLOUD COMPUTING

By

Prof. Ramesh SNSVSC
Dept of Computer Science and Engineering
Aditya College of Engineering & Technology
Surampalem.

UNIT-I
Systems modeling, Clustering and virtualization

Data Deluge Enabling New Challenges



Scalable Computing Over the Internet

- Past 60 years, computing technology has undergone a series of platform & environment changes in machine architecture, operating system platform, network connectivity, and application workload.
- Instead of using a centralized computer to solve computational problems, a parallel and distributed computing system uses multiple computers to solve large-scale problems over the Internet.
- Thus, distributed computing becomes data-intensive and network-centric.

Age of Internet Computing

- Billions of people use Internet every day. As a result, large data centers must provide high-performance computing services to huge numbers of users concurrently.
- Because of this high demand, the Linpack Benchmark for high-performance computing (HPC) applications is no longer optimal for measuring system performance.
- The emergence of computing clouds instead demands high-throughput computing (HTC) systems built with parallel and distributed computing technologies.
- We have to upgrade data centers using fast servers, storage systems, and high-bandwidth networks.
- The purpose is to advance network-based computing and web services with the emerging new technologies.

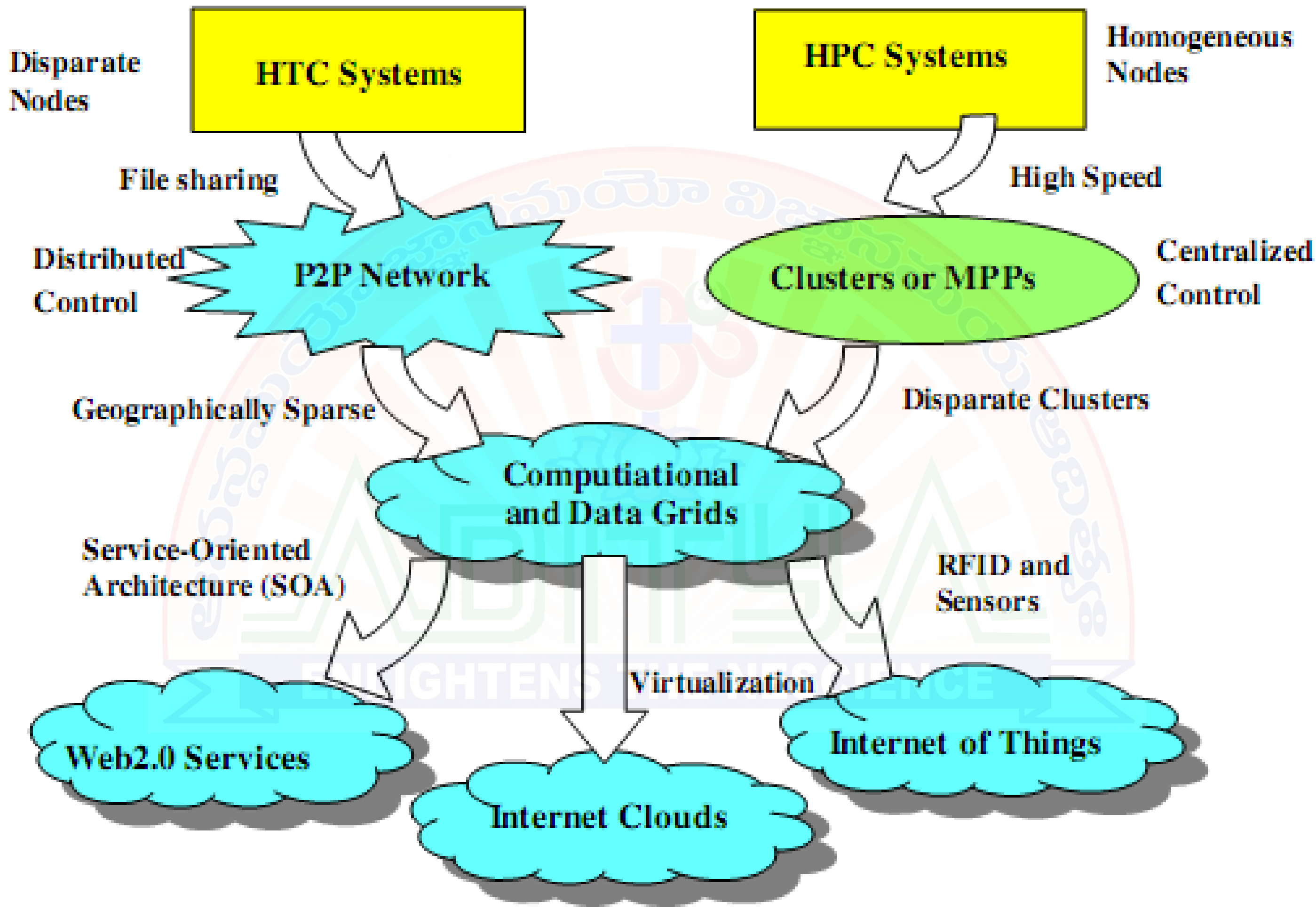
Platform Evolution

- Computer technology has gone through five generations of development, with each generation lasting from 10 to 20 years. Successive generations are overlapped in about 10 years.
- 1950 to 1970, a handful of mainframes, including the IBM 360 and CDC 6400, were built to satisfy the demands of large businesses and government organizations.
- From 1960 to 1980, lower-cost minicomputers such as the DEC PDP 11 and VAX Series became popular among small businesses and on college campuses.
- From 1970 to 1990, we saw widespread use of personal computers built with VLSI microprocessors.
- From 1980 to 2000, massive numbers of portable computers and pervasive devices appeared in both wired and wireless applications.
- Since 1990, the use of both HPC and HTC systems hidden clusters, grids or clouds have proliferated.

Platform Evolution

- The general computing trend is to leverage shared web resources and massive amounts of data over the Internet.
- The below figure illustrates the evolution of HPC and HTC systems.
- On the HPC side, supercomputers (massively parallel processors or MPPs) are gradually replaced by clusters of cooperative computers out of a desire to share computing resources.
- The cluster is often a collection of homogeneous compute nodes that are physically connected in close range to one another.
- On the HTC side, peer-to-peer (P2P) networks are formed for distributed file sharing and content delivery applications.
- A P2P system is built over many client machines. Peer machines are globally distributed in nature.
- P2P, cloud computing, and web service platforms are more focused on HTC applications than on HPC applications.
- Clustering and P2P technologies lead to the development of computational grids or data grids.

Clouds and Internet of Things



HPC: High-Performance Computing, **HTC**: High-Throughput Computing, **P2P**: Peer to Peer, **MPP**: Massively Parallel Processors

High - Performance Computing(HPC)

- For many years, HPC systems emphasize the raw speed performance. The speed of HPC systems has increased from Gflops(10^9) in the early 1990s to now Pflops (10^{15}) in 2010.

- This improvement was driven mainly by the demands from scientific, engineering, and manufacturing communities.

- There are Top 500 most powerful computer systems in the world but the number of supercomputer users is limited to less than 10% of all computer users.

- Users are using desktop computers or large servers when they conduct Internet searches and market-driven computing tasks.

High - Throughput Computing(HTC)

- The development of market-oriented high-end computing systems is undergoing a strategic change from an HPC paradigm to an HTC paradigm.
- This HTC paradigm pays more attention to high-flux computing.
- The main application for high-flux computing is in Internet searches and web services by millions or more users simultaneously.
- The performance goal thus shifts to measure high throughput or the number of tasks completed per unit of time.
- HTC technology needs to not only improve in terms of batch processing speed, but also address the acute problems of cost, energy savings, security, and reliability at many data and enterprise computing centers.

From Desktop/HPC/Grids to Internet Clouds in 30 Years

HPC moving from centralized supercomputers to geographically distributed desktops, clusters, and grids to clouds over last 30 years

R/D efforts on HPC, clusters, Grids, P2P, and virtual machines has laid the foundation of cloud computing that has been greatly advocated since 2007

Location of computing infrastructure in areas with lower costs in hardware, software, datasets, space, and power requirements – moving from desktop computing to datacenter-based clouds

Three New Computing Paradigms

- With the introduction of SOA, Web 2.0 services become available.
- Advances in virtualization make it possible to see the growth of Internet clouds as a new computing paradigm.
- The maturity of Radio-frequency identification (**RFID**), Global Positioning System (**GPS**), and sensor technologies has triggered the development of the Internet of Things (**IoT**).
- Computer definition changed to network, datacenter and now a cloud. i.e
  - The network is the computer.
  - The data center is the computer.
  - The cloud is the computer.

Computing Paradigm Distinctions

- Centralized Computing
  - All computer resources are centralized in one physical system.
- Parallel Computing
  - All processors are either tightly coupled with central shared memory or loosely coupled with distributed memory
- Distributed Computing
  - A distributed system consists of multiple autonomous computers, each with its own private memory, communicating over a network.
- Cloud Computing
  - An Internet cloud of resources that may be either centralized or decentralized. The cloud applies to parallel or distributed computing or both. Clouds may be built from physical or virtualized resources.

Computing Paradigm Distinctions

- Concurrent computing or concurrent programming refer to the union of parallel computing and distributing computing.
- Ubiquitous computing refers to computing with pervasive devices at any place and time using wired or wireless communication.
- The Internet of Things (IoT) is a networked connection of everyday objects including computers, sensors, humans, etc.
- The IoT is supported by Internet clouds to achieve ubiquitous computing with any object at any place and time. Finally, the term Internet computing is even broader and covers all computing paradigms over the Internet.

Distributed System Families

- In the future, both HPC and HTC systems will demand multicore or many-core processors that can handle large numbers of computing threads per core.
- Both HPC and HTC systems emphasize parallelism and distributed computing.
- Future HPC and HTC systems must be able to satisfy this huge demand in computing power in terms of throughput, efficiency, scalability, and reliability.
- The system efficiency is decided by speed, programming, and energy factors. Meeting these goals requires to yield the following design objectives:

Distributed System Families

- **Efficiency** measures the utilization rate of resources in an execution model by exploiting massive parallelism in HPC. For HTC, efficiency is more closely related to job throughput, data access, storage, and power efficiency.

- **Dependability** measures the reliability and self-management from the chip to the system and application levels. The purpose is to provide high-throughput service with Quality of Service (QoS) assurance, even under failure conditions.

- **Adaptation** in the programming model measures the ability to support billions of job requests over massive data sets and virtualized cloud resources under various workload and service models.

- **Flexibility** in application deployment measures the ability of distributed systems to run well in both HPC (science and engineering) and HTC (business) applications.

Scalable Computing Trends and New Paradigms

- **Moore's law** indicates that processor speed doubles every 18 months.
- **Gilder's law** indicates that network bandwidth has doubled each year in the past.
- The tremendous price/performance ratio of hardware was driven by the desktop, notebook, and tablet computing markets.
- Distributed System emphasize on both resource distribution and concurrency or high **Degree of Parallelism (DoP).**

Degrees of Parallelism

- **Bit-level parallelism (BLP)** converts bit-serial processing to word-level processing gradually. Over the years, users graduated from 4-bit microprocessors to 8-, 16-, 32-, and 64-bit CPUs. This led us to ILP.
- **Instruction-level Parallelism (ILP)** in which the processor executes multiple instructions simultaneously rather than only one instruction.
- **Data-level parallelism (DLP)** was made popular through SIMD (single instruction, multiple data) and vector machines using vector or array types of instructions.
- **Task-level parallelism (TLP)** is possible with the introduction of multicore processors and chip multiprocessors (CMPs).
- A modern processor explores all of the aforementioned parallelism types we will see an **Job-level parallelism (JLP)**.

Innovative Applications

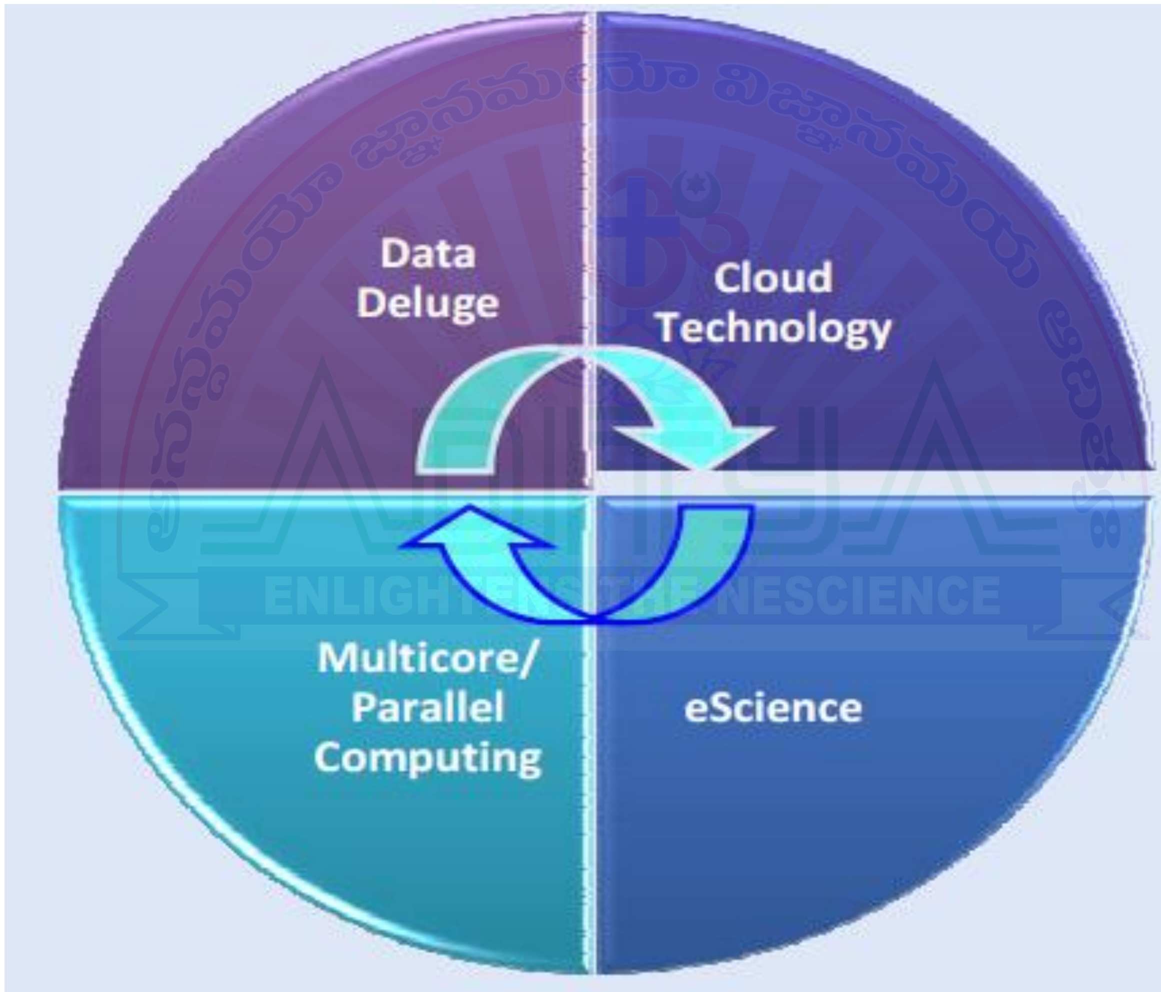
Table 1.1 Applications of High-Performance and High-Throughput Systems

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.

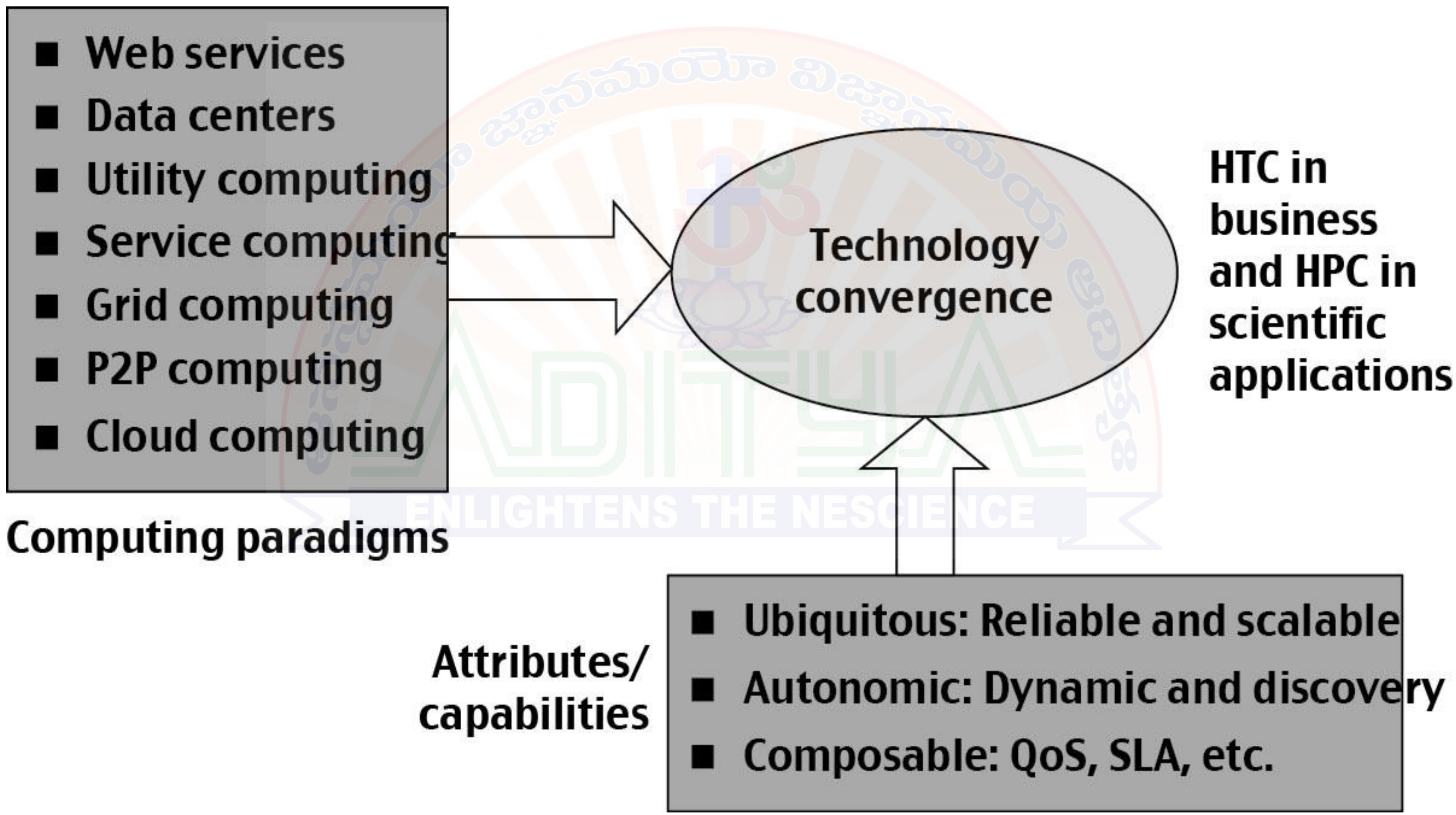
A trend towards Utility Computing

- The paradigms share some common characteristics. First, they are all ubiquitous in daily life. Reliability and scalability are two major design objectives
- Second, they are aimed at autonomic operations that can be self-organized to support dynamic discovery.
- Finally, these paradigms are composable with QoS(Quality of Service) and SLAs (service-level agreements).
- These paradigms and their attributes realize the computer utility vision.
- Utility computing focuses on a business model in which customers receive computing resources from a paid service provider. All grid/cloud platforms are regarded as utility service providers.

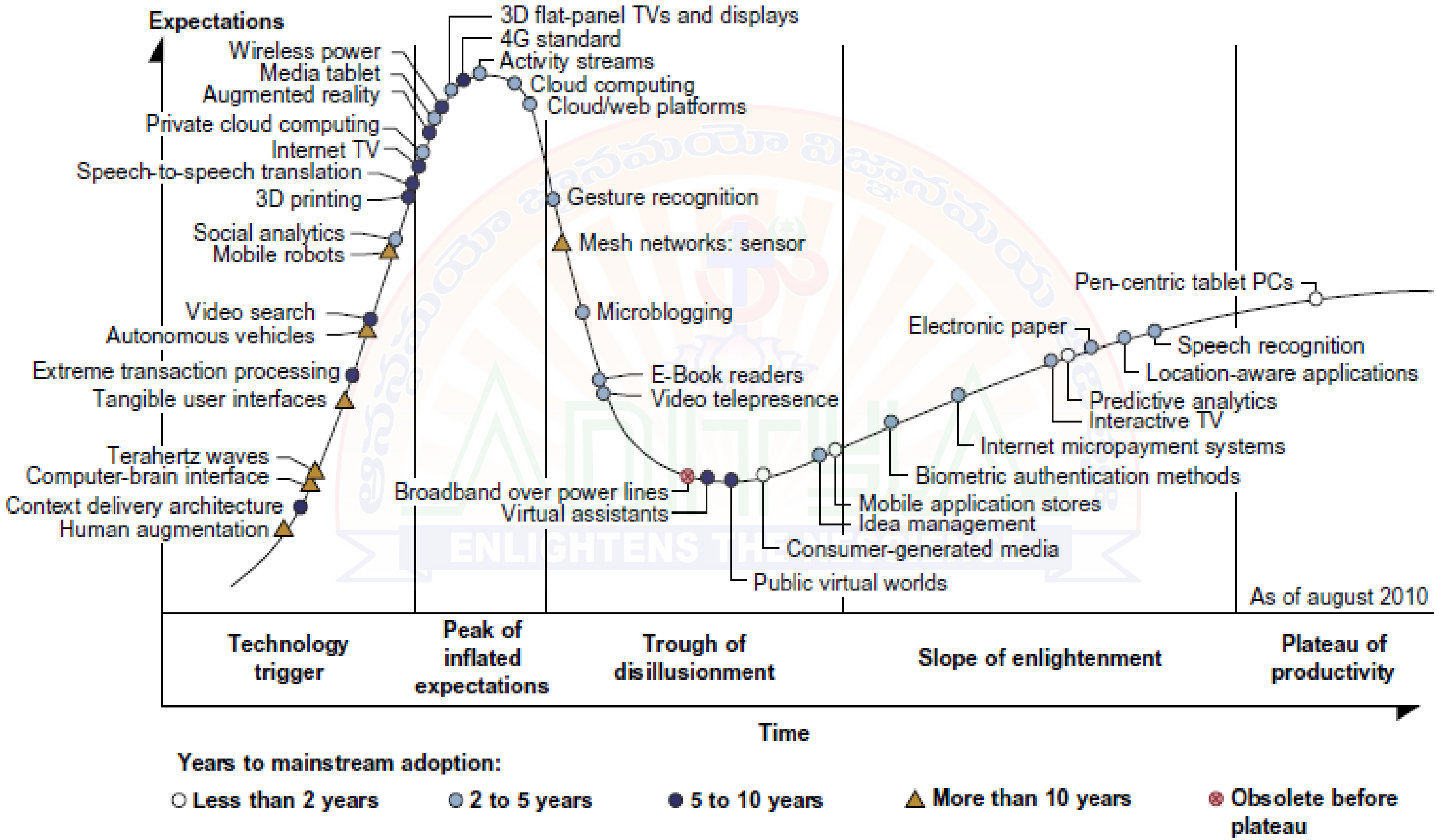
Interactions among 4 technical challenges : Data Deluge, Cloud Technology, eScience, and Multicore/Parallel Computing



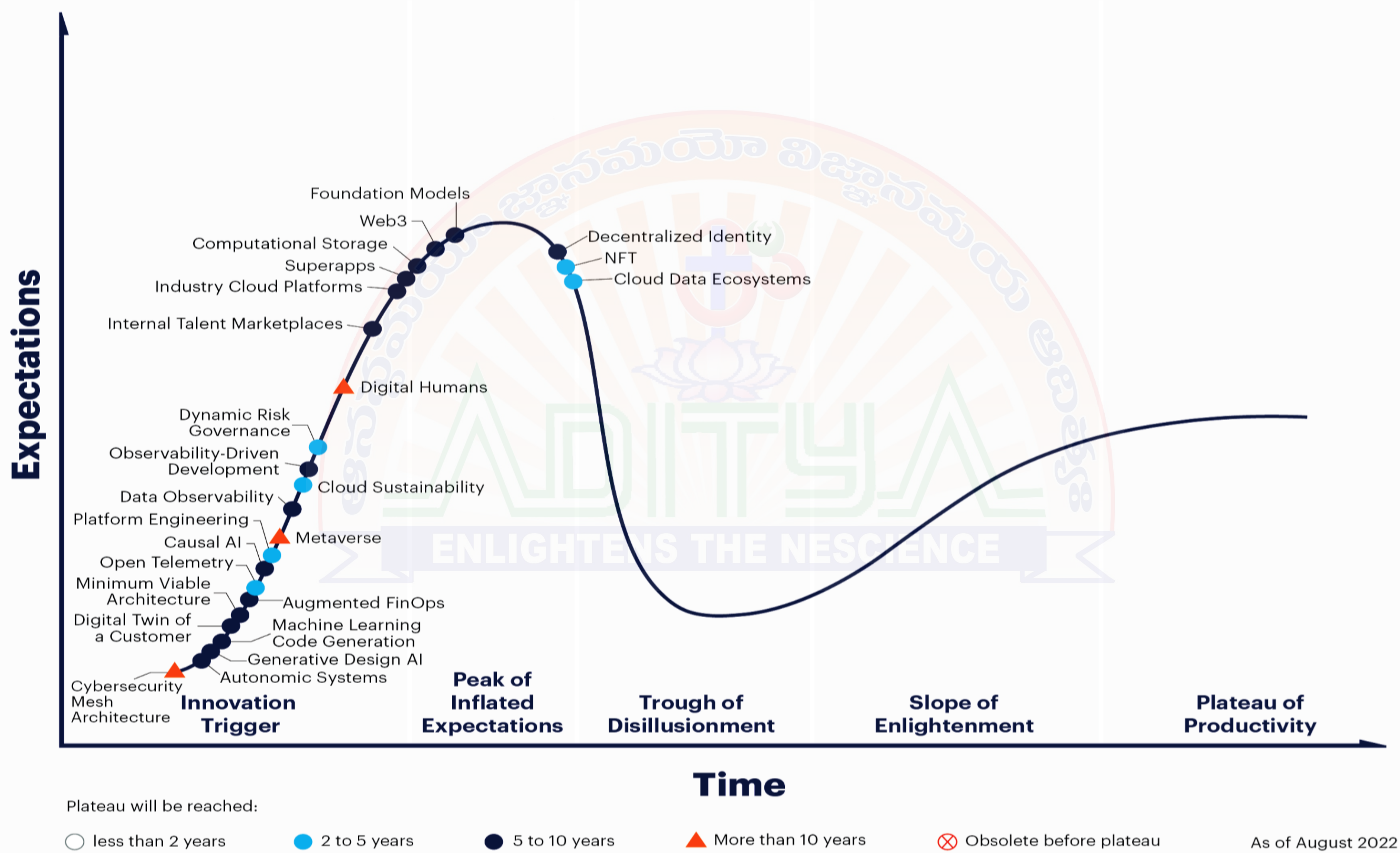
Technology Convergence toward HPC for Science and HTC for Business: A trend toward *Utility Computing*



Hype cycle for Emerging Technologies, 2010



Hype Cycle for Emerging Tech, 2022



The Internet of Things and Cyber-Physical Systems

- The recent two Internet development trends are :
  - Internet of Things
  - Cyber-Physical systems.

Internet of Things

- The concept of the IoT was introduced in 1999 at MIT
- The IoT refers to the networked interconnection of everyday objects, tools, devices, or computers.
- IoT is a wireless network of sensors that interconnect all things in our daily life.
- These things can be large or small and they vary with respect to time and place. The idea is to tag every object using RFID or a related sensor or electronic technology such as GPS

The Internet of Things and Cyber-Physical Systems

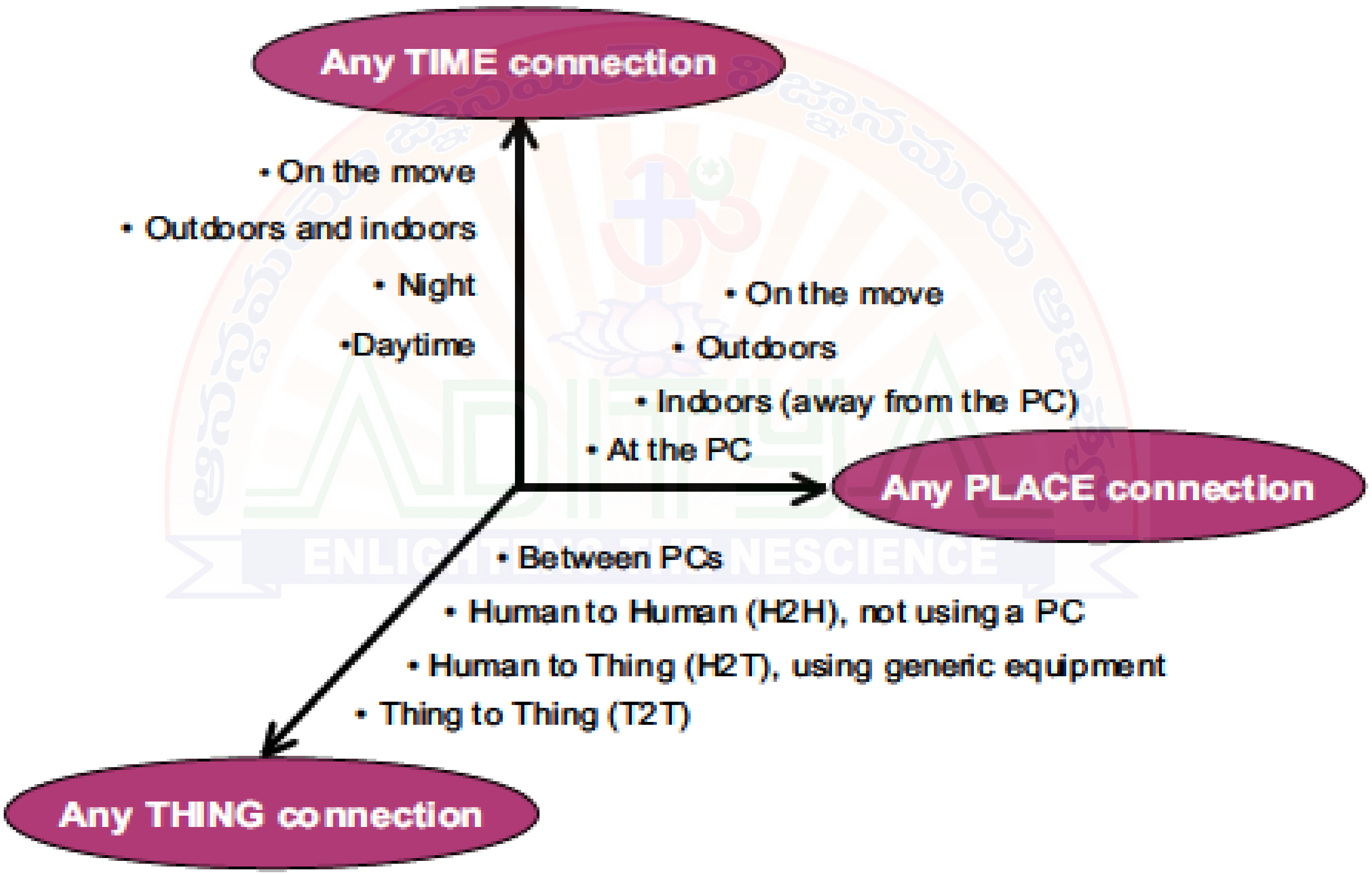
- In the IoT era, all objects and devices are instrumented, interconnected, and interacted with each other intelligently.

- Three communication patterns co-exist:
  - H2H (human-to-human),
  - H2T (human-to thing),
  - T2T (thing-to-thing).

- Cloud computing researchers expect to use the cloud and future Internet technologies to support fast, efficient, and intelligent interactions among humans, machines, and any objects on Earth.

The Internet of Things and Cyber-Physical Systems

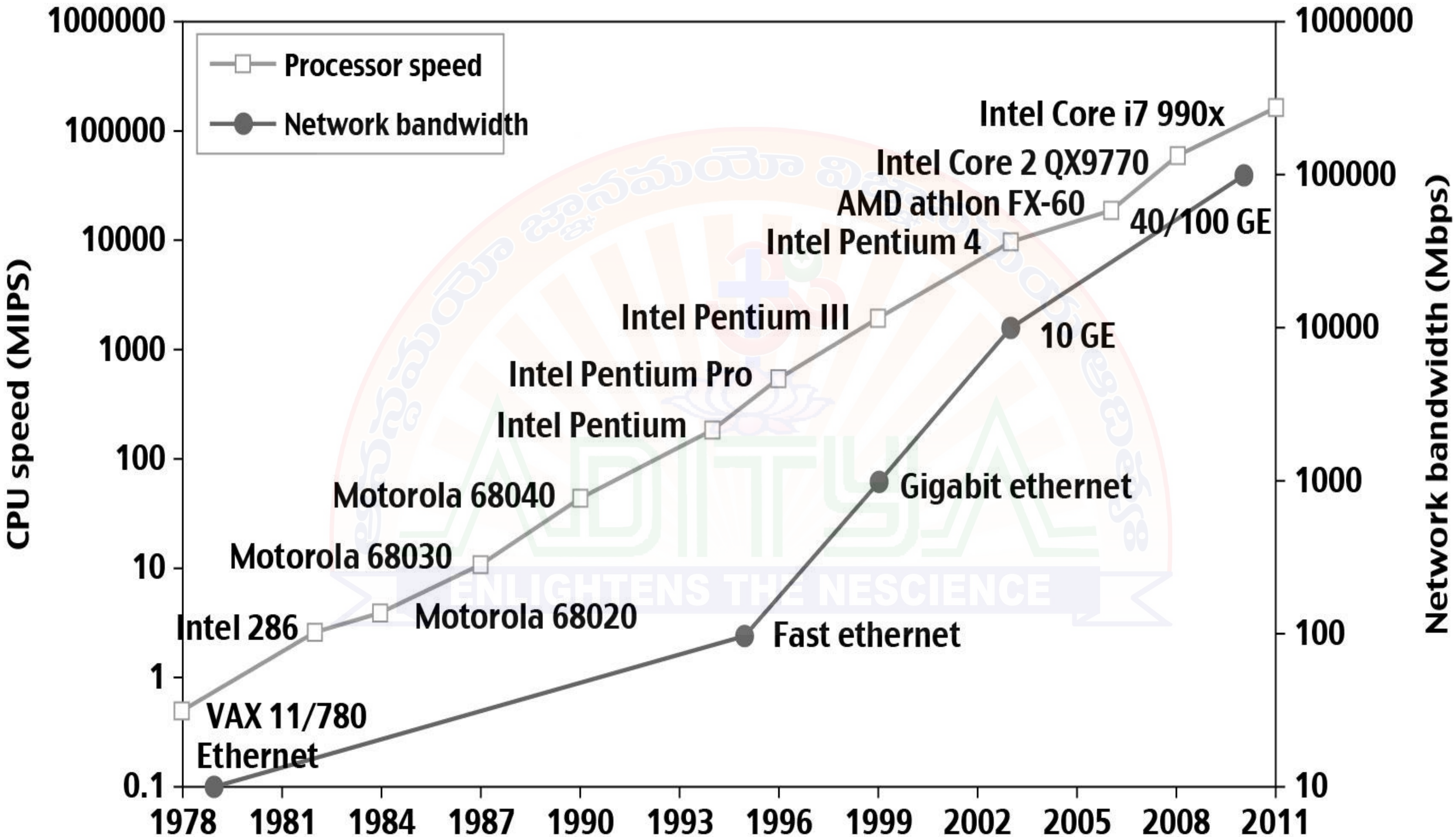
Opportunities of IoT in 3 Dimensions



The Internet of Things and Cyber-Physical Systems

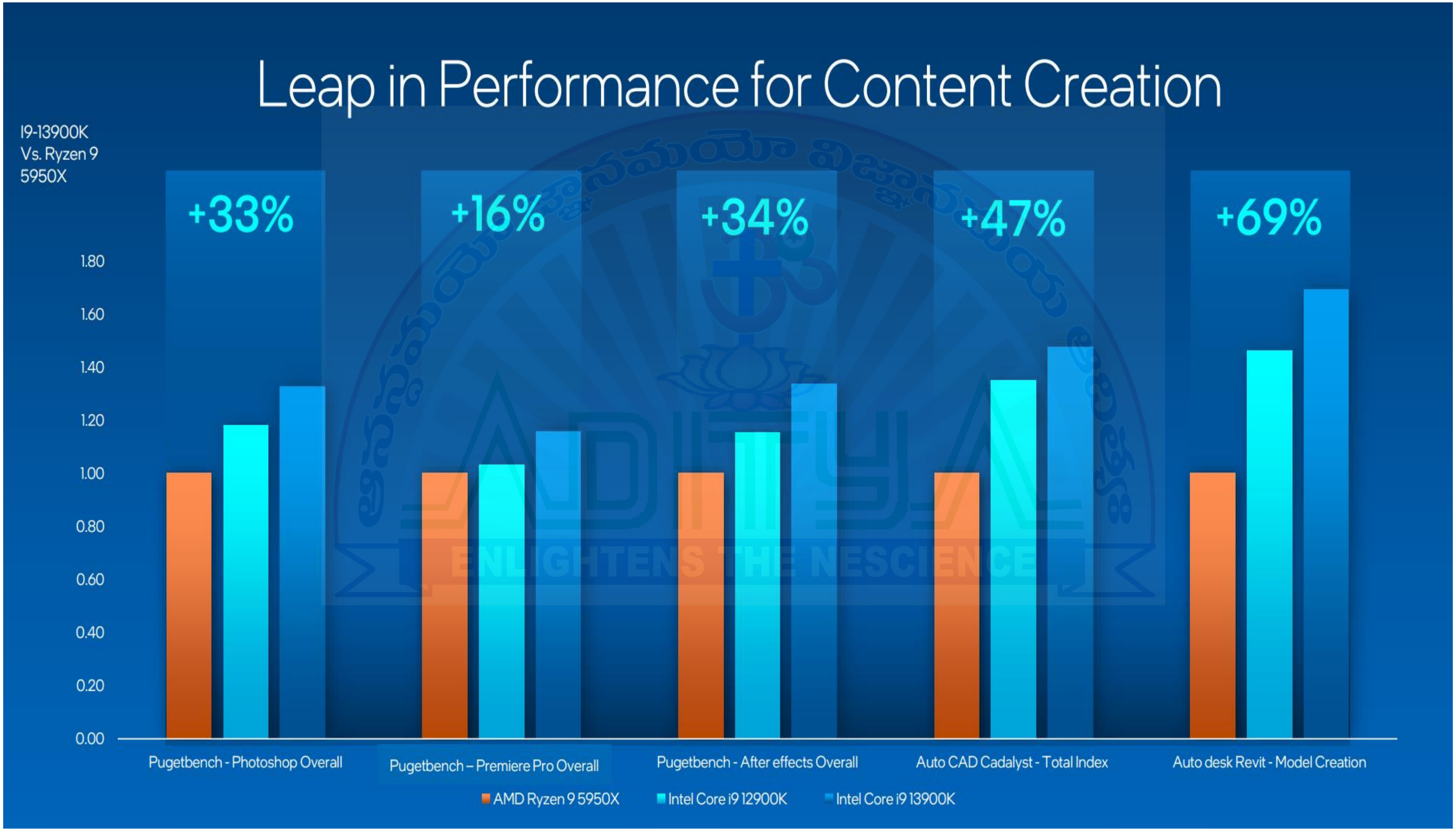
- **Cyber-Physical Systems**
- A cyber-physical system (CPS) is the result of interaction between computational processes and the physical world.
- A CPS integrates “Cyber” (heterogeneous, asynchronous) with “Physical” (concurrent and information-dense) objects.
- A CPS merges the “3C” technologies of
  - Computation,
  - Communication, and
  - Control … into an intelligent closed feedback system between the physical world and the information world.
- The IoT emphasizes various networking connections among physical objects, while the CPS emphasizes exploration of virtual reality (VR) applications in the physical world..

33 year Improvement in Processor and Network Technologies



* In 2023, latest Processor is i9 13900KS with 24 cores, 6.0 GHz and 36 MB Smart Cache

Improvement in Processor and Network Technologies



TECHNOLOGIES FOR NETWORK-BASED SYSTEMS

- It's time to explore hardware, software, and network technologies for distributed computing system design and applications.

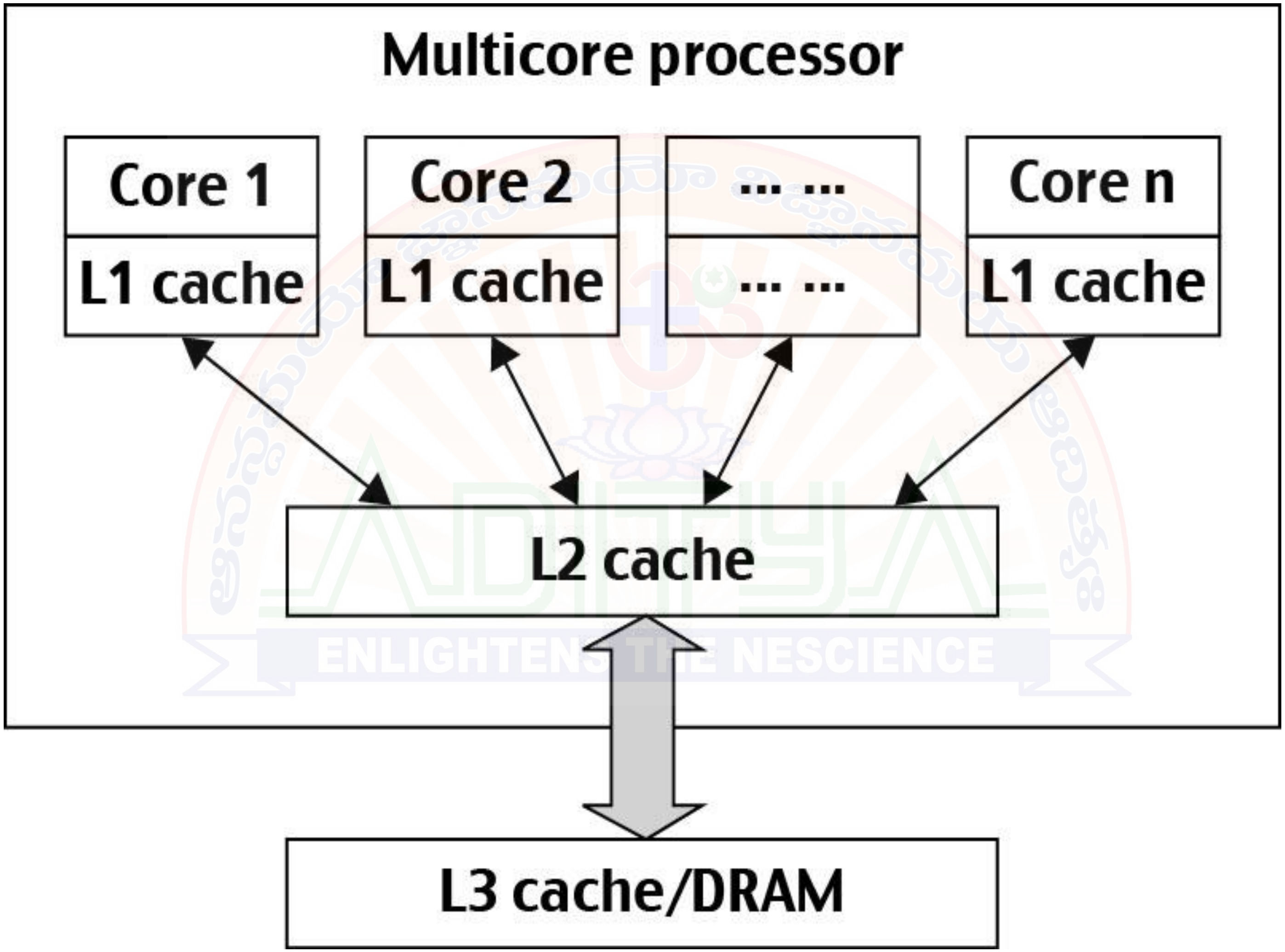
Multicore CPUs and Multithreading Technologies

Advances in CPU Processors

- Processor speed is measured in millions of instructions per second (MIPS) and network bandwidth is measured in megabits per second (Mbps) or gigabits per second (Gbps).
- The unit GE refers to 1 Gbps Ethernet bandwidth.
- Advanced CPUs or microprocessor chips assume a multicore architecture with dual, quad, six, or more processing cores.
- These processors exploit parallelism at ILP and TLP levels.

- These ILP techniques demand hardware and compiler support.
- In addition, DLP and TLP are highly explored in graphics processing units (GPUs) that adopt a many-core architecture with hundreds to thousands of simple cores.
- In modern multicore CPU(Figure below), each core is essentially a processor with its own private cache (L1 cache). Multiple cores are housed in the same chip with an L2 cache that is shared by all cores.
- In the future, multiple CMPs could be built on the same CPU chip with even the L3 cache on the chip.
- Multicore and multithreaded CPUs are equipped with many high-end processors, including the Intel i7, Xeon, AMD Opteron, Sun Niagara, IBM Power 6, and X cell processors. Each core could be also multithreaded.

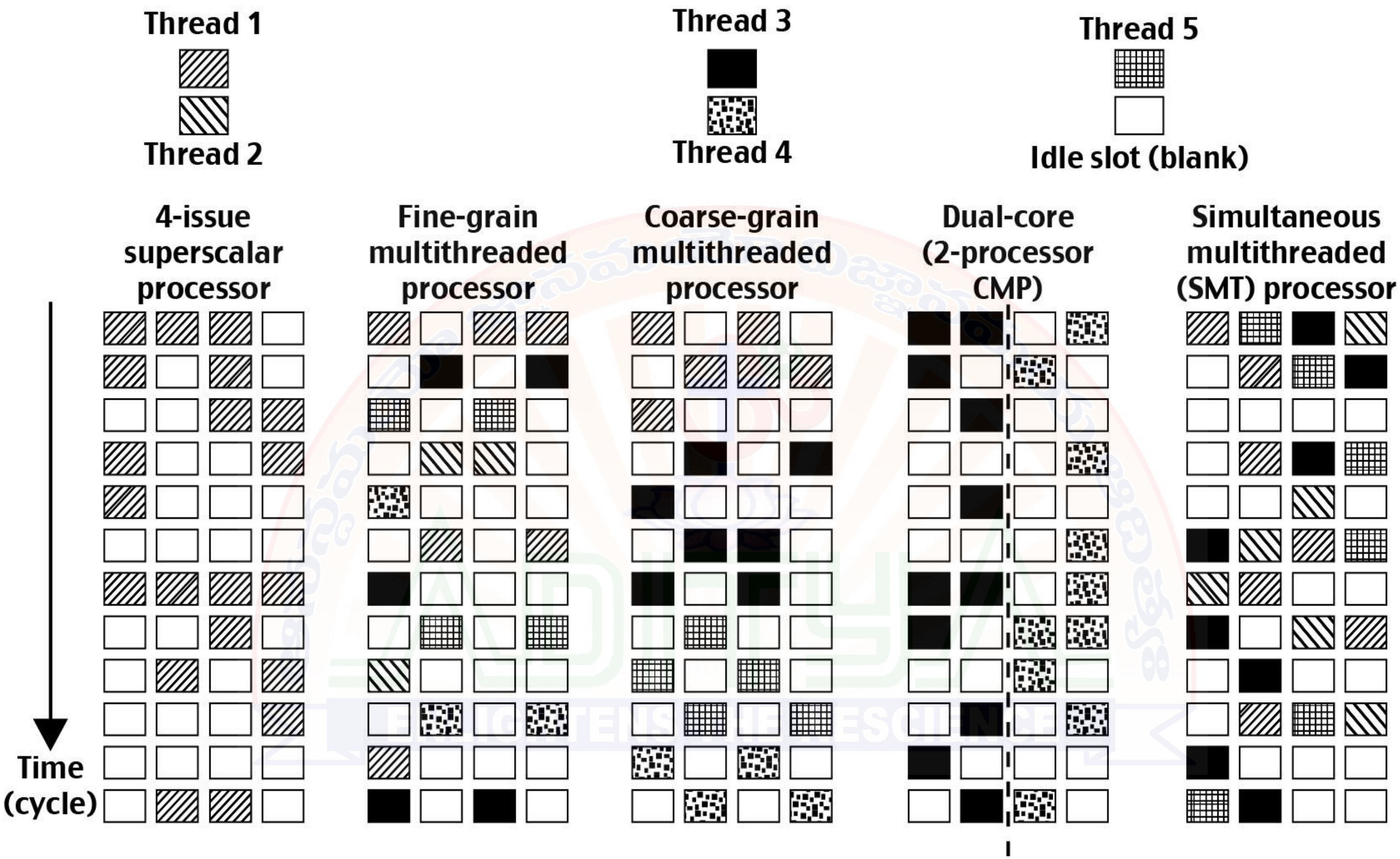
Modern Multi-core CPU Chip



Multicore CPU and Many-Core GPU Architectures

- Multicore CPUs may increase from the tens of cores to hundreds or more in the future. But the CPU has reached its limit in terms of exploiting massive DLP due to the memory problem.
- This has triggered the development of many-core GPUs with hundreds or more thin cores.
- The processor industry is also keen to develop asymmetric or heterogeneous chip multiprocessors that can house both fat CPU cores and thin GPU cores on the same chip.
- <u>Multithreading Technology (Below Figure)</u>
- Five independent threads of instructions
- Four pipelined data paths (functional units)
- Five processor categories

5 Micro-architectures of CPUs



Each row represents the issue slots for a single execution cycle:

- A filled box indicates that the processor found an instruction to execute in that issue slot on that cycle;
- An empty box denotes an unused slot.

Multi-threading Processors

- **Four-issue Superscalar**
  - Implements ILP within a single processor.
  - Executes more than one instruction during a clock cycle
  - Only instructions from the same thread are executed.
- **Fine-grain Multithreaded Processor**
  - Switches the execution of instructions from different threads per cycle
- **Coarse-grain Multithreaded Processor**
  - executes many instructions from the same thread for quite a few cycles before switching to another thread
- **Dual-core (2-processor CMP)**
  - assumes two processing cores, each a single-threaded two-way superscalar processor.
- **Simultaneous Multithread Processor (SMT)**
  - allows simultaneous scheduling of instructions from different threads in the same cycle..

GPU Computing to Exa scale and Beyond

- A GPU is a graphics coprocessor or accelerator mounted on a computer's graphics card or video card.
- A GPU offloads the CPU from tedious graphics tasks in video editing applications.
- The world's first GPU, the GeForce 256, was marketed by NVIDIA in 1999.
- These GPU chips can process a minimum of 10 million polygons per second.
- Unlike CPUs, GPUs have a throughput architecture that exploits massive parallelism by executing many concurrent threads slowly, instead of executing a single long thread very quickly.
- General Purpose computing on GPUs – GPGPUs are used in HPC

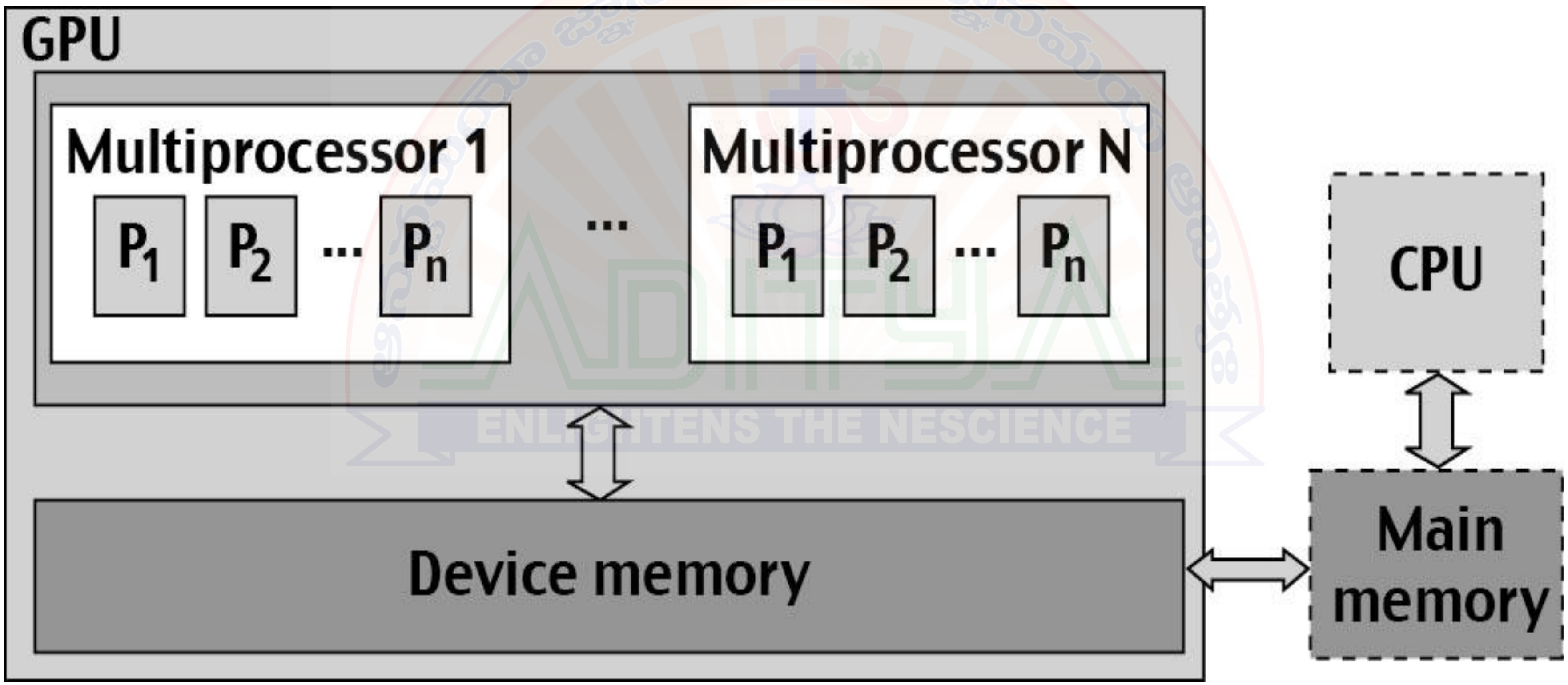
How GPUs Work

- Early GPUs functioned as coprocessors attached to the CPU. Today, the NVIDIA GPU has been upgraded to 128 cores on a single chip.
- Each core on a GPU can handle eight threads of instructions.
- This translates to having up to 1,024 threads executed concurrently on a single GPU.
- This is true massive parallelism, compared to only a few threads that can be handled by a conventional CPU.
- GPU is optimized to deliver much higher throughput with explicit management of on-chip memory as it is designed to handle large numbers of FLOPS in parallel.
- Conventional GPUs - used in mobile phones, game consoles, embedded systems, PCs, and servers.
- The NVIDIA CUDA Tesla or Fermi is used in GPU clusters or in HPC systems for parallel processing of massive floating-pointing data.

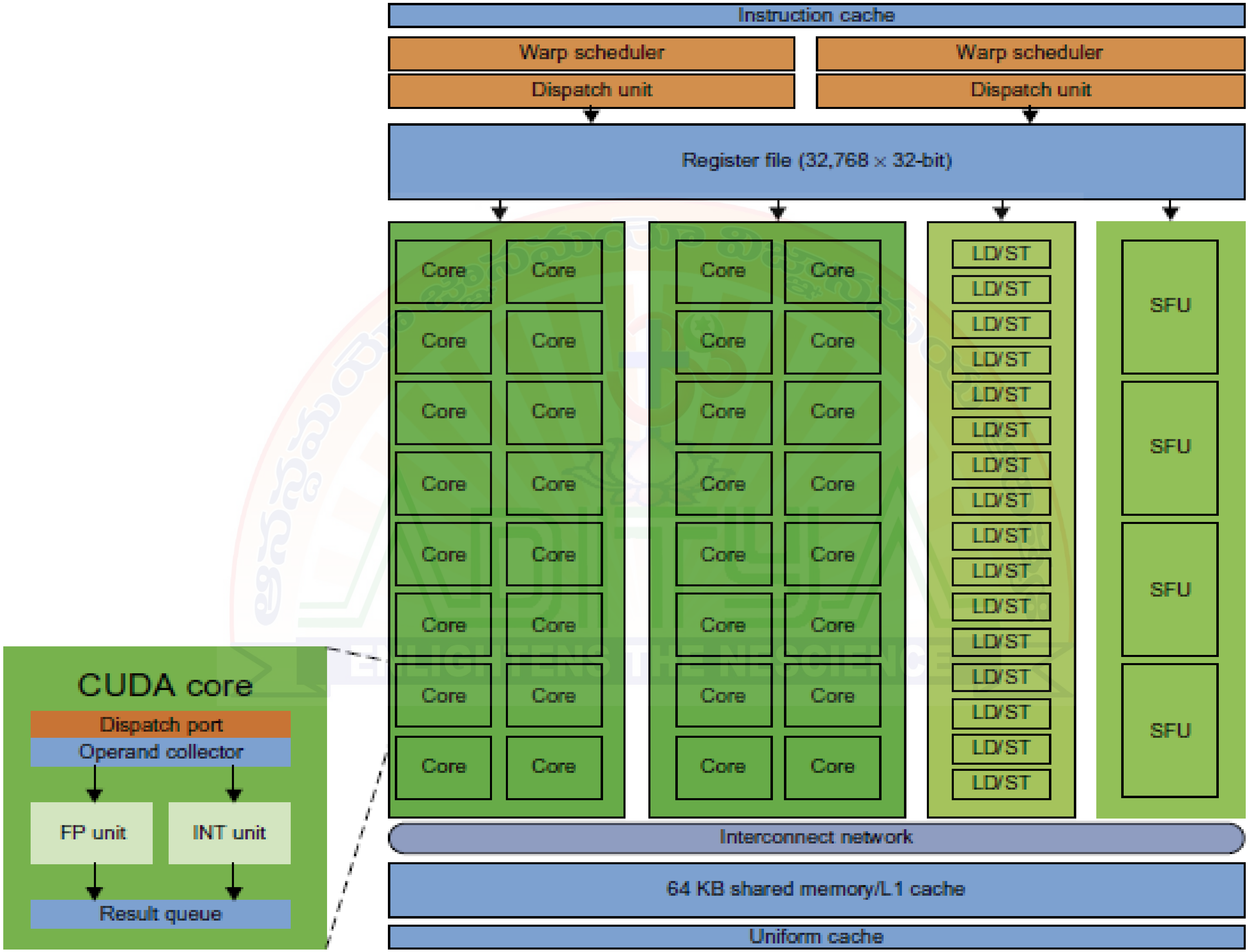
GPU Programming Model

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

Architecture of A Many-Core Multiprocessor GPU interacting with a CPU Processor

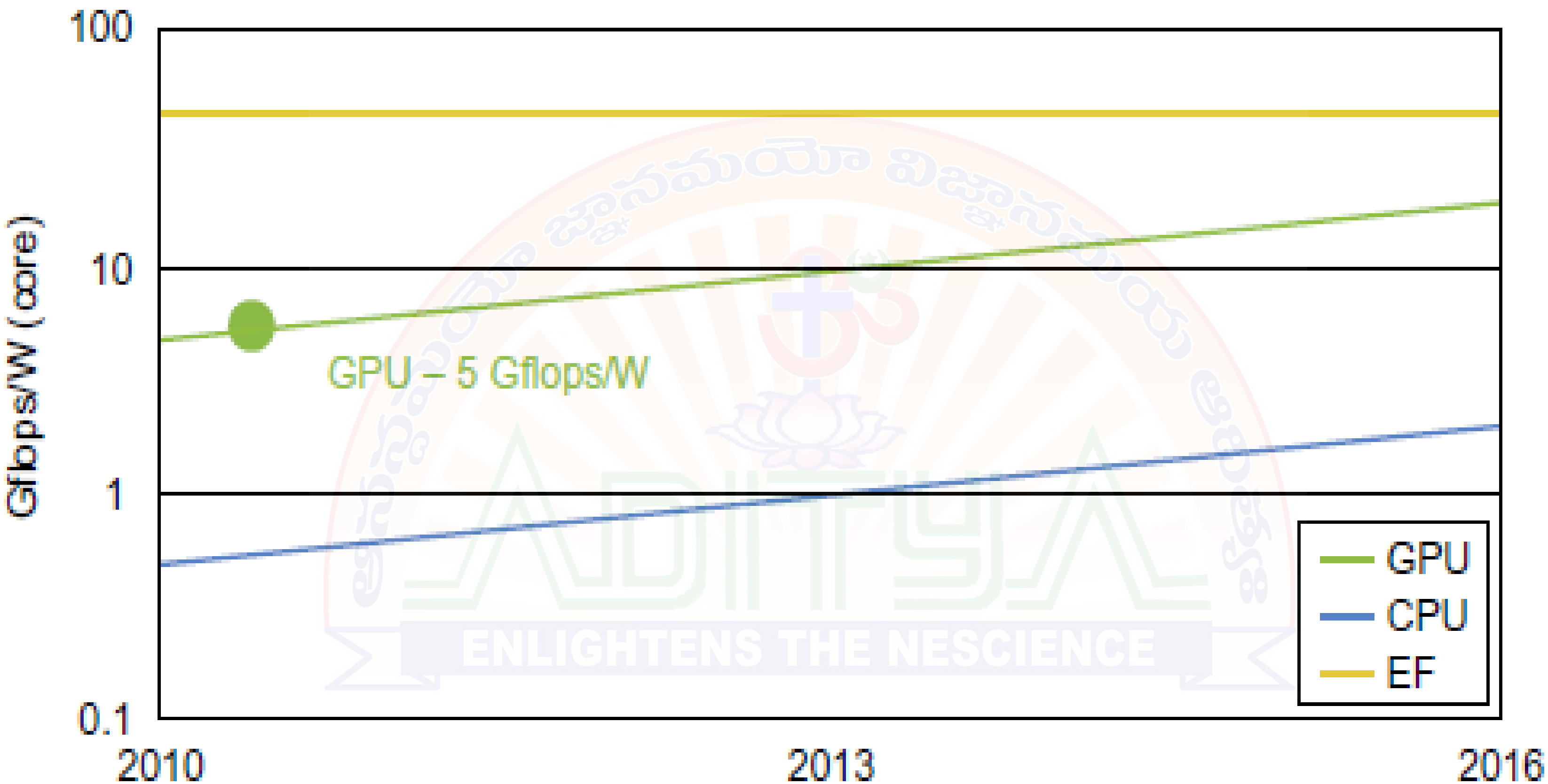


An Example - NVIDIA Fermi GPU



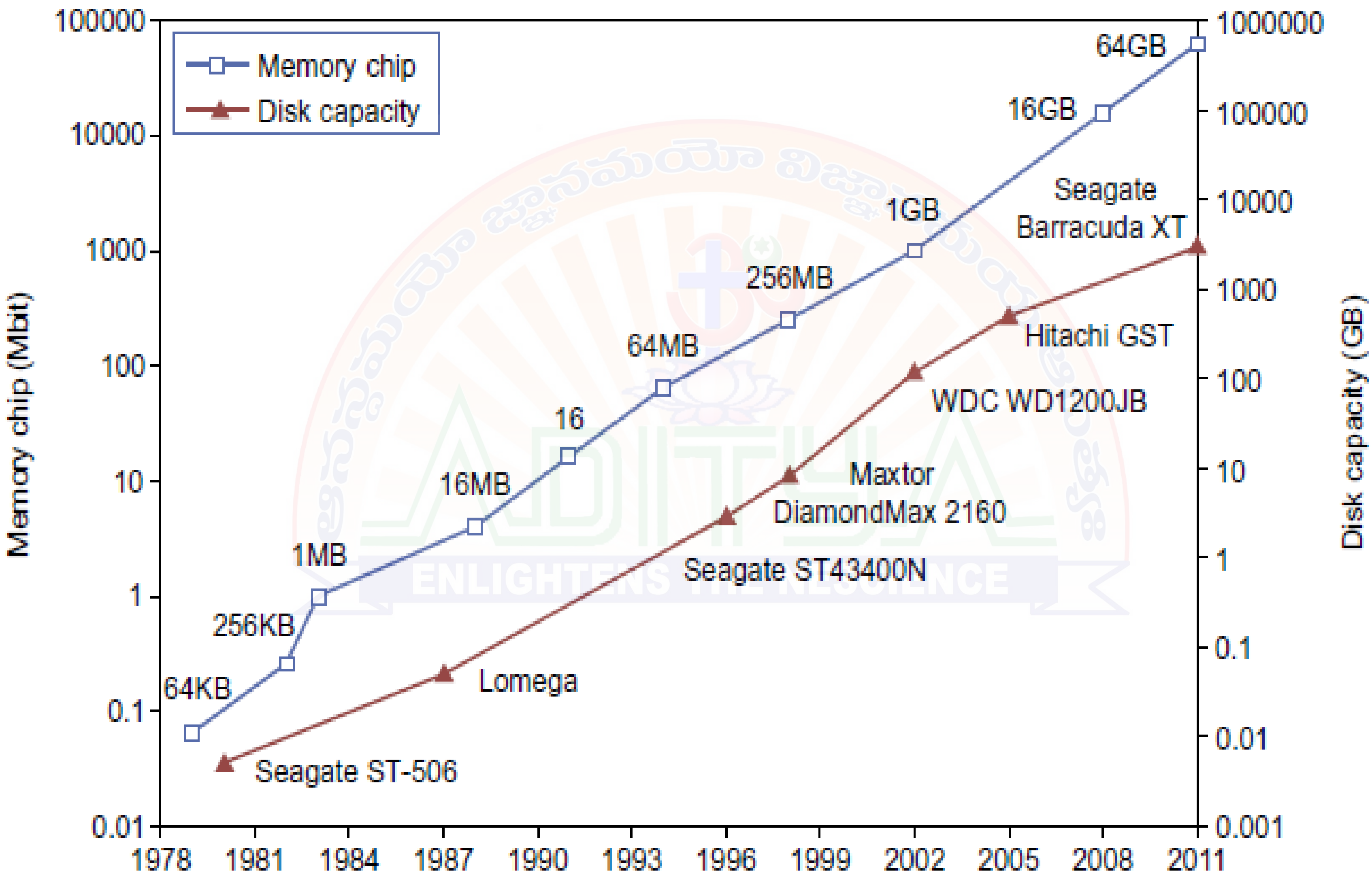
Load/Store Units (LD/ST) and Special Function Units (SFUs)

GPU Performance

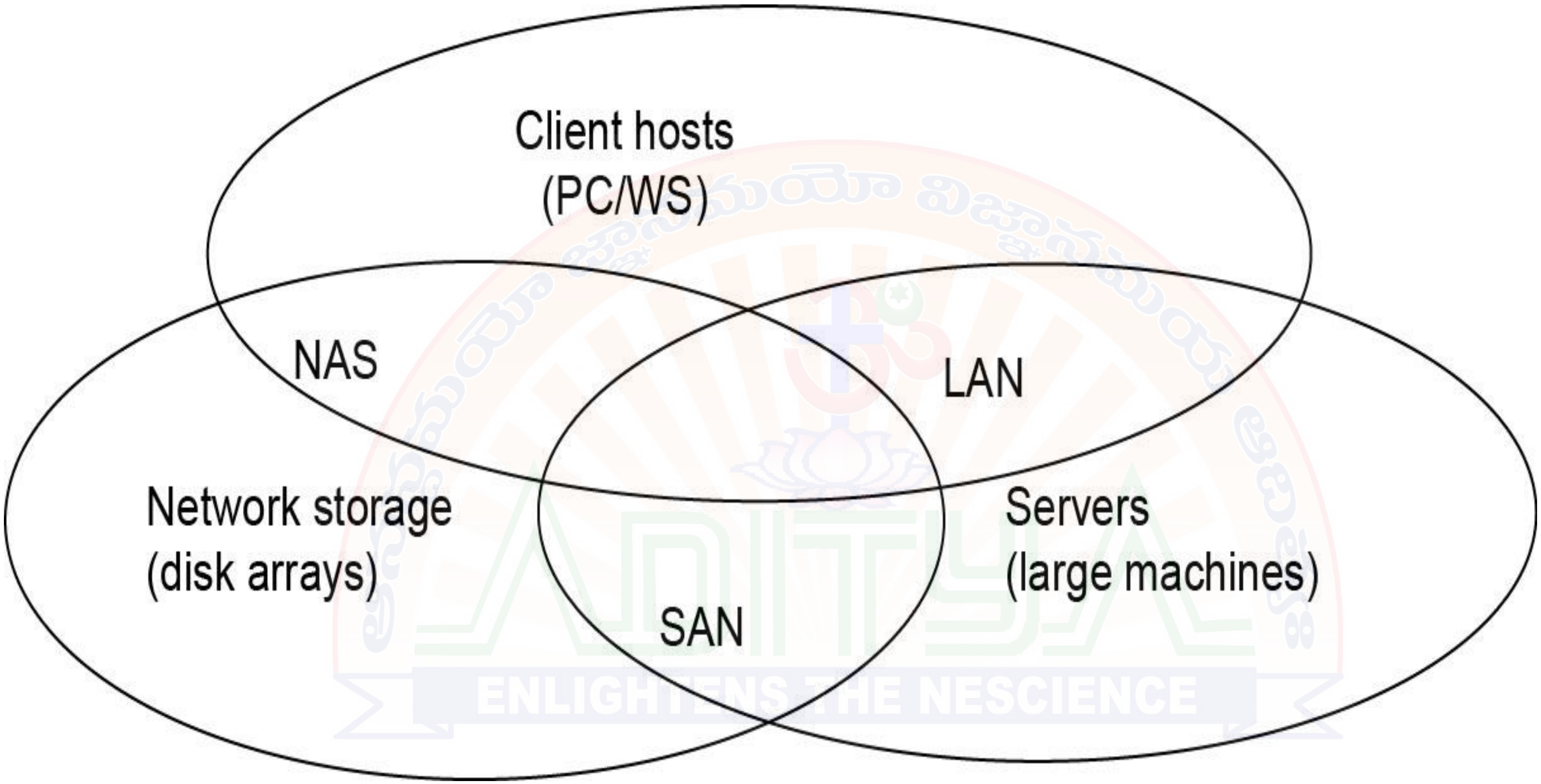


Bottom – CPU - 0.8 Gflops/Watt/Core (2011)
Middle – GPU - 5 Gflops/Watt/Core (2011)
Top - EF – Exascale computing (10^18 Flops)

33 year Improvement in Memory and Disk Technologies



Interconnection Networks

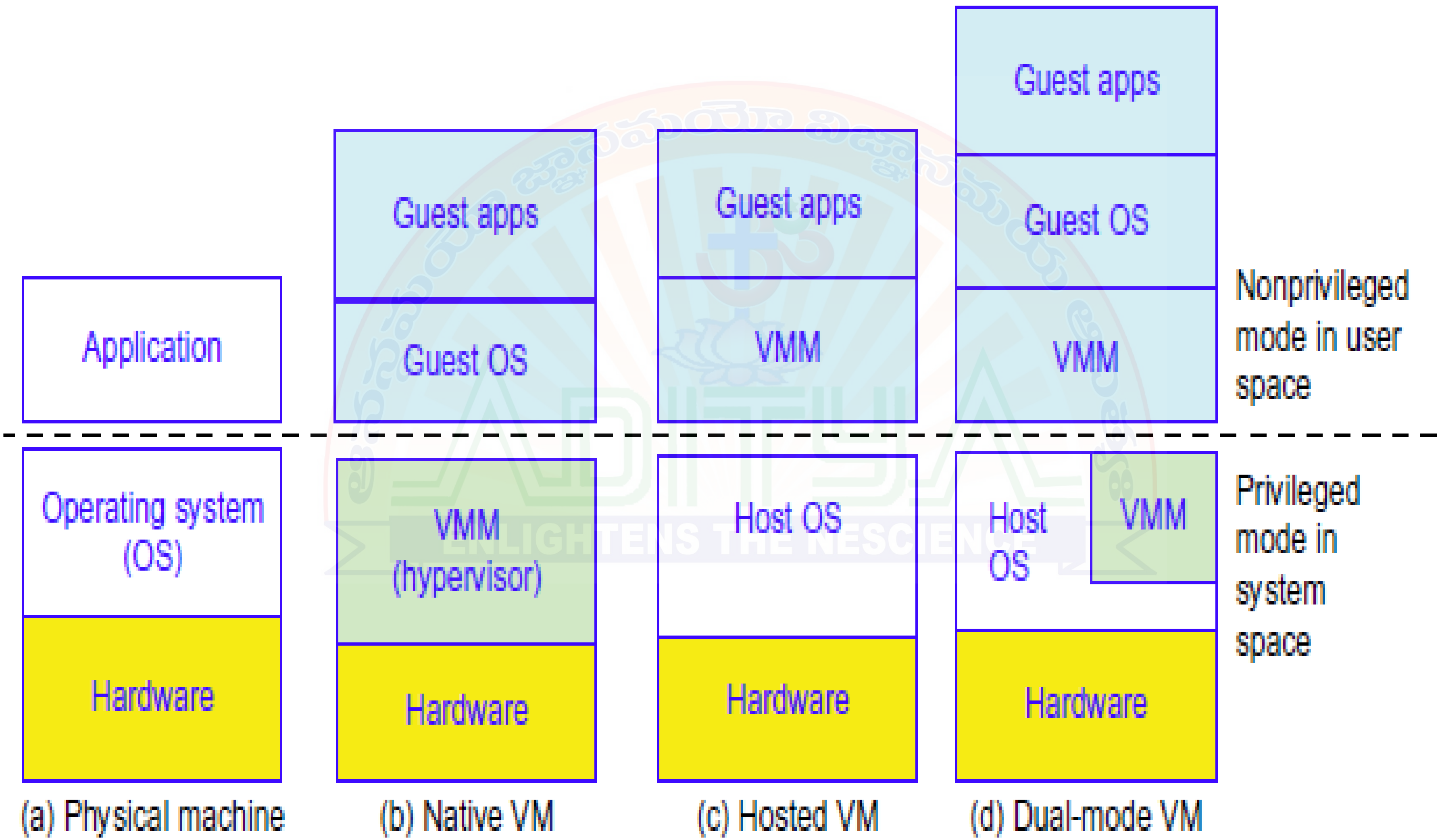


- SAN (storage area network) - connects servers with disk arrays
- LAN (local area network) – connects clients, hosts, and servers
- NAS (network attached storage) – connects clients with large storage systems

Virtual Machines(VM)

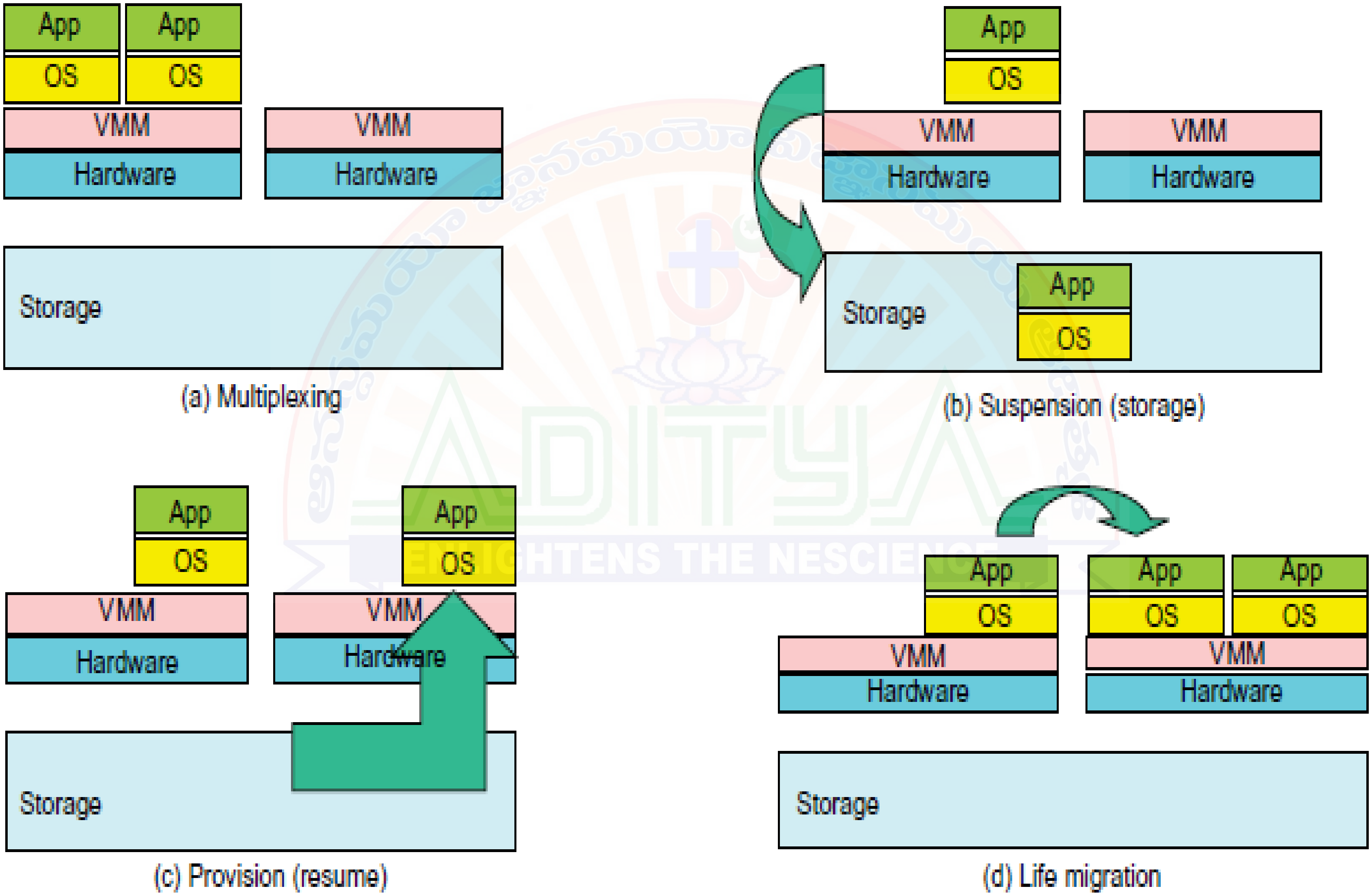
- Eliminate real machine constraint
  - Increases portability and flexibility
- Virtual machine adds software to a physical machine to give it the appearance of a different platform or multiple platforms.
- Benefits
  - Cross platform compatibility
  - Increase Security
  - Enhance Performance
  - Simplify software migration
- The VM can be provisioned for any hardware system. The VM is built with virtual resources managed by a guest OS to run a specific application. Between the VMs and the host platform, one needs to deploy a middleware layer called a virtual machine monitor (VMM).

Three VM Architectures

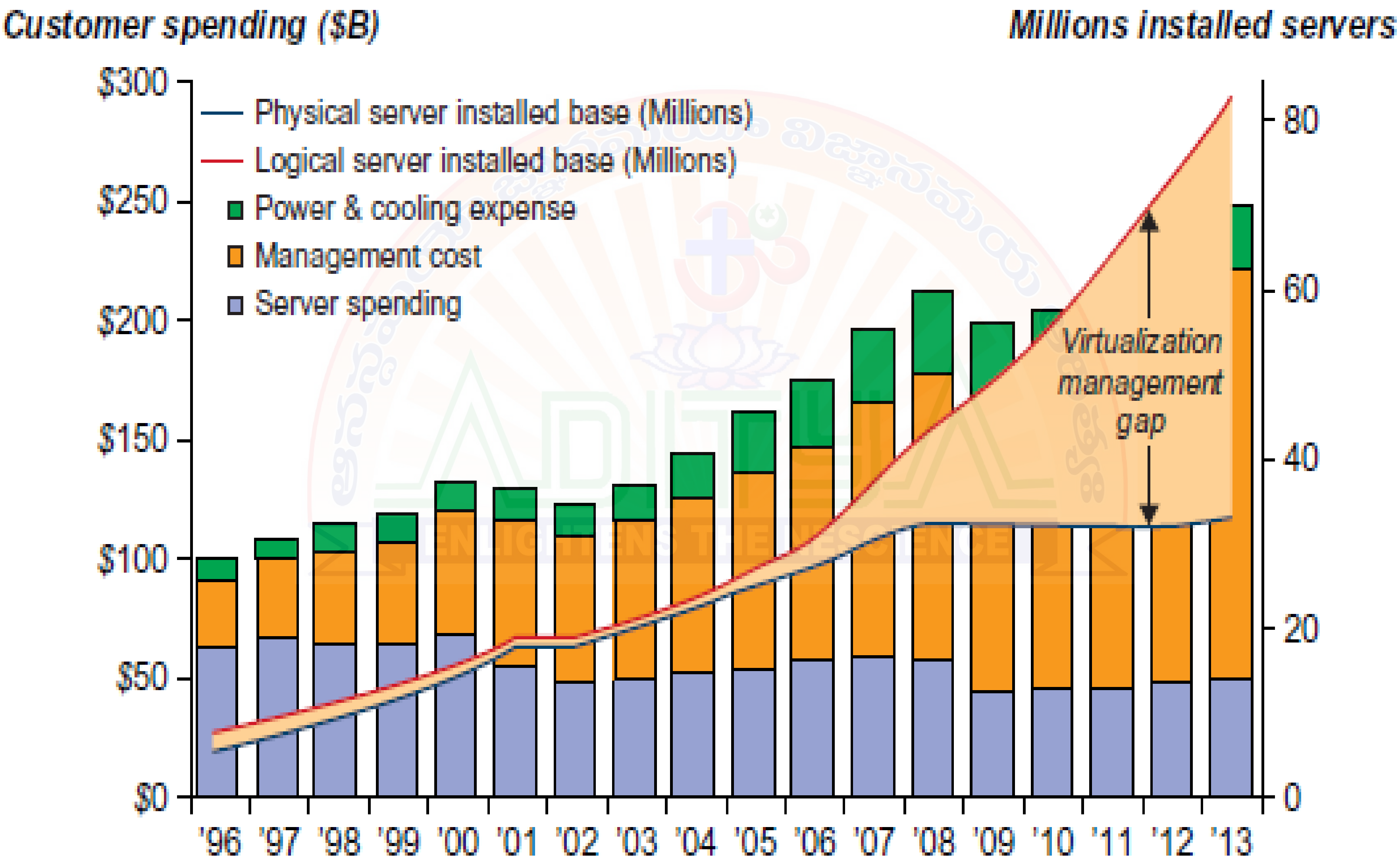


VM Primitive Operations

* Virtual Machine Monitor (VMM)



Virtual Infrastructure : Datacenter and Server Cost Distribution



Data Center Virtualization for Cloud Computing

- Cloud architecture is built with commodity hardware and network devices.
- All cloud platforms choose the popular x86 processors. Low-cost terabyte disks and Gigabit Ethernet are used to build data centers.
- Data center design emphasizes the performance/price ratio over speed performance alone.
- A large data center may be built with thousands of servers. Smaller data centers are typically built with hundreds of servers.

Data Center Virtualization for Cloud Computing

- The cost to build and maintain data center servers has increased over the years.
- According to a 2009 IDC report, typically the data centre costs are as follows
  - 30% - purchasing equipment (servers and disks),
  - 33% - attributed to the chiller,
  - 18% - power supply (UPS),
  - 9% - to computer room air conditioning
  - remaining - power distribution, lighting & transformer cost
- 60% of the cost to run a data center is allocated to management and maintenance.
- Using x86 servers, commodity switches and networks are more desirable for low cost data centers.

Convergence of Technologies

- Cloud computing is enabled by the convergence of technologies in four areas:
  - Hardware virtualization and multi-core chips,
  - utility and grid computing,
  - SOA, Web 2.0,
  - autonomic computing and data center automation.
- Hardware virtualization and multicore chips enable the existence of dynamic configurations in the cloud.
- Utility and grid computing technologies lay the necessary foundation for computing clouds.
- Recent advances in SOA, Web 2.0, are pushing the cloud another step forward.
- Autonomic computing and automated data center operations contribute to the rise of cloud computing.

System Models for Distributed and Cloud Computing

- Distributed & cloud computing systems are built over a large number of autonomous computer nodes.
- These node machines are interconnected by SANs, LANs, or WANs in a hierarchical manner.
- LAN switches can easily connect hundreds of machines as a working cluster.
- A WAN can connect many local clusters to form a very large cluster of clusters.
- A massive system with millions of computers connected to edge networks is build.
- Massive systems are classified into four groups:
  - Clusters,
  - P2P networks,
  - Computing grids,
  - Internet clouds

System Models for Distributed and Cloud Computing

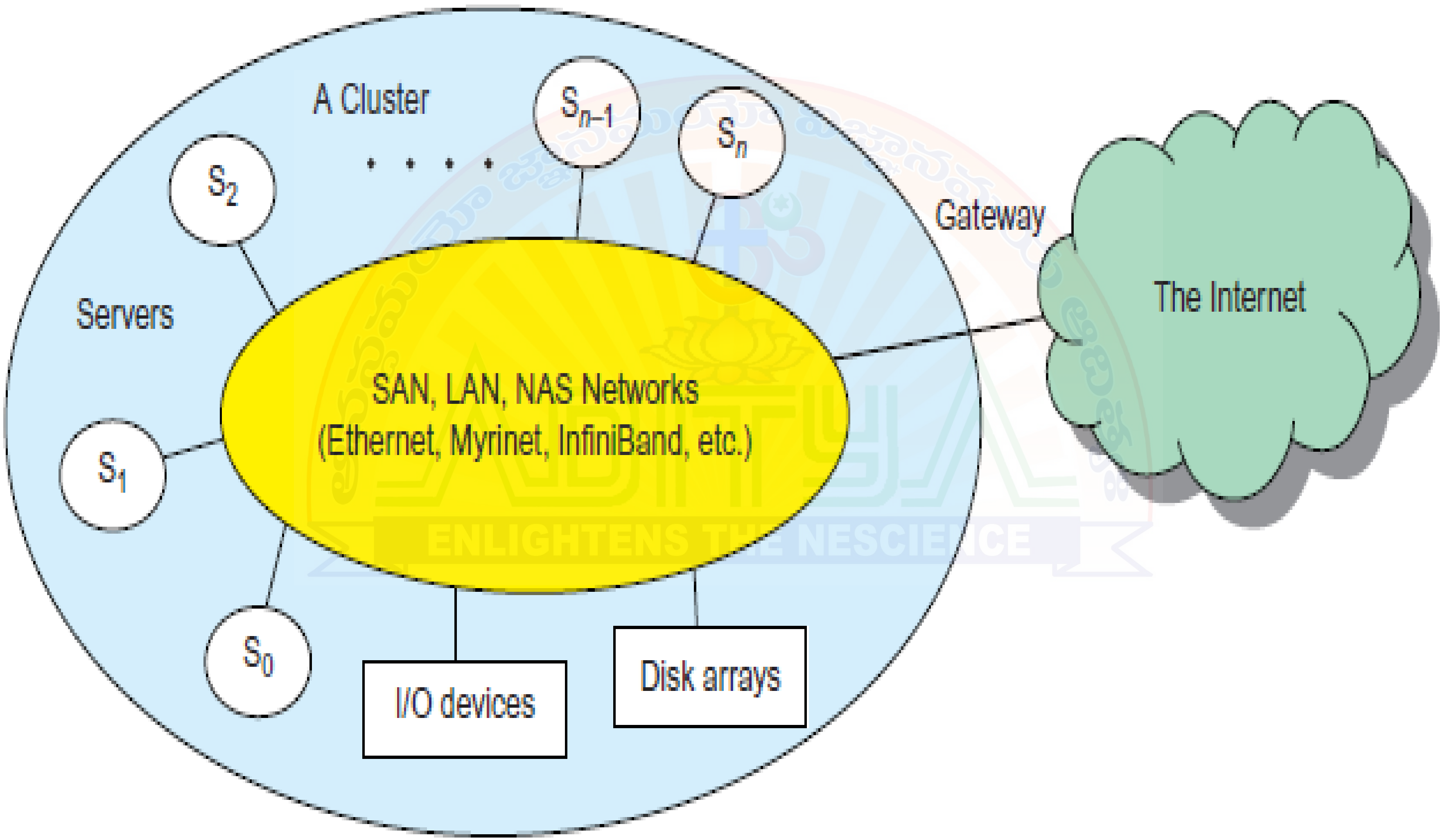
Table 1.2 Classification of Parallel and Distributed Computing Systems

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

Cluster Architecture

- A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.
- Can build scalable clusters with an increasing no. of nodes.
- The cluster is connected to the Internet via a virtual private network (VPN) gateway. The gateway IP address locates the cluster.
- The system image of a computer is decided by the way the OS manages the shared cluster resources.
- An ideal cluster should merge multiple system images into a single-system image (SSI).
- An SSI is an illusion created by software or hardware that presents a collection of resources as one integrated, powerful resource. SSI makes the cluster appear like a single machine to the user.

A Typical Cluster Architecture



Major Cluster Design Issues

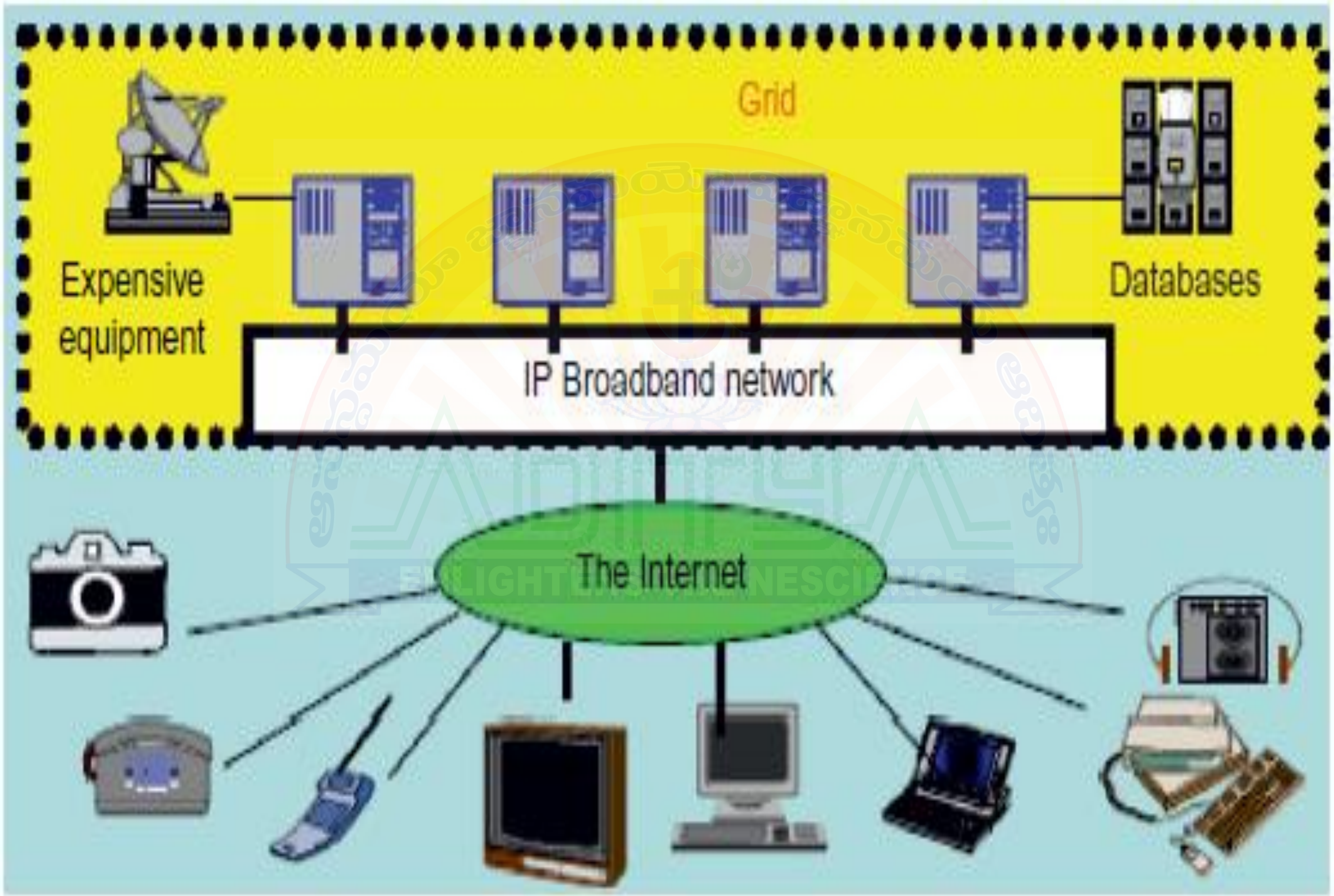
Table 1.3 Critical Cluster Design Issues and Feasible Implementations

Features	Functional Characterization	Feasible Implementations
Availability and Support	Hardware and software support for sustained HA in cluster	Failover, failback, check pointing, rollback recovery, nonstop OS, etc.
Hardware Fault Tolerance	Automated failure management to eliminate all single points of failure	Component redundancy, hot swapping, RAID, multiple power supplies, etc.
Single System Image (SSI)	Achieving SSI at functional level with hardware and software support, middleware, or OS extensions	Hardware mechanisms or middleware support to achieve DSM at coherent cache level
Efficient Communications	To reduce message-passing system overhead and hide latencies	Fast message passing, active messages, enhanced MPI library, etc.
Cluster-wide Job Management	Using a global job management system with better scheduling and monitoring	Application of single-job management systems such as LSF, Codine, etc.
Dynamic Load Balancing	Balancing the workload of all processing nodes along with failure recovery	Workload monitoring, process migration, job replication and gang scheduling, etc.
Scalability and Programmability	Adding more servers to a cluster or adding more clusters to a grid as the workload or data set increases	Use of scalable interconnect, performance monitoring, distributed execution environment, and better software tools

Computational Grids

- A computing grid offers an infrastructure that couples computers, software/middleware, special instruments, and people and sensors together.
- The grid is often constructed across LAN, WAN, or Internet backbone networks at a regional, national, or global scale.
- Enterprises or organizations present grids as integrated computing resources.
- They can also be viewed as virtual platforms to support virtual organizations.
- The computers used in a grid are primarily workstations, servers, clusters, and supercomputers.
- Personal computers, laptops, and PDAs can be used as access devices to a grid system.

Computational or Data Grid



Grid Families

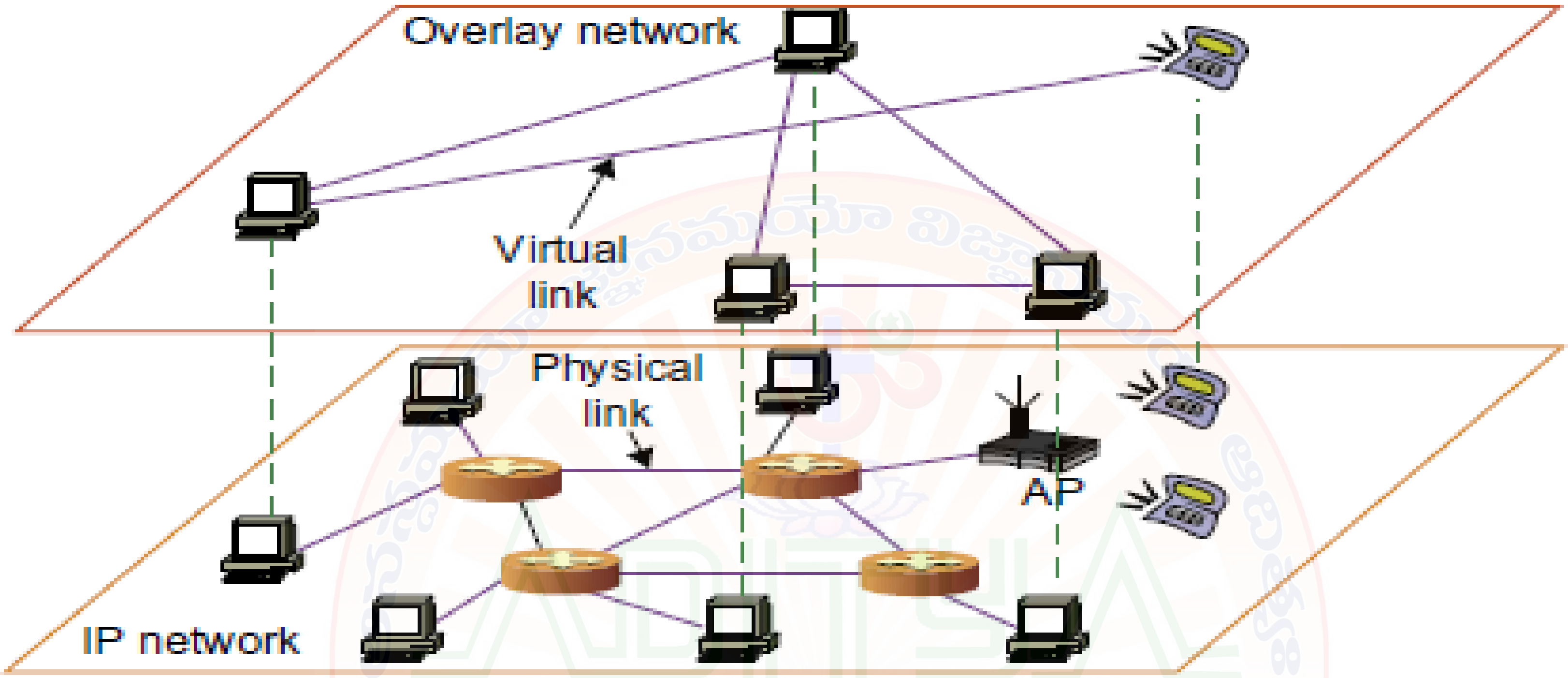
Table 1.4 Two Grid Computing Infrastructures and Representative Systems

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

Peer-to-Peer (P2P) Network

- A distributed system architecture
- Each computer in the network can act as a client or server for other network computers.
- No centralized control
- Typically many nodes, but unreliable and heterogeneous
- Nodes are symmetric in function
- Take advantage of distributed, shared resources (bandwidth, CPU, storage) on peer-nodes
- Fault-tolerant, self-organizing
- Operate in dynamic environment, frequent join and leave the network.

Peer-to-Peer (P2P) Network Representation



Overlay network - computer network built on top of another network.

- Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network.
- For example, distributed systems such as cloud computing, peer-to-peer networks, and client-server applications are overlay networks because their nodes run on top of the Internet.

P2P Application Families

Table 1.5 Major Categories of P2P Network Families [42]

System Features	Distributed File Sharing	Collaborative Platform	Distributed P2P Computing	P2P Platform
Attractive Applications	Content distribution of MP3 music, video, open software, etc.	Instant messaging, collaborative design and gaming	Scientific exploration and social networking	Open networks for public resources
Operational Problems	Loose security and serious online copyright violations	Lack of trust, disturbed by spam, privacy, and peer collusion	Security holes, selfish partners, and peer collusion	Lack of standards or protection protocols
Example Systems	Gnutella, Napster, eMule, BitTorrent, Aimster, KaZaA, etc.	ICQ, AIM, Groove, Magi, Multiplayer Games, Skype, etc.	SETI@home, Geonome@home, etc.	JXTA, .NET, FightingAid@home, etc.

The Cloud

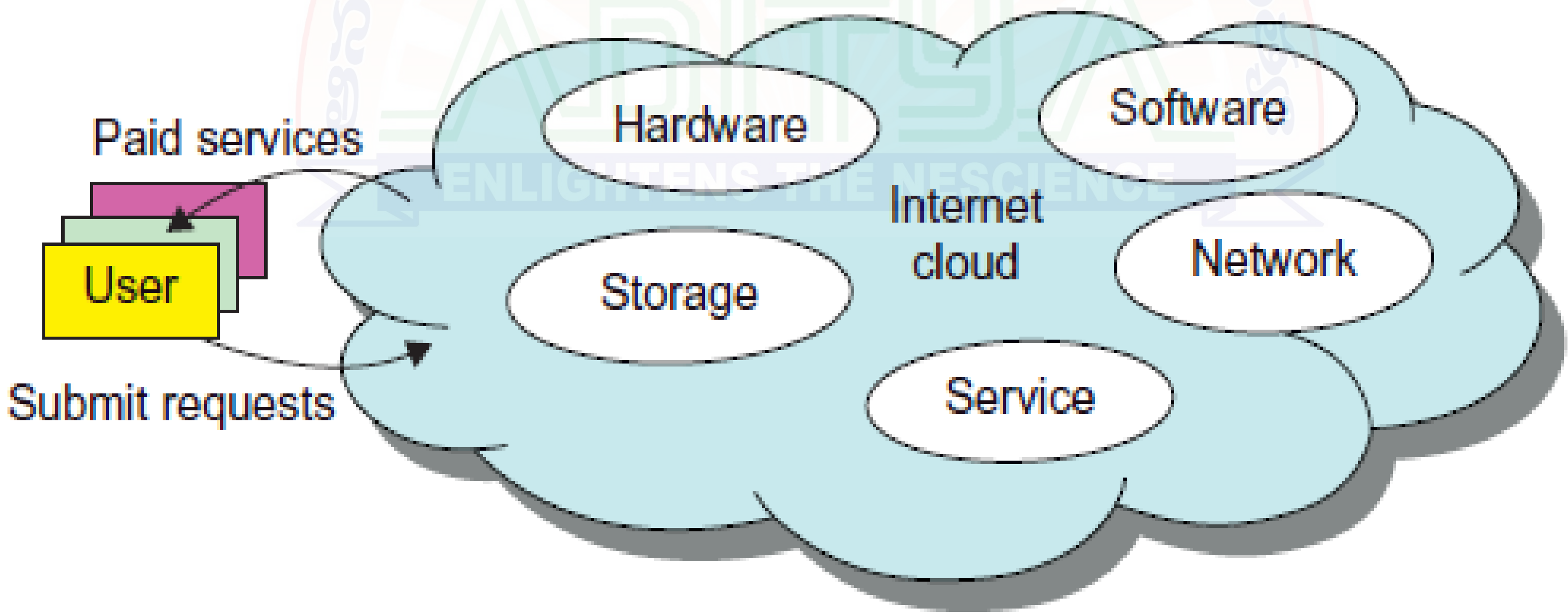
- Historical roots in today's Internet apps
    - Search, email, social networks
    - File storage (Live Mesh, Mobile Me, Flicker, …)
- A cloud infrastructure provides a framework to manage scalable, reliable, on-demand access to applications
- A cloud is the "invisible" backend to many of our mobile applications
- A model of computation and data storage based on "pay as you go" access to "unlimited" remote data center capabilities

The Cloud

- IBM's definition: "A cloud is a pool of virtualized computer resources. A cloud can host a variety of different workloads, including batch-style backend jobs and interactive and user-facing applications."
- Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically .
- The idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at data centers.
- Cloud computing leverages its low cost and simplicity to benefit both users and providers.
- Machine virtualization has enabled such cost-effectiveness. Cloud computing intends to satisfy many user applications simultaneously. The cloud ecosystem must be designed to be secure, trustworthy, and dependable.

Basic Concept of Internet Clouds

- Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet).
- The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams.
- Cloud computing entrusts remote services with a user's data, software and computation.



The Next Revolution in IT
Cloud Computing

- **Classical Computing**
  - Buy & Own
    - Hardware, System Software, Applications often to meet peak needs.
  - Install, Configure, Test, Verify, Evaluate
  - Manage
  - ..
  - Finally, use it
  - $$$$....$(High CapEx)

Every 18 months?

- **Cloud Computing**
  - Subscribe
  - Use

  - $ - pay for what you use, based on QoS

Cloud Service Models

Infrastructure as a service (IaaS)

- Most basic cloud service model
- Cloud providers offer servers, storage, networks, data center fabric and other resources.
- Virtual machines are run as guests by a hypervisor, such as Xen or KVM.
- Cloud users deploy their applications by then installing operating system images on the machines as well as their application software.
- Users do not manage or control the underlying cloud infrastructure.
- Cloud providers typically bill IaaS services on a utility computing basis, that is, cost will reflect the amount of resources allocated and consumed.
- Examples of IaaS include: Amazon Cloud Formation (and underlying services such as Amazon EC2), Rackspace Cloud, Terremark, and Google Compute Engine.

Cloud Service Models

Platform as a service (PaaS)

- Cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server.
- Application developers develop and run their software on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.
- Examples of PaaS include: Amazon Elastic Beanstalk, Cloud Foundry, Heroku, Force.com, EngineYard, Mendix, Google App Engine, Microsoft Azure and OrangeScape.

Cloud Service Models

Software as a service (SaaS)

- Cloud providers install and operate application software in the cloud and cloud users access the browser –initiated application software from cloud.
- The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point.
- The SaaS model applies to business processes, industry applications, consumer relationship management (CRM), enterprise resources planning (ERP), human resources (HR), and collaborative applications
- Examples of SaaS include: Google Apps, innkeypos, Quickbooks Online, Limelight Video Platform, Salesforce.com, and Microsoft Office 365.

The Cloud Landscape

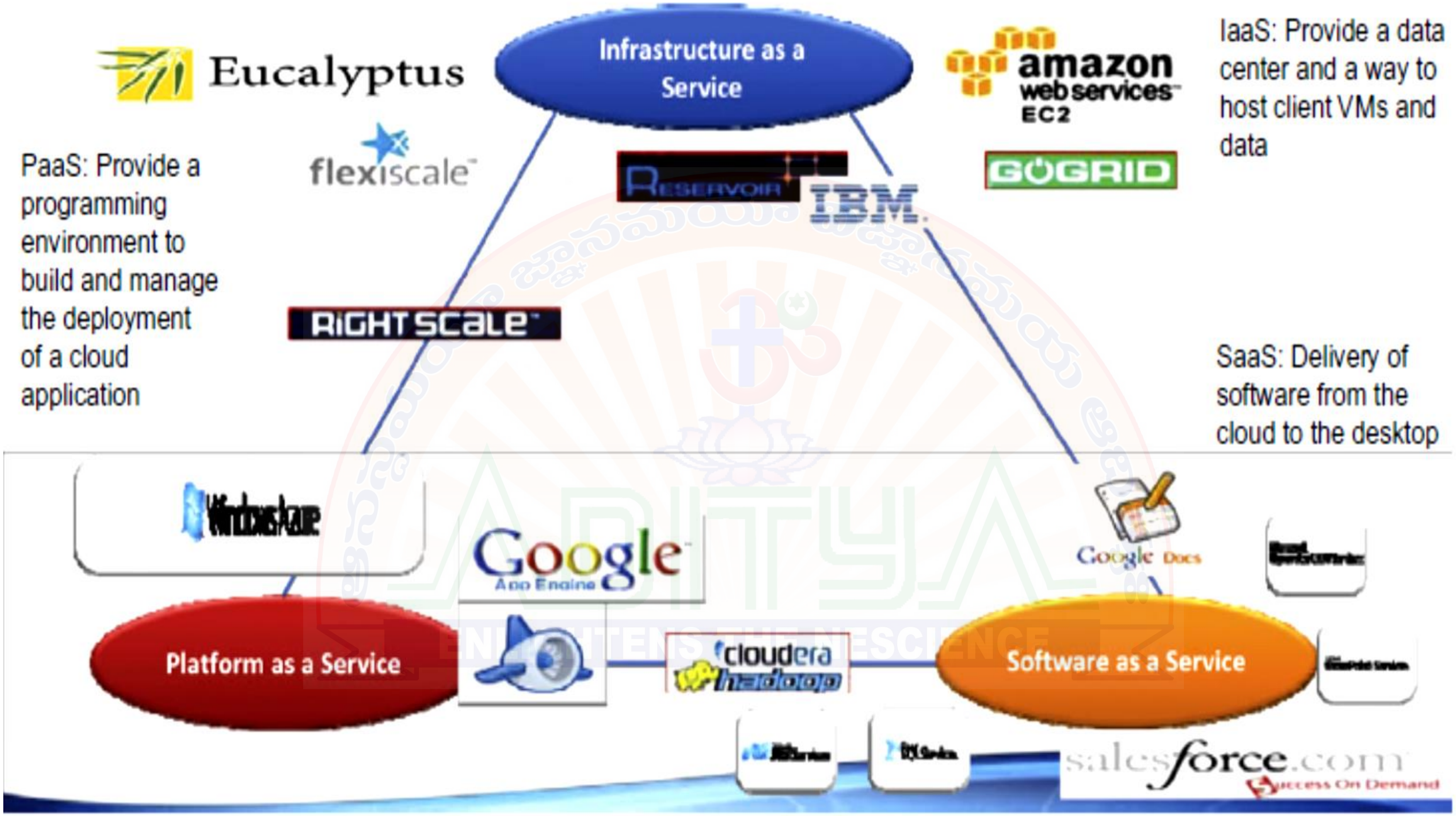


FIGURE 1.19

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

- Internet clouds offer four deployment modes: private, public, managed, and hybrid.
- These modes demand different levels of security implications. The different SLAs imply the security.
- Eight reasons to adapt the cloud for upgraded Internet applications and web services:

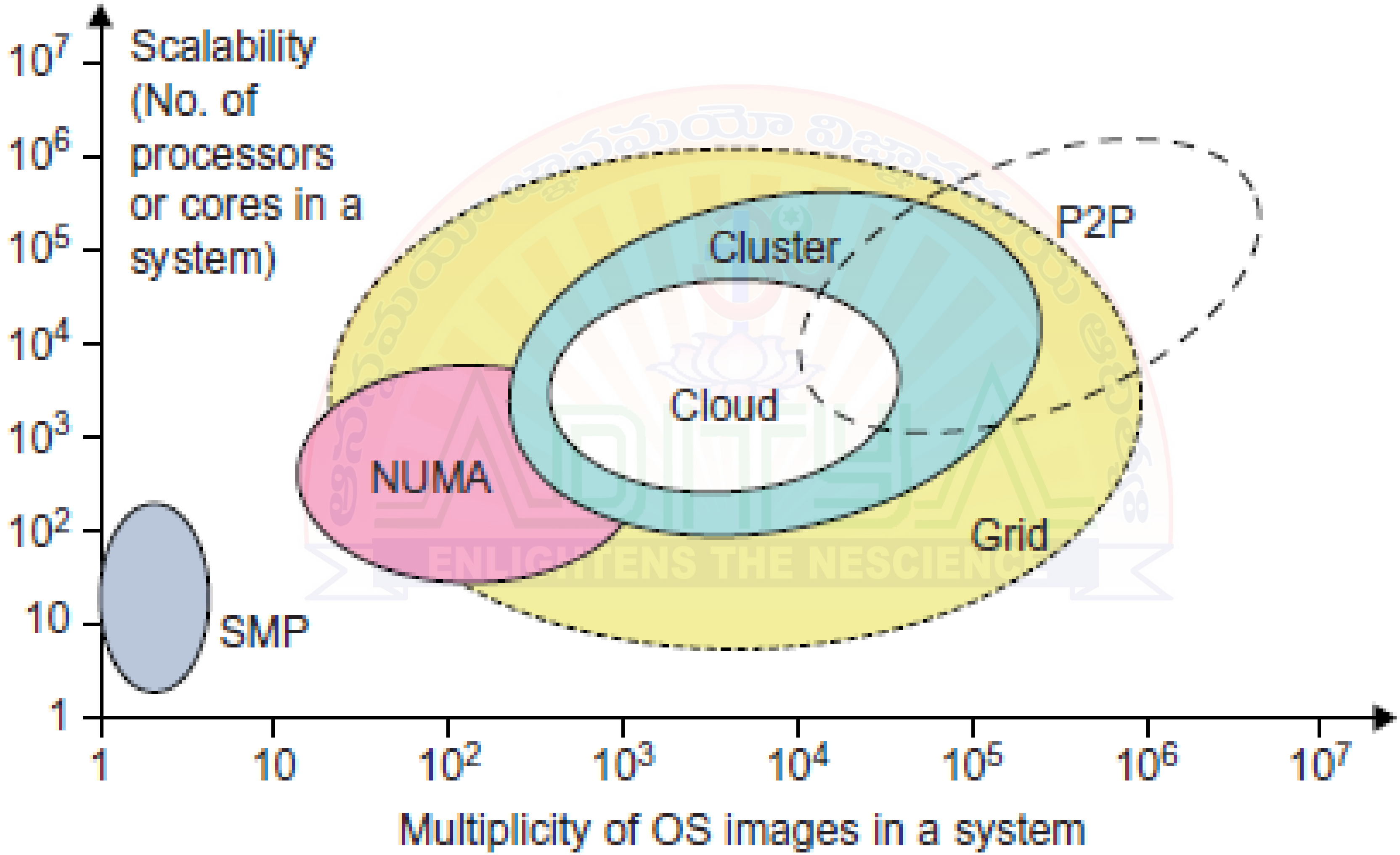
1. Desired location in areas with protected space and **higher energy efficiency**
2. **Sharing of peak-load** capacity among a large pool of users, improving overall utilization
3. Separation of infrastructure **maintenance** duties from domain-specific application development
4. Significant **reduction in cloud computing cost**, compared with traditional computing paradigms
5. Cloud computing programming and application development
6. Service and data discovery and **content/service distribution**
7. **Privacy, security,** copyright, and reliability issues
8. **Service agreements**, business models, and **pricing policies**

Performance, Security, and Energy Efficiency

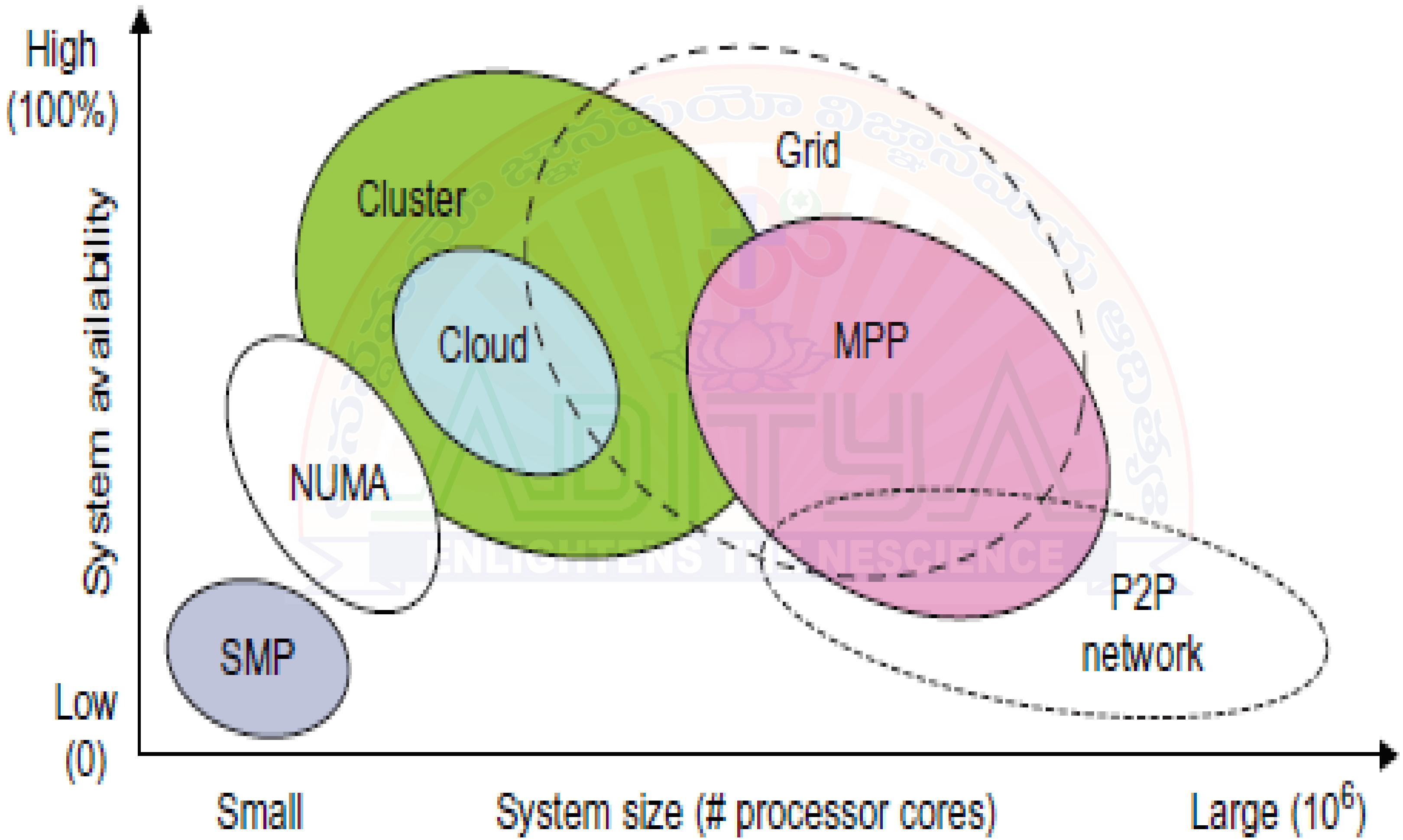
Dimensions of Scalability

- **Size** – increasing performance by increasing machine size
- **Software** – upgrade to OS, libraries, new apps.
- **Application** – matching problem size with machine size
- **Technology** – adapting system to new technologies

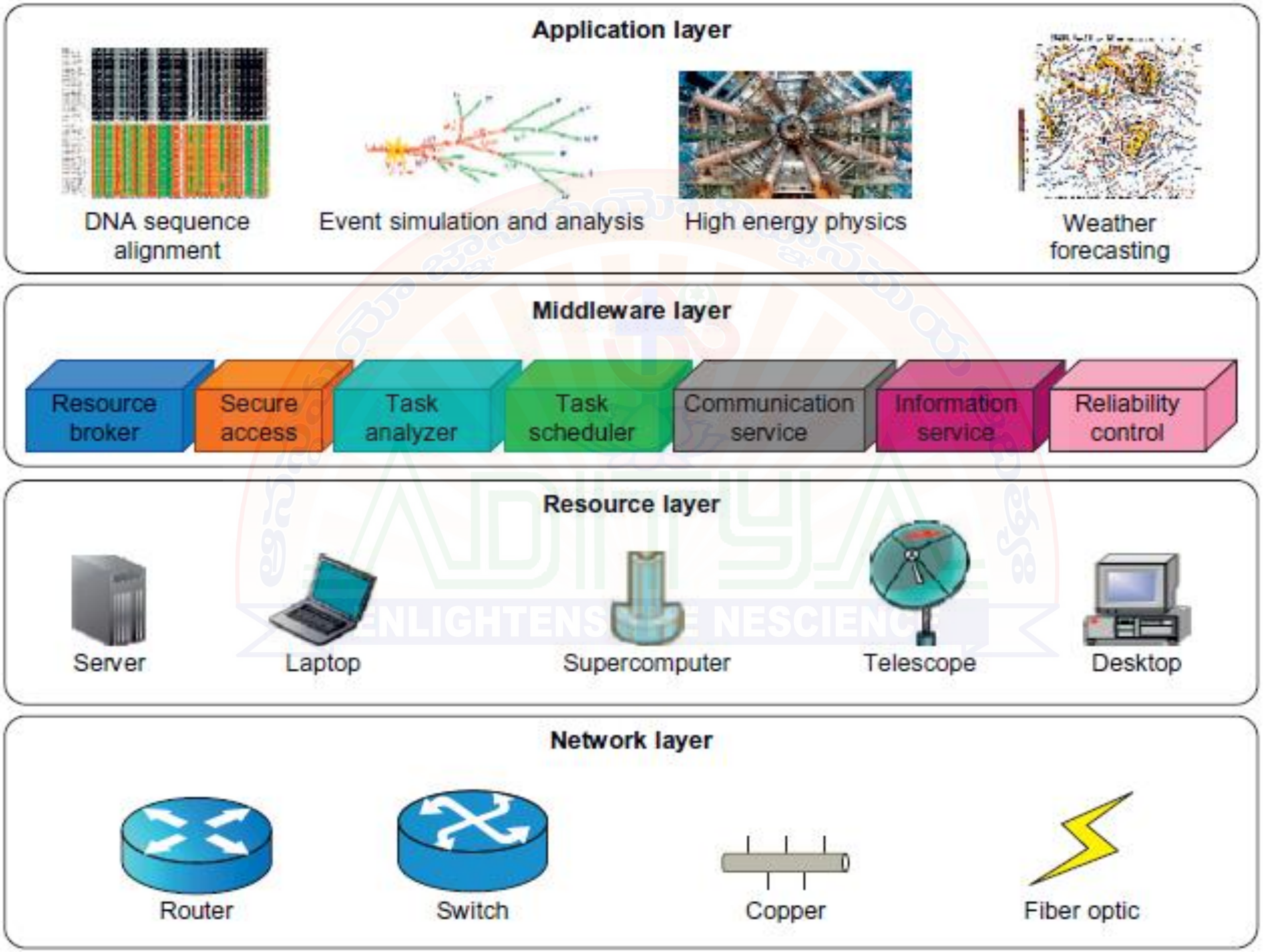
System Scalability vs. OS Multiplicity



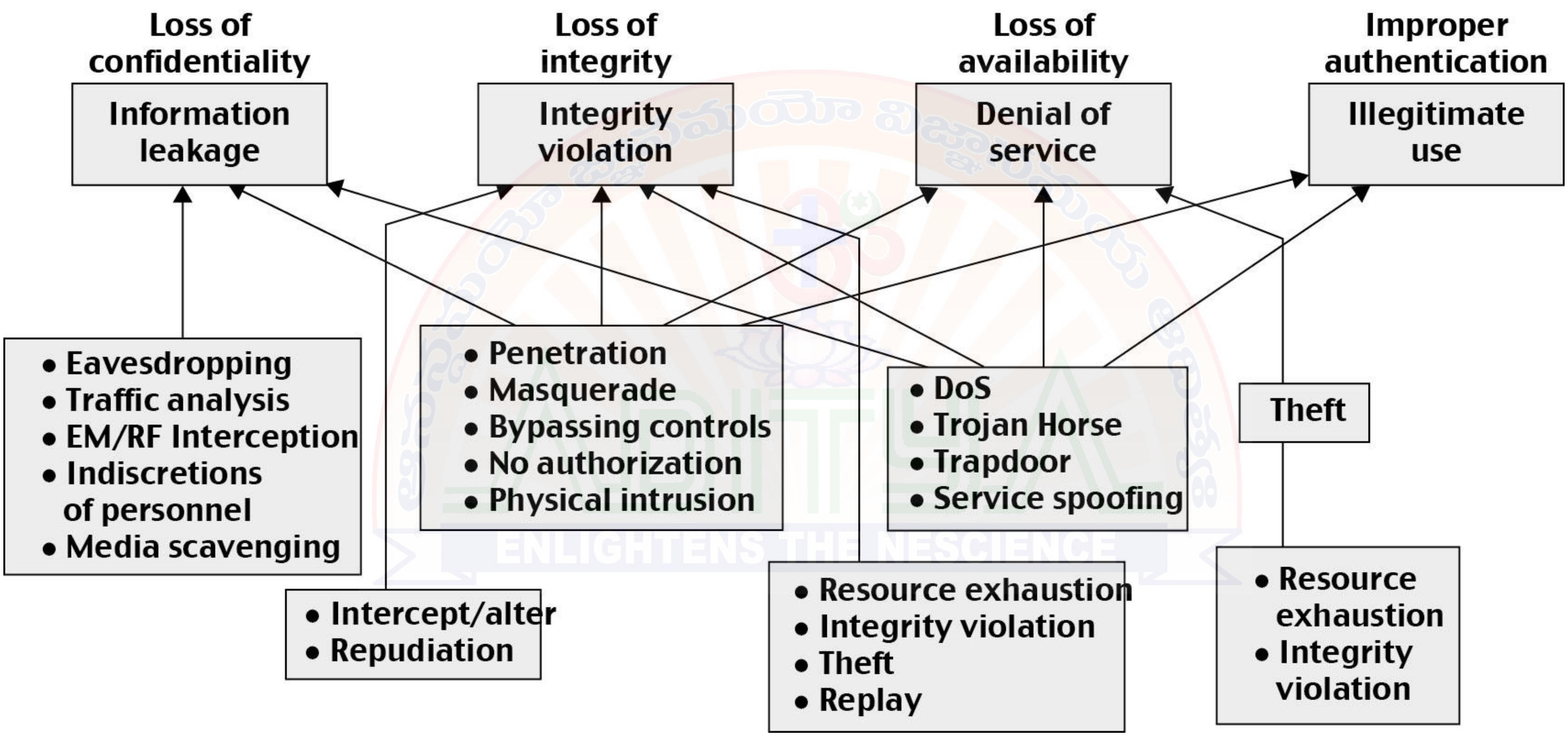
System Availability vs. Configuration Size



Operational Layers of Distributed Computing System



Security: System Attacks and Network Threats



Energy Efficiency in Distributed Computing

- Primary performance goals in conventional parallel and distributed computing systems are high performance and high throughput, considering some form of performance reliability.
- Energy consumption in parallel and distributed computing systems raises various monetary, environmental, and system performance issues.
- To run a server farm (data center) a company has to spend a huge amount of money for hardware, software, operational support, and energy every year.
- In addition to identifying unused/underutilized servers for energy savings, it is also necessary to apply appropriate techniques to decrease energy consumption in active distributed systems with negligible influence on their performance.
- By introducing energy-aware applications, the challenge is to design sophisticated multilevel and multi-domain energy management applications without hurting performance.
- It is also responsible for applying energy-efficient techniques, particularly in task scheduling in middleware layer.
- In the recent past, several mechanisms have been developed for more efficient power management of hardware, operating systems and processors like Dynamic power management (DPM) and dynamic voltage-frequency scaling (DVFS).

UNIT-II
Virtual Machines and Virtualization of Clusters and Data Centers

Virtualization for Datacenter Automation

to serve millions of clients, simultaneously

- Resource Sharing, Low cost, Application Flexibility
- Server Consolidation in Virtualized Datacenter
- Virtual Storage Provisioning and Deprovisioning
- Cloud Operating Systems for Virtual Datacenters
- Trust Management in virtualized Datacenters

Virtual Machine, Guest Operating System, and VMM (Virtual Machine Monitor) :

Virtual Machine

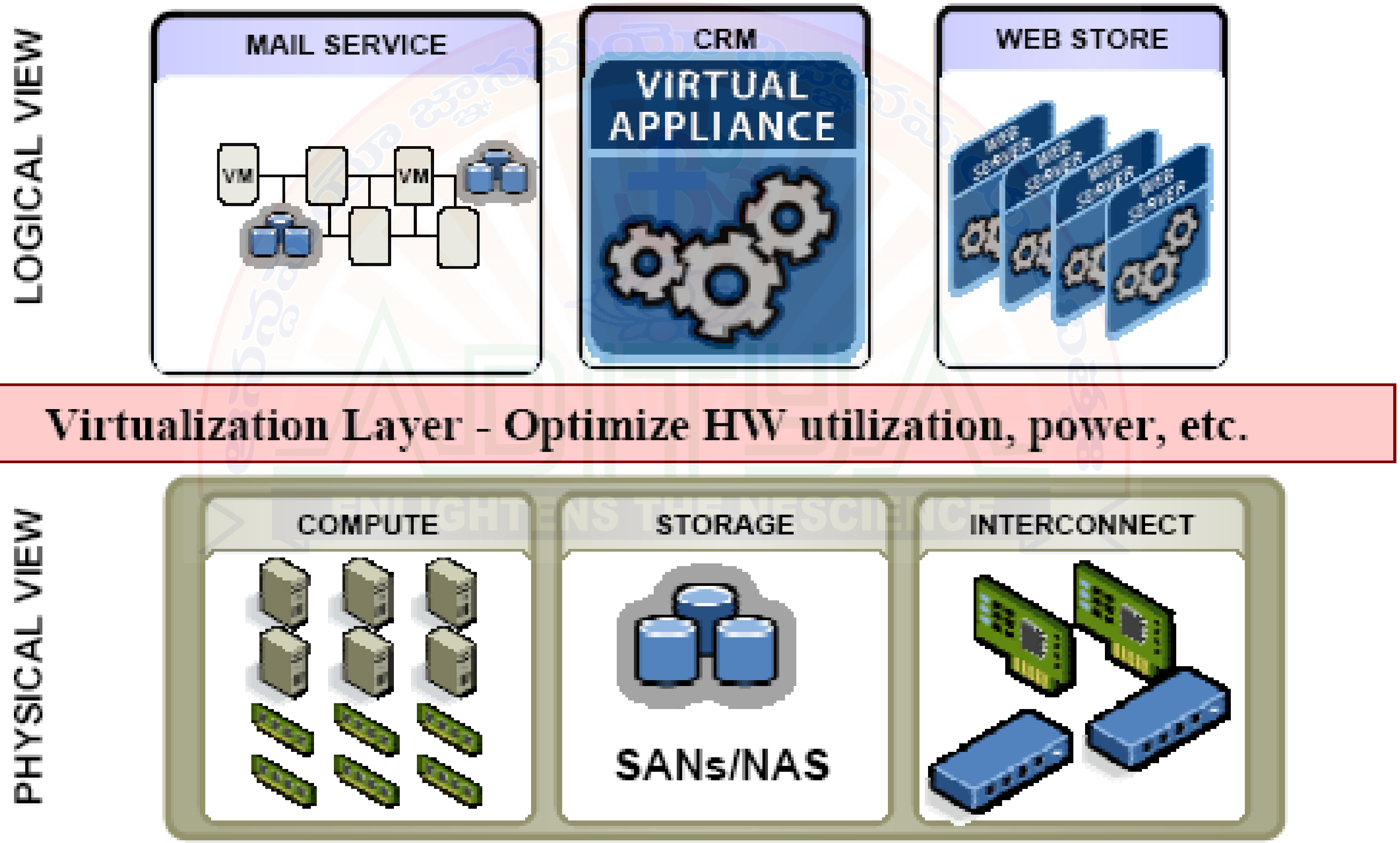
A representation of a real machine using software that provides an operating environment which can run or host a guest operating system.

Guest Operating System

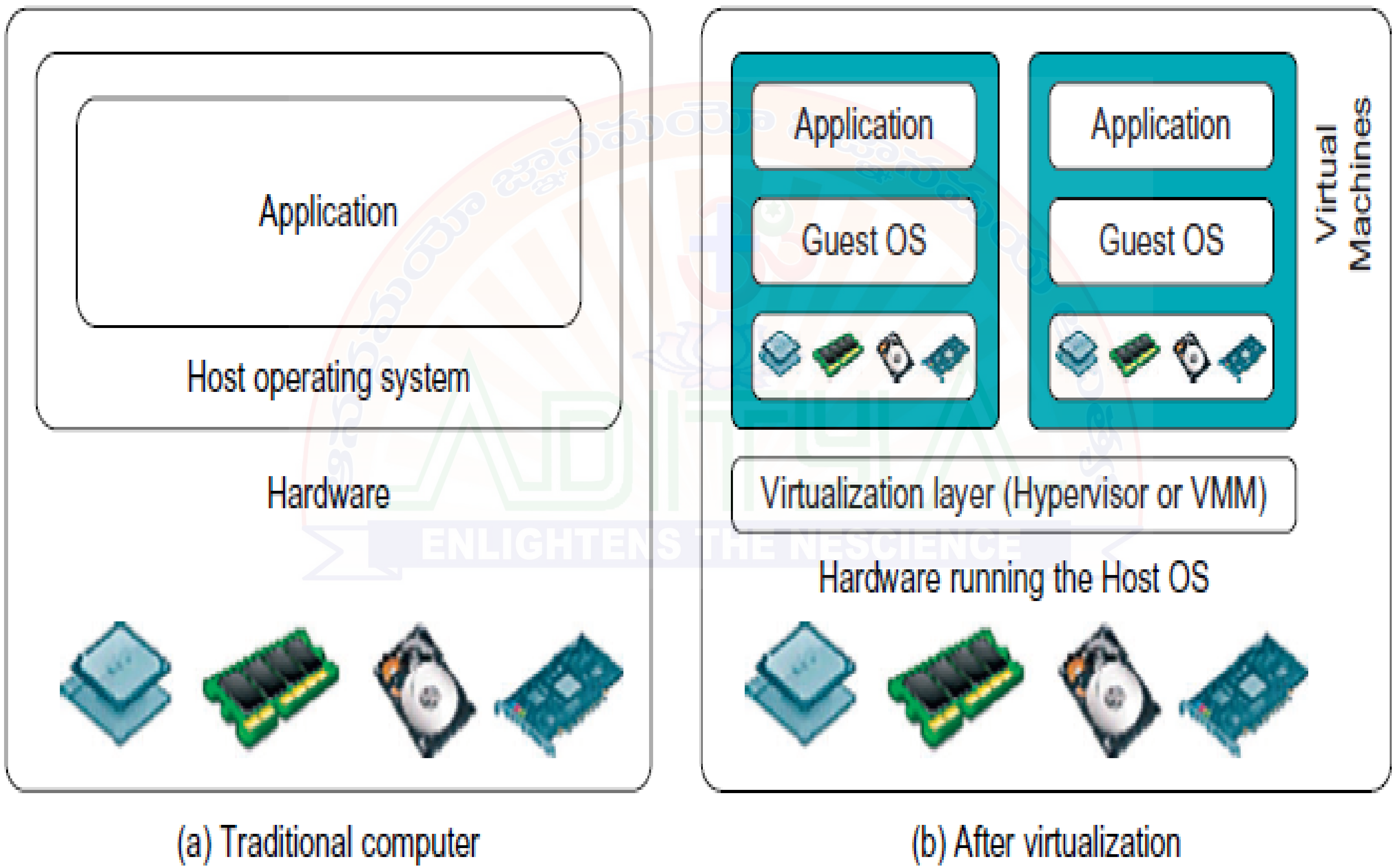
An operating system running in a virtual machine environment that would otherwise run directly on a separate physical system.

The Virtualization layer is the middleware between the underlying hardware and virtual machines represented in the system, also known as ***Virtual Machine Monitor*** (**VMM**) or ***Hypervisor***.

User’s view of virtualization



Difference between Traditional Computer and Virtual machines



Virtualization Ranging from Hardware to Applications in Five Abstraction Levels

Application level

JVM / .NET CLR / Panot

Library (user-level API) level

WINE/ WABI/ LxRun / Visual MainWin / vCUDA

Operating system level

Jail / Virtual Environment / Ensim's VPS / FVM

Hardware abstraction layer (HAL) level

VMware / Virtual PC / Denali / Xen / L4 /
Plex 86 / User mode Linux / Cooperative Linux

Instruction set architecture (ISA) level

Bochs / Crusoe / QEMU / BIRD / Dynamo

Virtualization at ISA (Instruction Set Architecture) level:

Emulating a given ISA by the ISA of the host machine.

- e.g, MIPS binary code can run on an x-86-based host machine with the help of ISA emulation.
  - Typical systems: Bochs, Crusoe, Quemu, BIRD, Dynamo

Advantage:

- It can run a large amount of legacy binary codes written for various processors on any given new hardware host machines
- best application flexibility

Shortcoming & limitation:

- One source instruction may require tens or hundreds of native target instructions to perform its function, which is relatively slow.
- V-ISA requires adding a processor-specific software translation layer in the complier.

Virtualization at Hardware Abstraction level:

Virtualization is performed right on top of the hardware. It generates virtual hardware environments for VMs, and manages the underlying hardware through virtualization.

Typical systems: VMware, Virtual PC, Denali, Xen

Advantage:

Has higher performance and good application isolation

Shortcoming & limitation:

Very expensive to implement (complexity)

Virtualization at Operating System (OS) level:

It is an abstraction layer between traditional OS and user Applications. This virtualization creates isolated containers on a single physical server and the OS-instance to utilize the hardware and software in datacenters.

Typical systems: Jail / Virtual Environment / Ensim's VPS / FVM

Advantage:

Has minimal starup/shutdown cost, low resource requirement, and high scalability; synchronize VM and host state changes.

Shortcoming & limitation:

All VMs at the operating system level must have the same kind of guest OS Poor application flexibility and isolation.

Library Support level:

It creates execution environments for running alien programs on a platform rather than creating VM to run the entire operating system. It is done by API call interception and remapping.

Typical systems: Wine, WAB, LxRun , VisualMainWin

Advantage:

It has very low implementation effort

Shortcoming & limitation:

poor application flexibility and isolation

User-Application level:

It virtualizes an application as a virtual machine. This layer sits as an application program on top of an operating system and exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.

Typical systems: JVM , NET CLI , Panot

Advantage:

has the best application isolation

Shortcoming & limitation:

low performance, low application flexibility and high implementation complexity.

Relative Merits of Virtualization at Various Levels

Table 3.1 Relative Merits of Virtualization at Various Levels

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

More Xs mean higher merit

Hypervisor

A hypervisor is a hardware virtualization technique allowing multiple operating systems, called guests to run on a host machine. This is also called the Virtual Machine Monitor (VMM).

Type 1: Bare Metal Hypervisor

sits on the bare metal computer hardware like the CPU, memory, etc.

All guest operating systems are a layer above the hypervisor.

The original CP/CMS hypervisor developed by IBM was of this kind.

Type 2: Hosted Hypervisor

Run over a host operating system.

Hypervisor is the second layer over the hardware.

Guest operating systems run a layer over the hypervisor.

The OS is usually unaware of the virtualization

Hypervisor

There are three requirements for a **VMM**.

1) A VMM should provide an environment for programs which is essentially identical to the original machine.
2) Programs run in this environment should show, at worst, only minor decreases in speed.
3) A VMM should be in complete control of the system resources.

Any program run under a VMM should exhibit a function identical to that which it runs on the original machine directly.

Hypervisor

Two possible exceptions in terms of differences are permitted with this requirement:

1) differences caused by the availability of system resources

2) differences caused by timing dependencies.

Complete control of these resources by a VMM includes 3 issues:

The VMM is responsible for allocating hardware resources for programs;

It is not possible for a program to access any resource not explicitly allocated to it;

It is possible under certain circumstances for a VMM to regain control of resources already allocated.

Major VMM and Hypervisor Providers

VMM Provider	Host CPU	Guest CPU	Host OS	Guest OS	VM Architecture
VMware Work-station	X86, x86-64	X86, x86-64	Windows, Linux	Windows, Linux, Solaris, FreeBSD, Netware, OS/2, SCO, BeOS, Darwin	Full Virtualization
VMware ESX Server	X86, x86-64	X86, x86-64	No host OS	The same as VMware workstation	Para-Virtualization
XEN	X86, x86-64, IA-64	X86, x86-64, IA-64	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, windows XP and 2003 Server	Hypervisor
KVM	X86, x86-64, IA64, S390, PowerPC	X86, x86-64, IA64, S390, PowerPC	Linux	Linux, Windows, FreeBSD, Solaris	Para-Virtualization

Virtualization support at the OS Level

With the help of VM technology, a new computing mode known as cloud computing is emerging.

Cloud computing is transforming the computing landscape by shifting the hardware and staffing costs of managing a computational center to third parties, just like banks.

However, cloud computing has at least **two challenges**.

The **first** is the ability to use a variable number of physical machines and VM instances depending on the needs of a problem.

The **second** challenge concerns the slow operation of instantiating new VMs.

Why OS- Level Virtualization ?

- It is slow to initialize a hardware-level VM because each VM creates its own image from scratch.
- To reduce the performance overhead of hardware-level virtualization, OS-level virtualization is used.
- OS virtualization inserts a virtualization layer inside an operating system to partition a machine's physical resources.
- It enables multiple isolated VMs within a single operating system kernel.
- This kind of VM is often called a Virtual execution Environment (VE), Virtual Private System (VPS), or simply container.
- From the user's point of view, VEs look like real servers.
- This means a VE has its own set of processes, file system, user accounts, network interfaces with IP addresses, routing tables, firewall rules, and other personal settings.

Advantages of OS Extensions

- The benefits of OS extensions are twofold:

(1) VMs at the OS level have minimal startup/shutdown costs, low resource requirements, and high scalability

(2) It is possible for a VM and its host environment to synchronize state changes when necessary.

Disadvantages of OS Extensions

(1) The main disadvantage of OS extensions is that all the VMs at operating system level on a single container must have the same kind of guest operating system.

Virtualization Support for the Linux Platform

- OpenVZ is an OS-level tool designed to support Linux platforms to create virtual environments for running VMs under different guest OSes.
- OpenVZ is an open source container-based virtualization solution built on Linux.
- To support virtualization and isolation of various subsystems, limited resource management, and check pointing, OpenVZ modifies the Linux kernel.

OS LEVEL VIRTUALIZATION

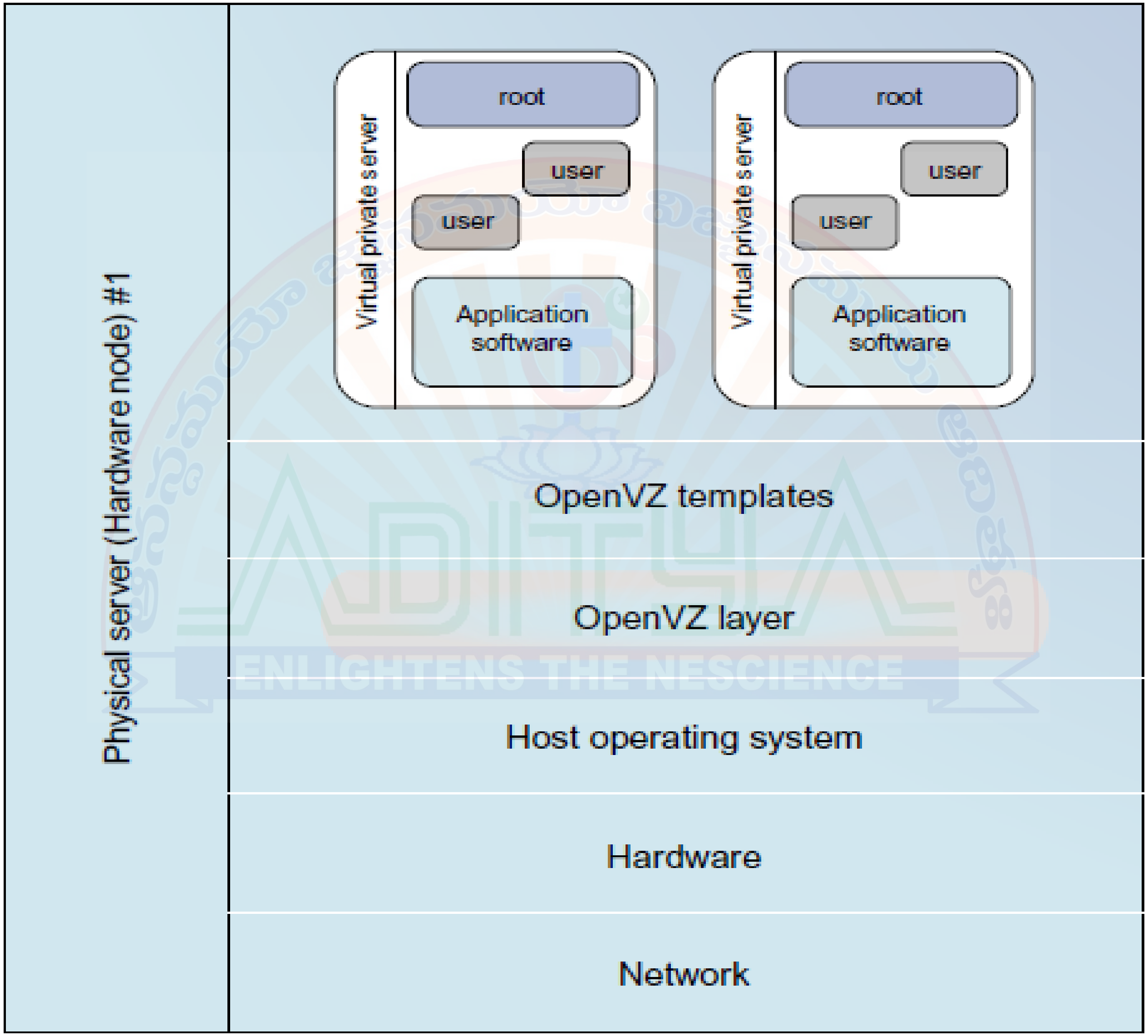


Table 3.3 Virtualization Support for Linux and Windows NT Platforms

Virtualization Support and Source of Information	Brief Introduction on Functionality and Application Platforms
Linux vServer for Linux platforms (http://linux-vserver.org/)	Extends Linux kernels to implement a security mechanism to help build VMs by setting resource limits and file attributes and changing the root environment for VM isolation
OpenVZ for Linux platforms [65]; http://ftp.openvz.org/doc/OpenVZ-Users-Guide.pdf)	Supports virtualization by creating *virtual private servers (VPSes)*; the VPS has its own files, users, process tree, and virtual devices, which can be isolated from other VPSes, and checkpointing and live migration are supported
FVM (Feather-Weight Virtual Machines) for virtualizing the Windows NT platforms [78])	Uses system call interfaces to create VMs at the NY kernel space; multiple VMs are supported by virtualized namespace and copy-on-write

Virtualization with Middleware/Library Support

- Library-level virtualization is also known as user-level Application Binary Interface (ABI) or API emulation.
- This type of virtualization can create execution environments for running alien programs on a platform rather than creating a VM to run the entire operating system.
- API call interception and remapping are the key functions performed.
- Eg: Windows Application Binary Interface (WABI), lxrun, WINE, Visual MainWin, and vCUDA

Virtualization with Middleware/Library Support

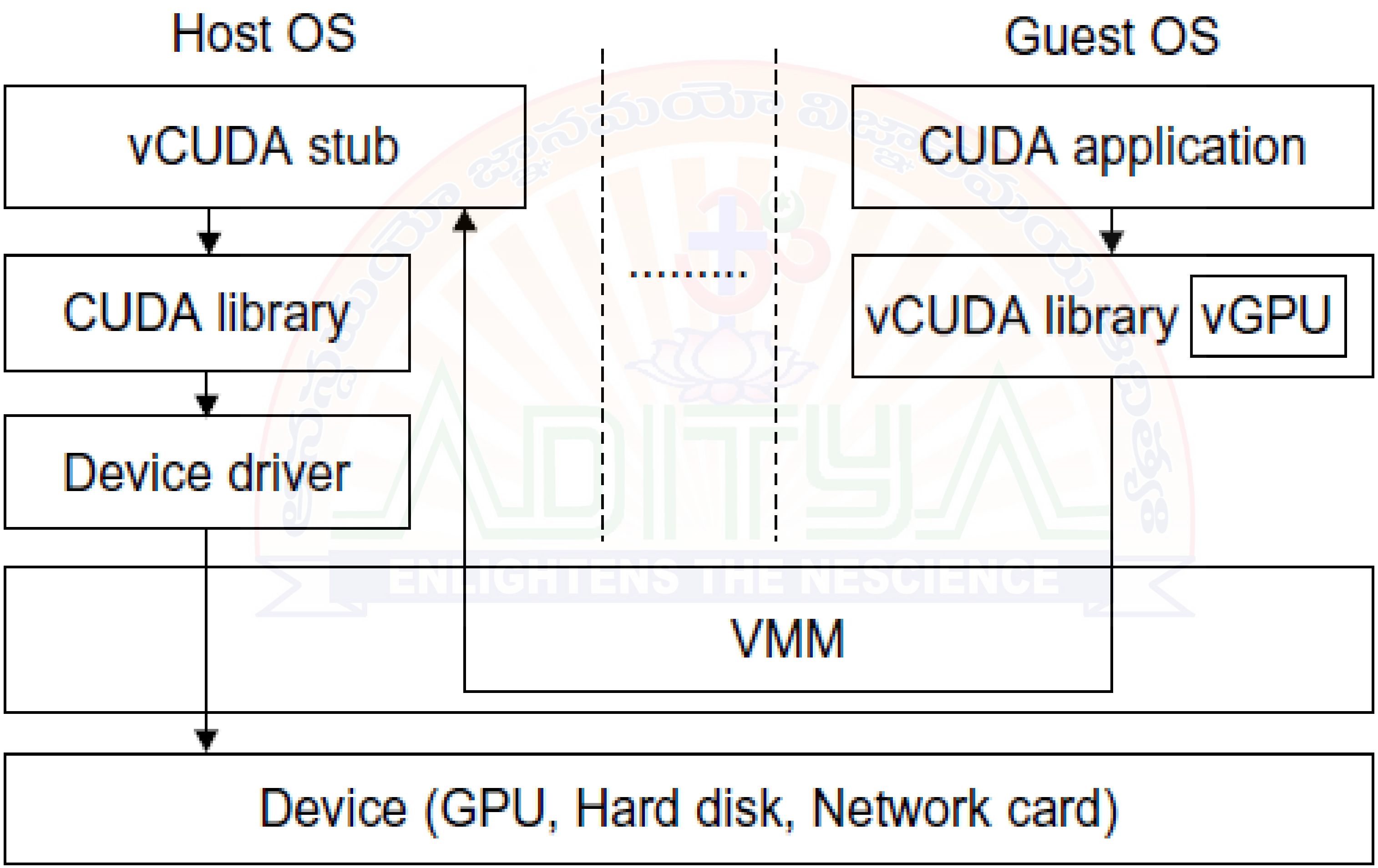
Table 3.4 Middleware and Library Support for Virtualization

Middleware or Runtime Library and References or Web Link	Brief Introduction and Application Platforms
WABI (http://docs.sun.com/app/docs/doc/802-6306)	Middleware that converts Windows system calls running on x86 PCs to Solaris system calls running on SPARC workstations
Lxrun (Linux Run) (http://www.ugcs.caltech.edu/~steven/lxrun/)	A system call emulator that enables Linux applications written for x86 hosts to run on UNIX systems such as the SCO OpenServer
WINE (http://www.winehq.org/)	A library support system for virtualizing x86 processors to run Windows applications under Linux, FreeBSD, and Solaris
Visual MainWin (http://www.mainsoft.com/)	A compiler support system to develop Windows applications using Visual Studio to run on Solaris, Linux, and AIX hosts
vCUDA (Example 3.2) (IEEE *IPDPS* 2009 [57])	Virtualization support for using general-purpose GPUs to run data-intensive applications under a special guest OS

The Basic Concept of vCUDA Architecture

- CUDA is a programming model and library for general-purpose GPUs.
- vCUDA virtualizes the CUDA library and can be installed on guest OSes.
- When CUDA applications run on a guest OS and issue a call to the CUDA API, vCUDA intercepts the call and redirects it to the CUDA API running on the host OS.
- The vCUDA employs a client-server model to implement CUDA virtualization.
- It consists of three user space components: the vCUDA library, a virtual GPU in the guest OS (which acts as a client), and the vCUDA stub in the host OS (which acts as a server).
- The vCUDA library resides in the guest OS as a substitute for the standard CUDA library. It is responsible for intercepting and redirecting API calls from the client to the stub.
- Besides these tasks, vCUDA also creates vGPUs and manages them. vGPU is responsible for storing the CUDA API flow
- The vCUDA stub returns the results to the guest OS after processing.

The Basic Concept of vCUDA Architecture



Virtualization Structures/Tools and Mechanisms

- Before virtualization, the operating system manages the hardware. After virtualization, a virtualization layer is inserted between the hardware and the operating system.
- In such a case, the virtualization layer is responsible for converting portions of the real hardware into virtual hardware.
- Therefore, different operating systems such as Linux and Windows can run on the same physical machine, simultaneously.
- Depending on the position of the virtualization layer, there are several classes of VM architectures, namely the
    hypervisor architecture (VMM),
- host-based virtualization, and
- para-virtualization.

Hypervisor Architecture

- The hypervisor supports hardware-level virtualization (see Figure 3.1(b)) on bare metal devices like CPU, memory, disk and network interfaces.
- The hypervisor software sits directly between the physical hardware and its OS.
- This virtualization layer is referred to as either the VMM or the hypervisor.
- The hypervisor provides hypercalls for the guest OSes and applications.
- Depending on the functionality, a hypervisor can assume a micro-kernel architecture like the Microsoft Hyper-V. Or it can assume a monolithic hypervisor architecture like the VMware ESX for server virtualization.

Eg: Xen, KVM

The XEN Architecture

Xen is an open source hypervisor program developed by Cambridge University. Xen is a micro-kernel hypervisor, which separates the policy from the mechanism. The Xen hypervisor implements all the mechanisms, leaving the policy to be handled by Domain 0, as shown in Figure 3.5. Xen does not include any device drivers natively [7]. It just provides a mechanism by which a guest OS can have direct access to the physical devices. As a result, the size of the Xen hypervisor is kept rather small. Xen provides a virtual environment located between the hardware and the OS. A number of vendors are in the process of developing commercial Xen hypervisors, among them are Citrix XenServer [62] and Oracle VM [42].

The XEN Architecture

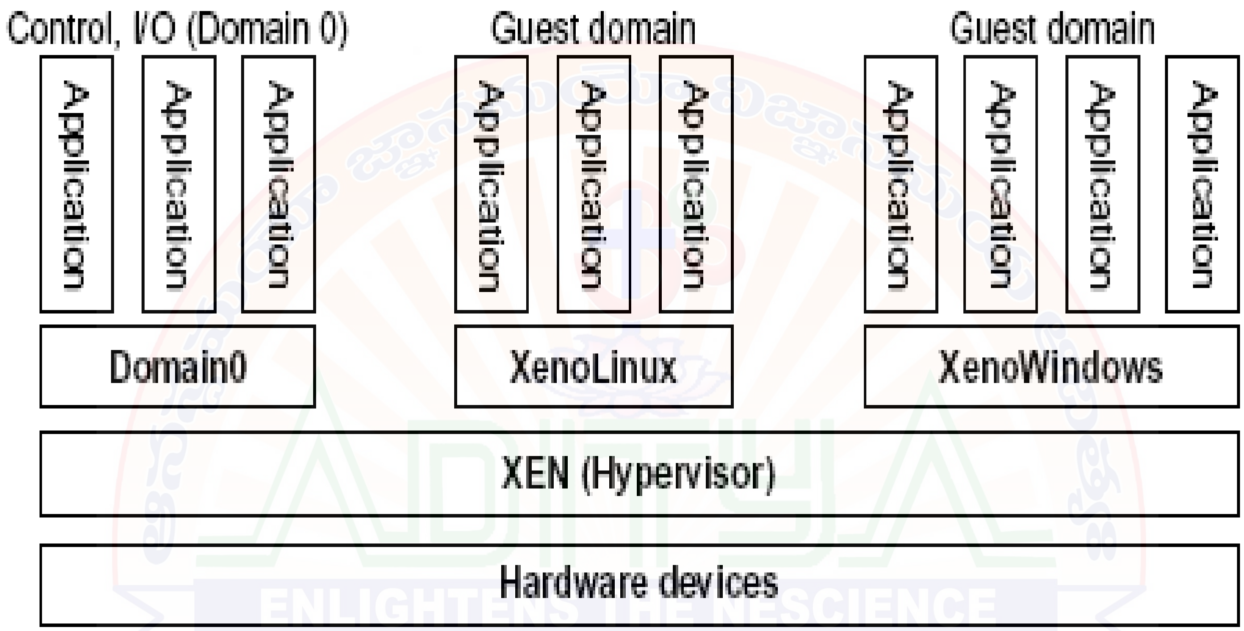


FIGURE 3.5

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

The XEN Architecture

The core components of a Xen system are the hypervisor, kernel, and applications. The organization of the three components is important. Like other virtualization systems, many guest OSes can run on top of the hypervisor. However, not all guest OSes are created equal, and one in particular controls the others. The guest OS, which has control ability, is called Domain 0, and the others are called Domain U. Domain 0 is a privileged guest OS of Xen. It is first loaded when Xen boots without any file system drivers being available. Domain 0 is designed to access hardware directly and manage devices. Therefore, one of the responsibilities of Domain 0 is to allocate and map hardware resources for the guest domains (the Domain U domains).

For example, Xen is based on Linux and its security level is C2. Its management VM is named Domain 0, which has the privilege to manage other VMs implemented on the same host. If Domain 0 is compromised, the hacker can control the entire system. So, in the VM system, security policies are needed to improve the security of Domain 0. Domain 0, behaving as a VMM, allows users to create, copy, save, read, modify, share, migrate, and roll back VMs as easily as manipulating a file, which flexibly provides tremendous benefits for users. Unfortunately, it also brings a series of security problems during the software life cycle and data lifetime.

Full Virtualization

Depending on implementation technologies, hardware virtualization can be classified into two categories:

Full virtualization and

Host-based virtualization

- Does not need to modify host OS, and critical instructions are emulated by software through the use of binary translation.
- VMware Workstation applies full virtualization, which uses binary translation to automatically modify x86 software on-the-fly to replace critical instructions.
- Advantage: no need to modify OS.
- Disadvantage: binary translation slows down the performance.

Full Virtualization

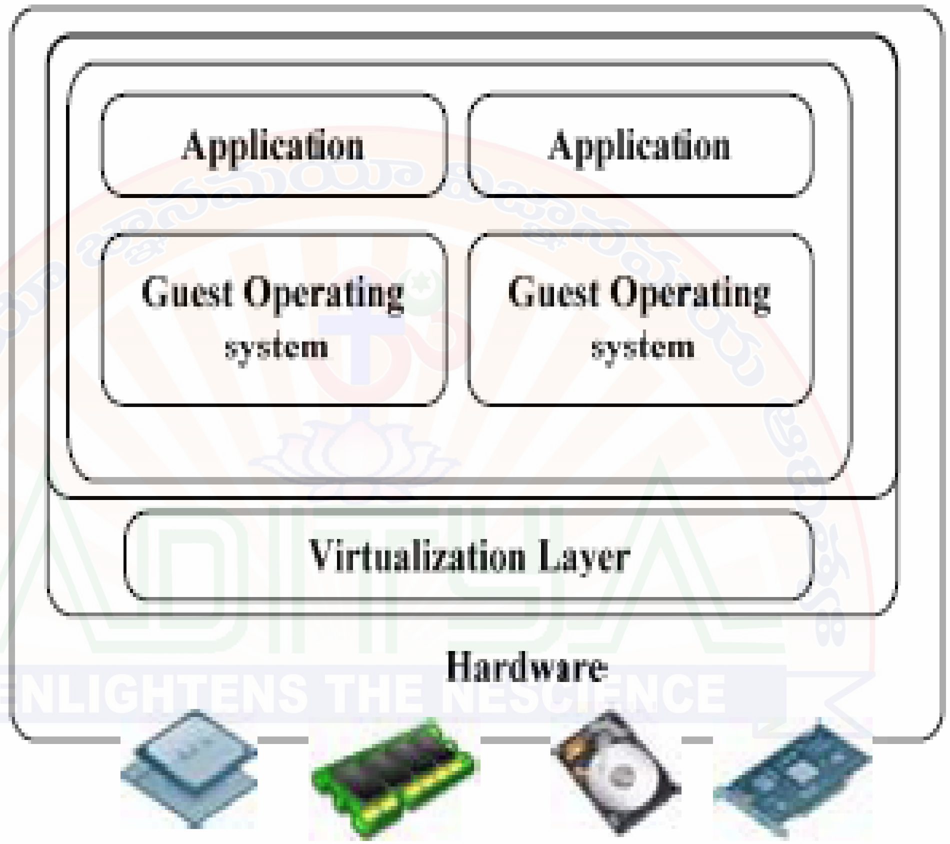


Figure 6.9 The concept of full virtualization using a hypervisor or a VMM directly sitting on top of the bare hardware devices. Note that no host OS is used here as in Figure 6.11.

Host-based Virtualization

- Virtualization layer on top of the host OS.
- This host OS is still responsible for managing the hardware.
- The guest OSes are installed and run on top of the virtualization layer.
- Dedicated applications may run on the VMs. Certainly, some other applications can also run with the host OS directly

Host-based Virtualization

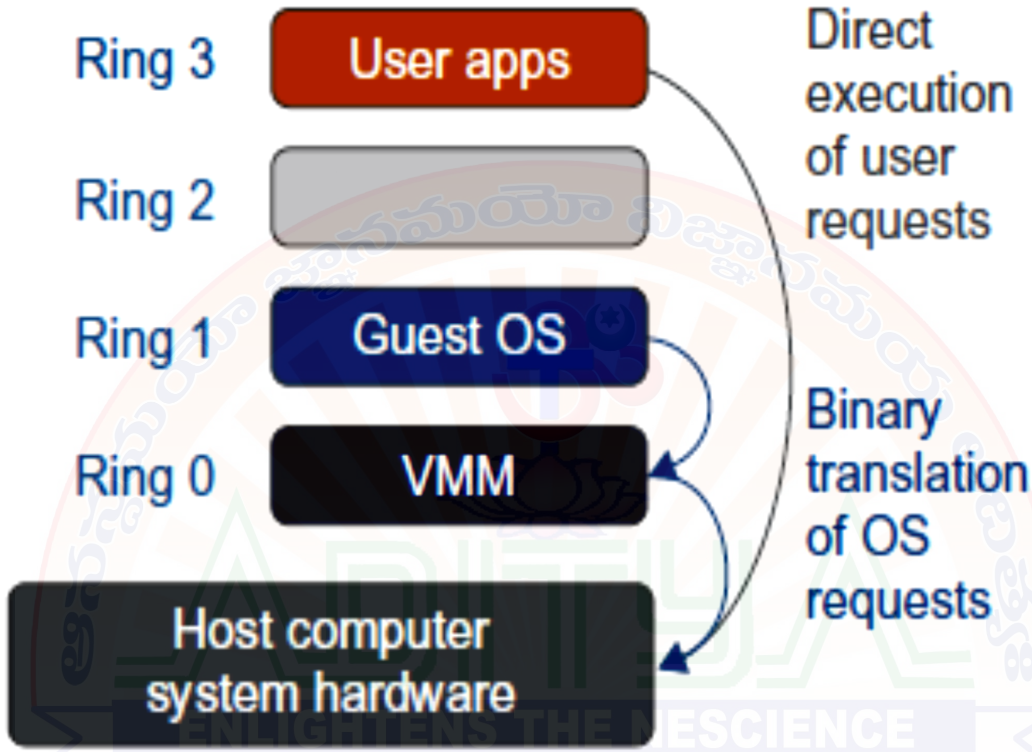


FIGURE 3.6

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

- Needs to modify the Guest OS.
- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications.
- Performance degradation is a critical issue of a virtualized system.
- Attempts to reduce the virtualization overhead, and thus improve performance by modifying only the guest OS kernel.
- The guest OS are Para virtualized, non-virtualizable instructions are replaced by hypercalls that communicate directly with the hypervisor or VMM.
- The x86 processor offers 4 instruction execution rings: Rings 0, 1, 2, 3.
- The lower the ring number, the higher the privilege of instruction being executed.
- OS is responsible for managing the hardware and the privileged instructions to execute at Ring 0, while user-level applications run at Ring 3.
- Eg: KVM, Xen, VMware ESX Server

Para-Virtualization Architecture

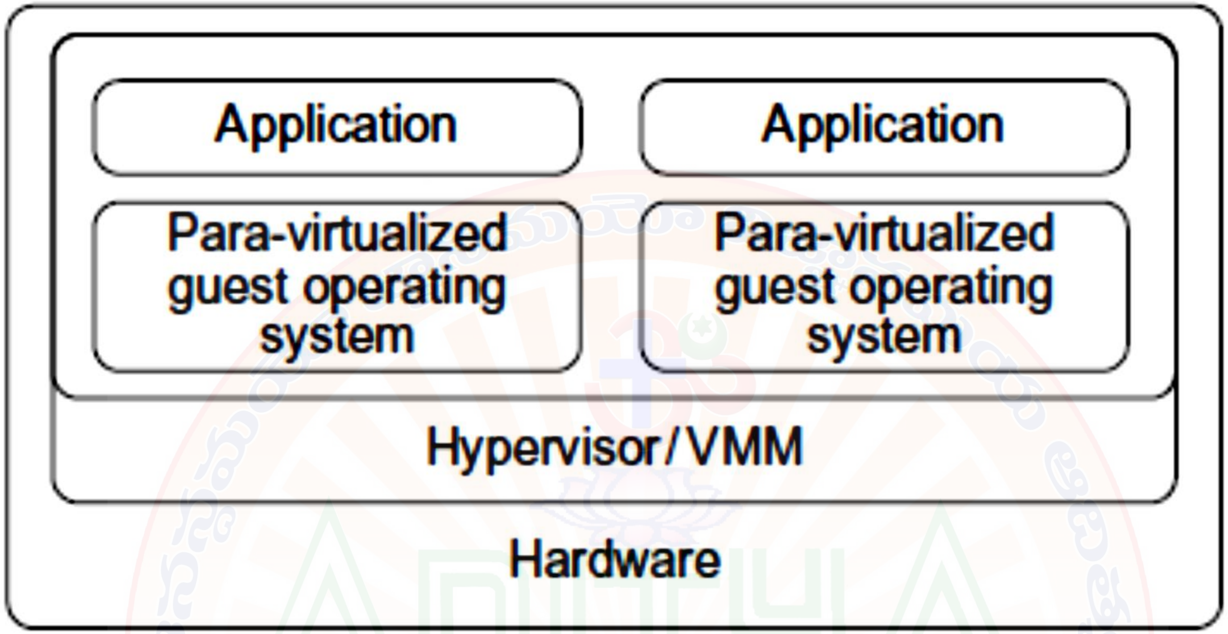


FIGURE 3.7

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 process (See Figure 3.8 for more details.)

Para-Virtualization Architecture

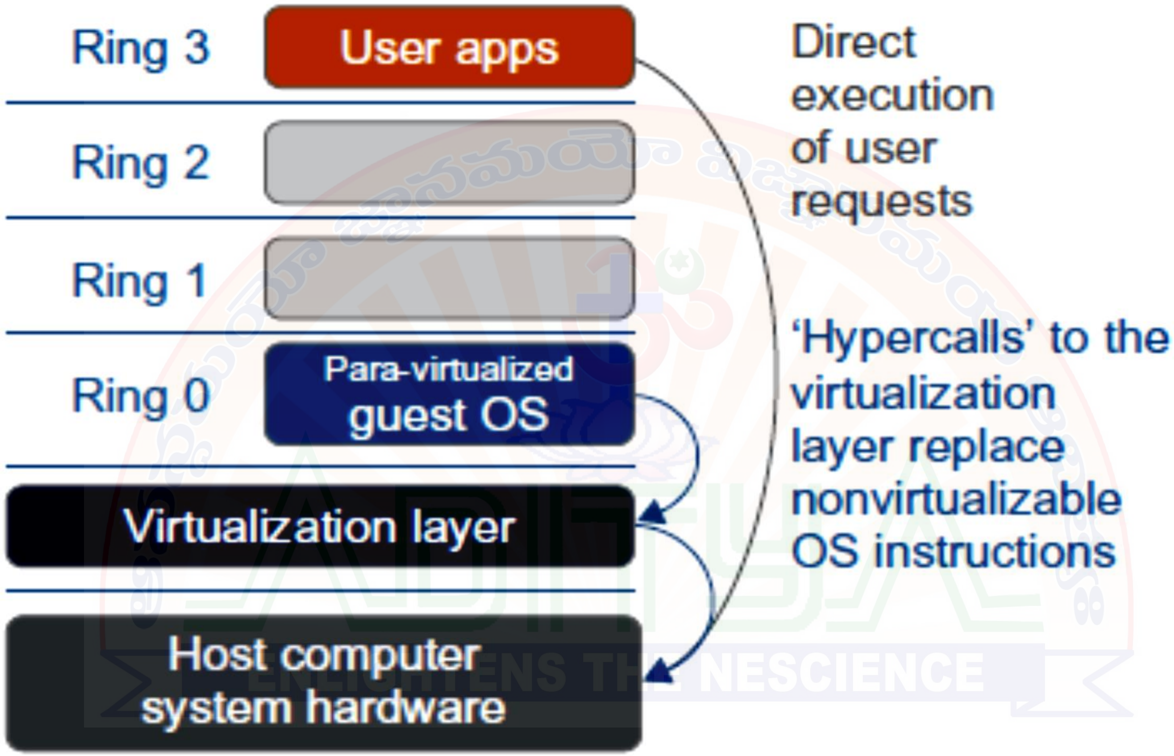


FIGURE 3.8

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

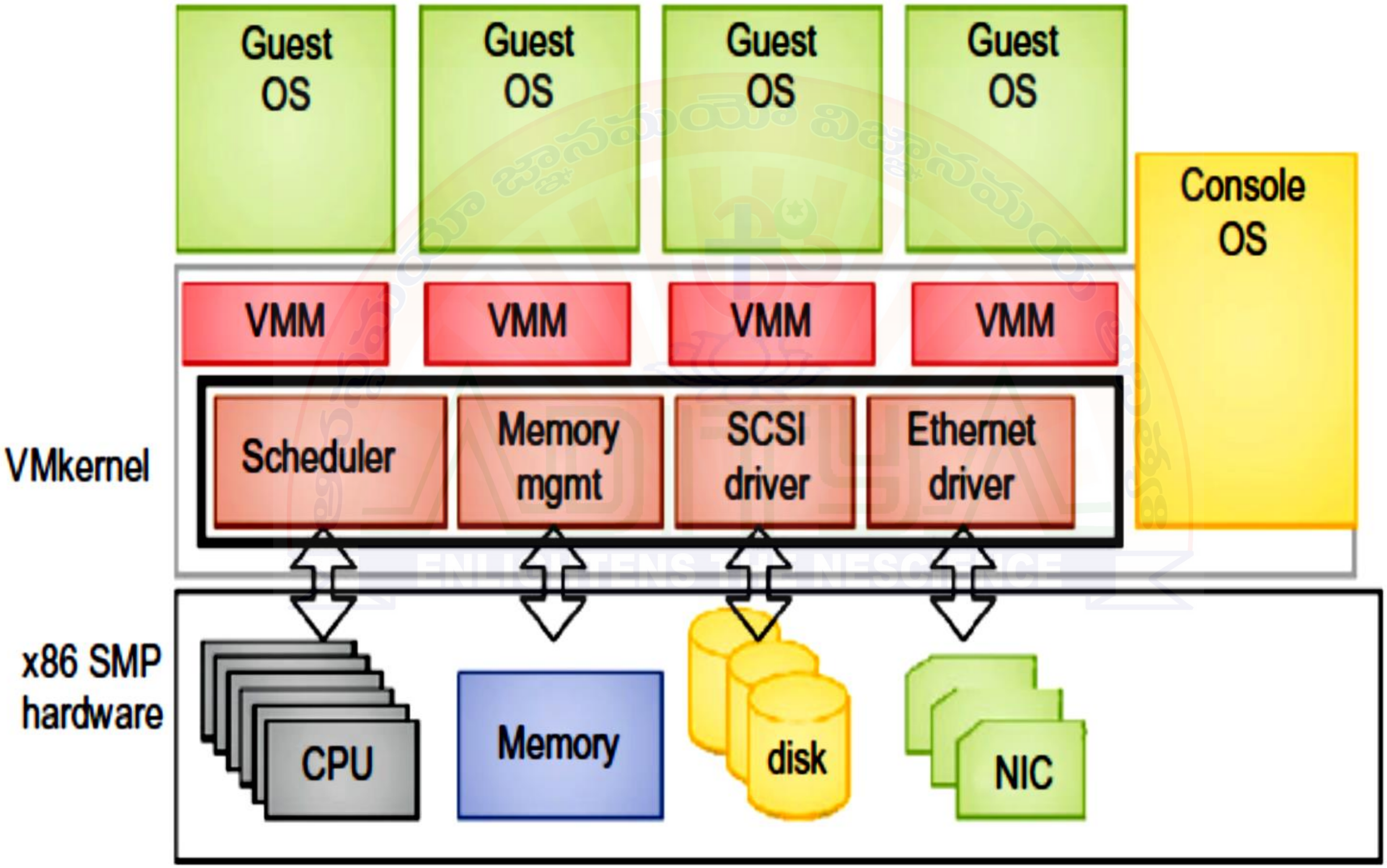
Linux Para-Virtualization System : KVM (Kernel-Based VM)

- This is a Linux para-virtualization system—a part of the Linux version 2.6.20 kernel.
- Memory management and scheduling activities are carried out by the existing Linux kernel.
- The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine.
- KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants.

Para-Virtualization with Compiler Support

- Unlike the full virtualization architecture which intercepts and emulates privileged and sensitive instructions at runtime, para-virtualization handles these instructions at compile time.
- The guest OS kernel is modified to replace the privileged and sensitive instructions with hyper calls to the hypervisor or VMM.
- Xen assumes such a para-virtualization architecture.
- The guest OS running in a guest domain may run at Ring 1 instead of at Ring 0.
- This implies that the guest OS may not be able to execute some privileged and sensitive instructions.
- The privileged instructions are implemented by hyper calls to the hypervisor.

VMWare ESX Server for Para-Virtualization



Virtualization of CPU, Memory and I/O Devices

- CPU virtualization demands hardware-assisted traps of sensitive instructions by the VMM
- Memory virtualization demands special hardware support (shadow page tables by VMWare or extended page table by Intel) to help translate virtual address into physical address and machine memory in two stages.
- I/O virtualization is the most difficult one to realize due to the complexity if I/O service routines and the emulation needed between the guest OS and host OS.

CPU Virtualization

- A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode.
- Unprivileged instructions of VMs run directly on the host machine and Other critical instructions should be handled carefully.
- The critical instructions are divided into 3 categories:
- privileged instructions,
- control sensitive instructions
- behavior-sensitive instructions.
- Privileged instructions execute in a privileged(supervisor) mode and will be trapped if executed outside this mode.
- Control-sensitive instructions attempt to change the configuration of resources used.
- Behavior-sensitive instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

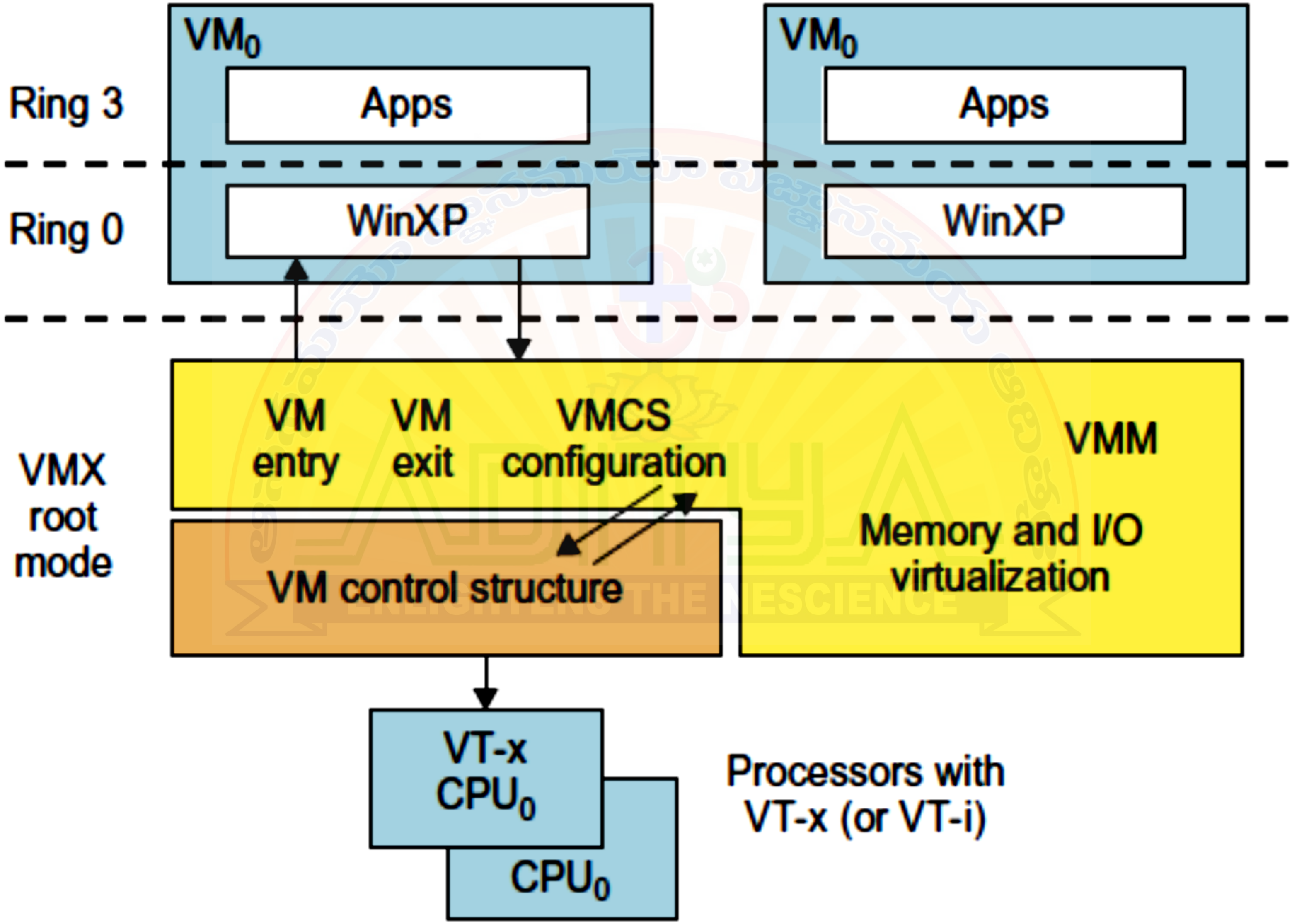
CPU Virtualization

- A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode.

- When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM. In this case, the VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system.

- Eg: RISC CPU can be virtualized

CPU Virtualization



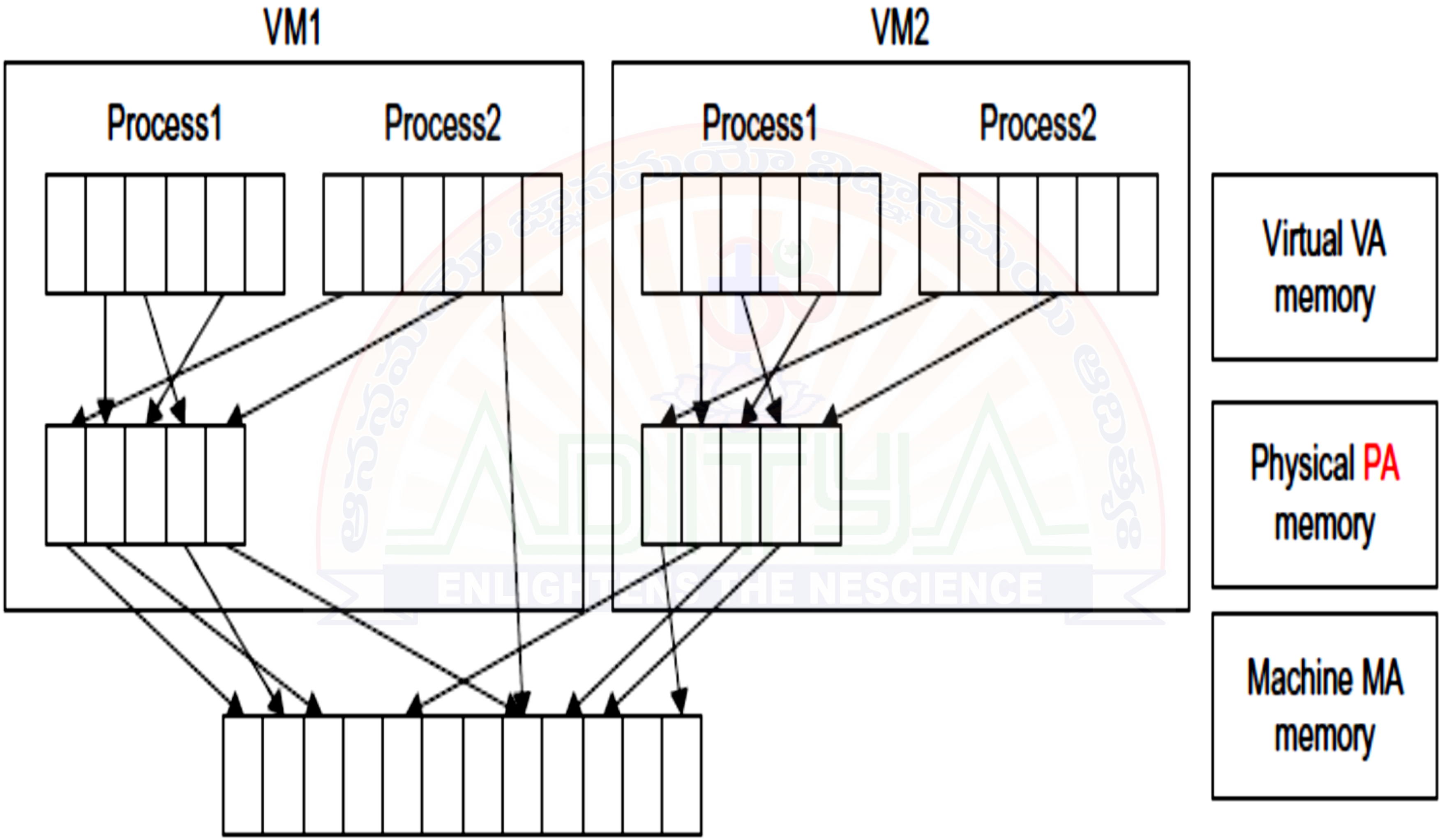
Memory Virtualization

- Virtual memory virtualization is similar to the virtual memory support provided by modern OS.
- In traditional execution environment, OS maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory.
- All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance.
- However, in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.

Memory Virtualization

- That means a two-stage mapping process should be maintained by the guest OS and the VMM, respectively:
- virtual memory to physical memory and
- physical memory to machine memory.
- MMU virtualization should be supported, which is transparent to the guest OS.
- The guest OS controls the mapping of virtual addresses to the physical memory addresses of VMs.
- Guest OS cannot directly access the actual machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory.
- The VMM page table is called shadow page table.

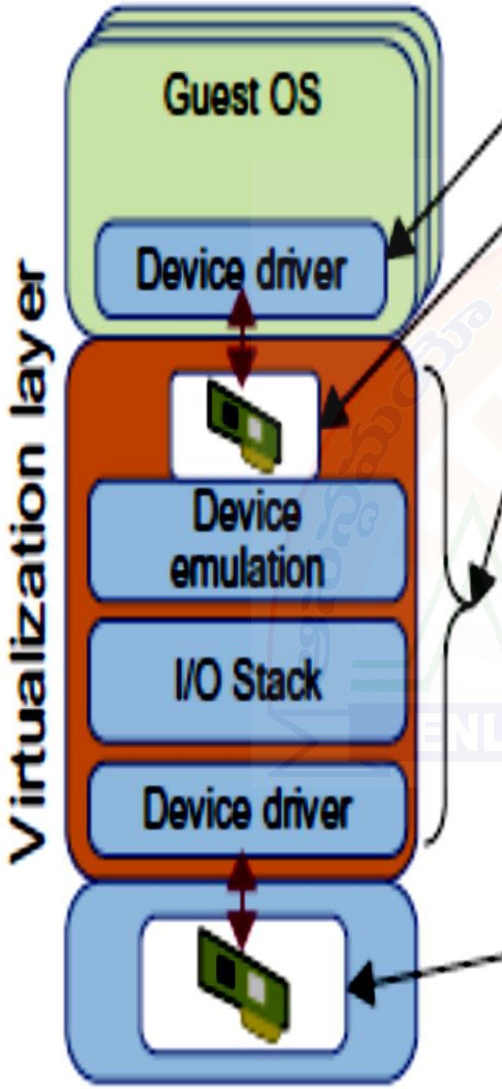
Memory Virtualization



I/O Virtualization

- I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.
- There are 3 ways to implement I/O virtualization:
- full device emulation,
- para-virtualization, and
- direct I/O.
- Full device emulation is the first approach for I/O virtualization.
- All the functions of a device (such as device enumeration, identification, interrupts, and DMA) are replicated in software located in the VMM and acts as a virtual device.
- The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.
- Software Emulation is slower.

I/O Virtualization



- Guest device driver
- Virtual device
- Virtualization layer
  - emulates the virtual device
  - remaps guest and real I/O addresses
  - multiplexes and drives the physical device
  - I/O features. e.g., COW disks
- Real device
  - may be different from virtual device

I/O Virtualization

- The para-virtualization method is also known as the split driver model consisting of a frontend driver and a backend driver. Eg: Used in Xen

- The frontend driver is running in Domain U and the backend driver is running in Domain 0.

- The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices.

- Para-I/O-virtualization has a higher CPU overhead.

- Direct I/O virtualization lets the VM access devices directly

Virtualization in Multi-Core Processors

- Virtualizing a multi-core processor is relatively more complicated than virtualizing a uni-core processor.
- Muti-core virtualization has raised some new challenges despite its high performance.
- There are mainly two difficulties:
- Application programs must be parallelized to use all cores fully and software must explicitly assign tasks to the cores, which is a very complex problem.
- The second challenge has spawned research involving scheduling algorithms and resource management policies.
- New programming models, languages, and libraries are needed to make parallel programming easier.

Physical versus Virtual Processor Cores

Physical Cores	Virtual Cores
The actual physical cores present in the processor.	There can be more virtual cores visible to a single OS than there are physical cores.
More burden on the software to write applications which can run directly on the cores.	Design of software becomes easier as the hardware assists the software in dynamic resource utilization.
Hardware provides no assistance to the software and is hence simpler.	Hardware provides assistance to the software and is hence more complex.
Poor resource management.	Better resource management.
The lowest level of system software has to be modified.	The lowest level of system software need not be modified.

Multi-Core Virtualization
VCPU vs. Traditional CPU

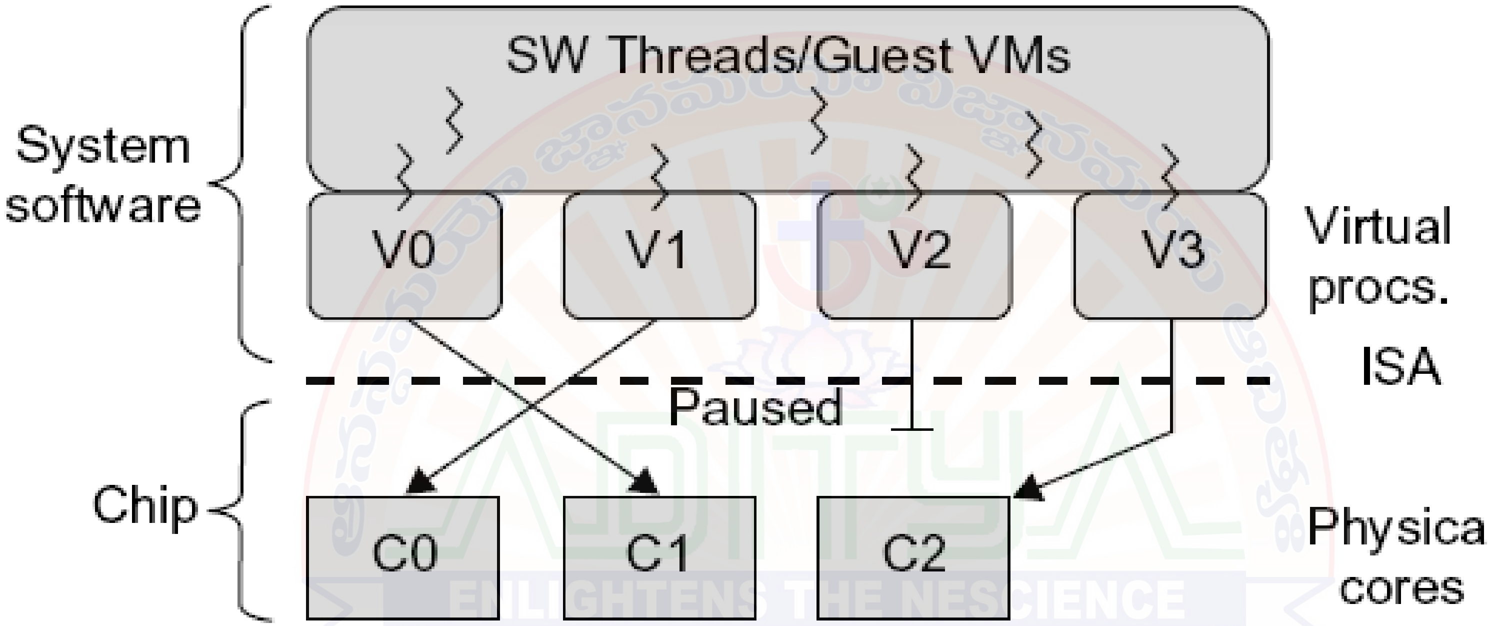
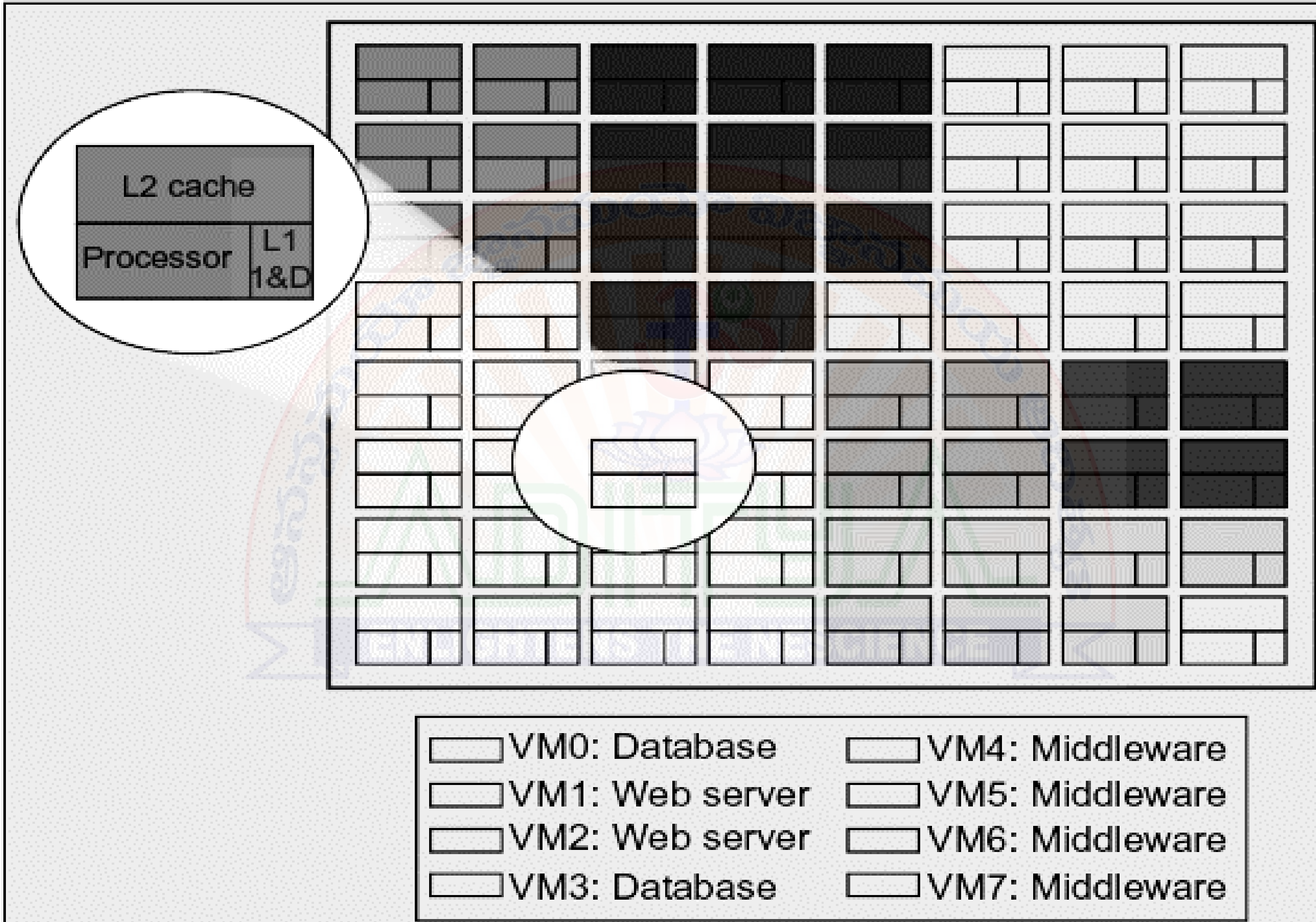


Figure 3.16 Four VCPUs are exposed to the software, only three cores are actually present. VCPUs V0, V1, and V3 have been transparently migrated, while VCPU V2 has been transparently suspended.

Multi-Core Virtualization



(a) Mapping of VMs into adjacent cores

Virtual Clusters in Many Cores
Space Sharing of VMs -- Virtual Hierarchy

(b) Multiple virtual clusters assigned to various workloads

VIRTUAL CLUSTERS AND RESOURCE MANAGEMENT

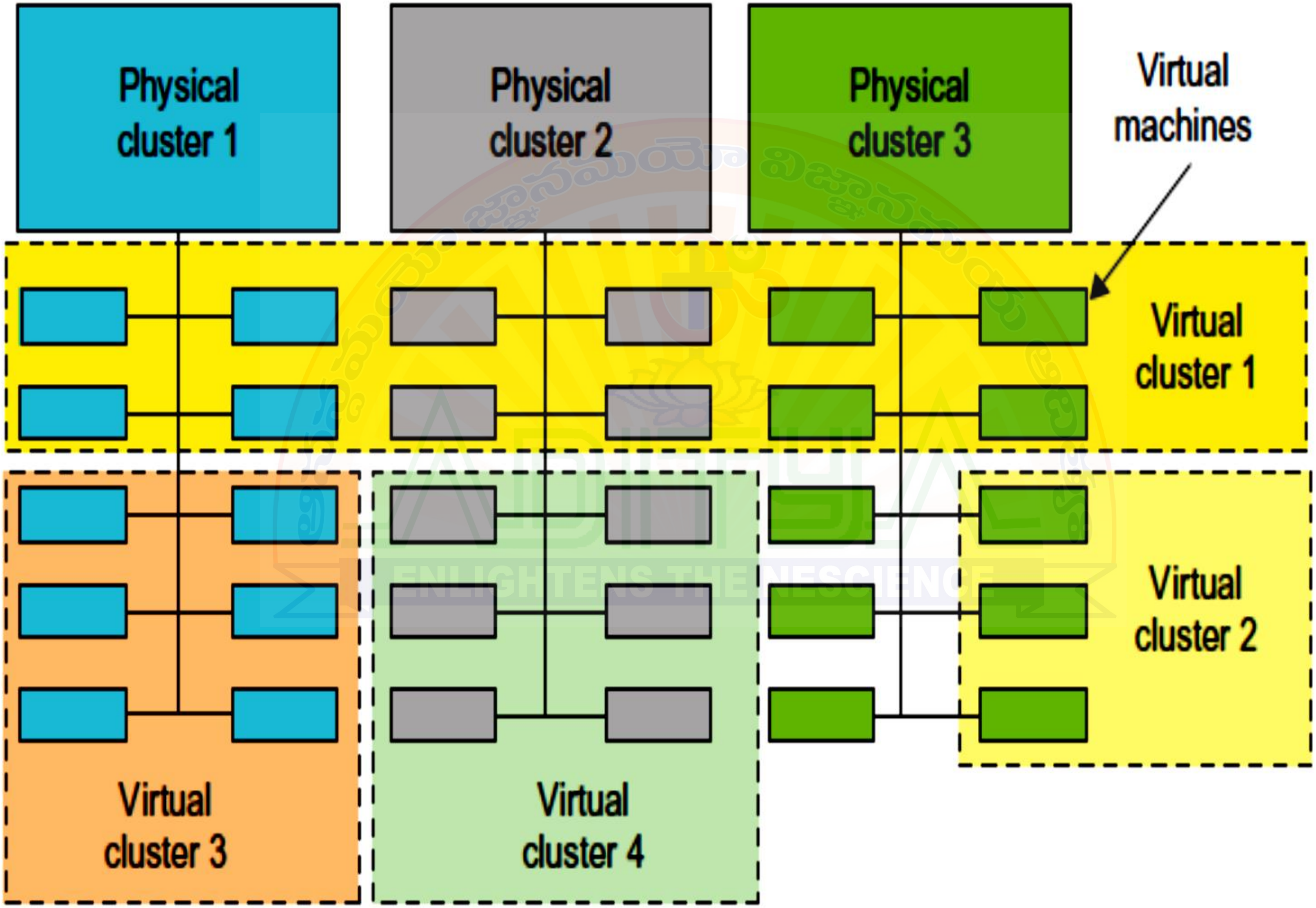
- A physical cluster is a collection of servers (physical machines) interconnected by a physical network.
- Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters.
- The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks.
- Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters.

VIRTUAL CLUSTERS AND RESOURCE MANAGEMENT

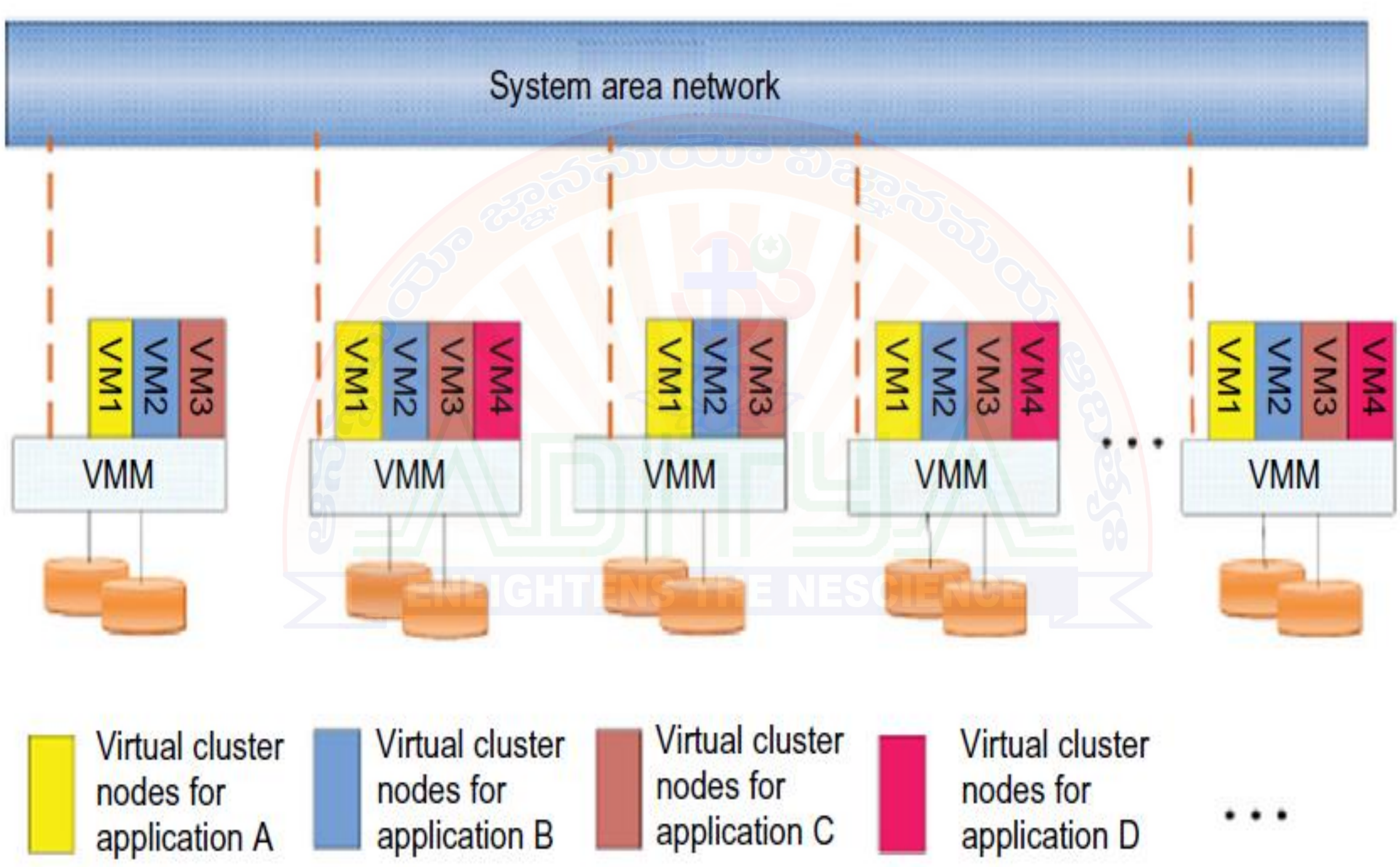
Virtual Cluster Characteristics

- The virtual cluster nodes can be either physical or virtual machines. Multiple VMs running with different OSs can be deployed on the same physical node.
- A VM runs with a guest OS, which is often different from the host OS, that manages the resources in the physical machine, where the VM is implemented.
- The purpose of using VMs is to consolidate multiple functionalities on the same server. This will greatly enhance the server utilization and application flexibility.
- VMs can be colonized (replicated) in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery.
- The size (number of nodes) of a virtual cluster can grow or shrink dynamically, similarly to the way an overlay network varies in size in a P2P network.
- The failure of any physical nodes may disable some VMs installed on the failing nodes. But the failure of VMs will not pull down the host system.

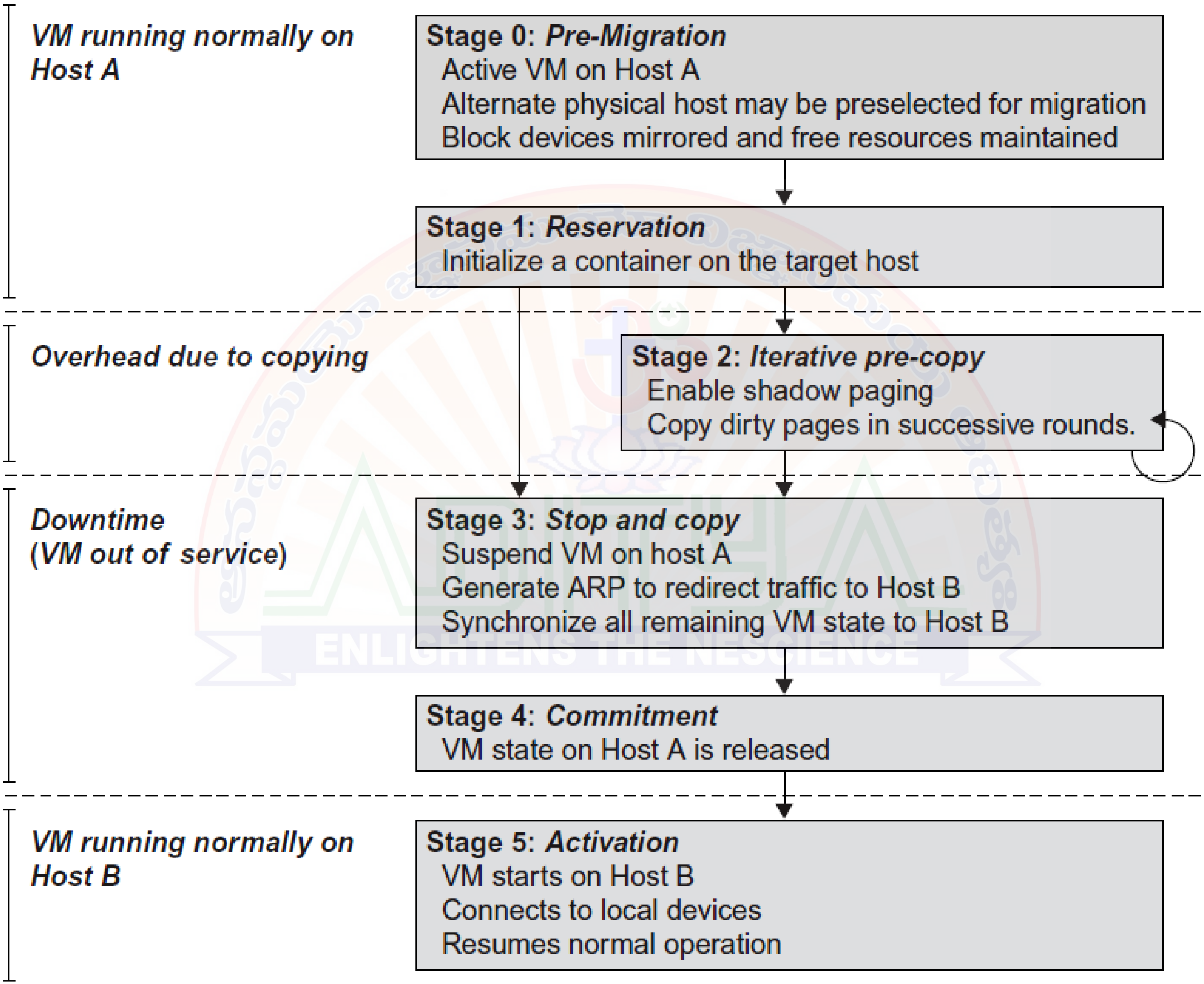
Virtual Clusters vs. Physical Clusters



The concept of a virtual cluster based on application partitioning



Live Migration of Virtual Machines



Migration of Memory, Files, and Network Resources

Memory Migration

- This is one of the most important aspects of VM migration.
- Moving the memory instance of a VM from one physical host to another can be approached in any number of ways.
- The techniques employed for this purpose depend upon the characteristics of application/workloads supported by the guest OS.
- Memory migration can be in a range of hundreds of megabytes to a few gigabytes in a typical system today, and it needs to be done in an efficient manner.
- The Internet Suspend-Resume (ISR) technique exploits temporal locality as memory states are likely to have considerable overlap in the suspended and the resumed instances of a VM.
- Temporal locality refers to the fact that the memory states differ only by the amount of work done since a VM was last suspended before being initiated for migration.

Migration of Memory, Files, and Network Resources

File System Migration

- To support VM migration, a system must provide each VM with a consistent, location-independent view of the file system that is available on all hosts.
- A simple way to achieve this is to provide each VM with its own virtual disk which the file system is mapped to and transport the contents of this virtual disk along with the other states of the VM.
- However, due to the current trend of high capacity disks, migration of the contents of an entire disk over a network is not a viable solution.
- Another way is to have a global file system across all machines where a VM could be located.
- This way removes the need to copy files from one machine to another because all files are network accessible.

Migration of Memory, Files, and Network Resources

File System Migration

- In smart copying, the VMM exploits spatial locality. Typically, people often move between the same small number of locations, such as their home and office.
- In these conditions, it is possible to transmit only the difference between the two file systems at suspending and resuming locations.
- This technique significantly reduces the amount of actual physical data that has to be moved.
- In situations where there is no locality to exploit, a different approach is to synthesize much of the state at the resuming site.

Migration of Memory, Files, and Network Resources

Network Migration

- A migrating VM should maintain all open network connections without relying on forwarding mechanisms on the original host or on support from mobility or redirection mechanisms.
- To enable remote systems to locate and communicate with a VM, each VM must be assigned a virtual IP address known to other entities.
- This address can be distinct from the IP address of the host machine where the VM is currently located.
- Each VM can also have its own distinct virtual MAC address.
- The VMM maintains a mapping of the virtual IP and MAC addresses to their corresponding VMs. In general, a migrating VM includes all the protocol states and carries its IP address with it.

Migration of Memory, Files, and Network Resources

Network Migration

- Live migration means moving a VM from one physical node to another while keeping its OS environment and applications unbroken.
- This capability is being increasingly utilized in today's enterprise environments to provide efficient online system maintenance, reconfiguration, load balancing, and proactive fault tolerance.
- It provides desirable features to satisfy requirements for computing resources in modern computing systems, including server consolidation, performance isolation, and ease of management.

Dynamic Deployment of Virtual Clusters

Virtual Cluster Projects

Table 3.5 Experimental Results on Four Research Virtual Clusters

Project Name	Design Objectives	Reported Results and References
Cluster-on-Demand at Duke Univ.	Dynamic resource allocation with a virtual cluster management system	Sharing of VMs by multiple virtual clusters using Sun GridEngine [12]
Cellular Disco at Stanford Univ.	To deploy a virtual cluster on a shared-memory multiprocessor	VMs deployed on multiple processors under a VMM called Cellular Disco [8]
VIOLIN at Purdue Univ.	Multiple VM clustering to prove the advantage of dynamic adaptation	Reduce execution time of applications running VIOLIN with adaptation [25,55]
GRAAL Project at INRIA in France	Performance of parallel algorithms in Xen-enabled virtual clusters	75% of max. performance achieved with 30% resource slacks over VM clusters

VIRTUALIZATION FOR DATA-CENTER AUTOMATION

- Data centers have grown rapidly in recent years, and all major IT companies Google, Yahoo!, Amazon, Microsoft, HP, Apple, and IBM are pouring their resources into building new data centers.
- All these companies have invested billions of dollars in datacenter construction and automation.
- Data-center automation means that huge volumes of hardware, software, and database resources in these data centers can be allocated dynamically to millions of Internet users simultaneously, with guaranteed QoS and cost-effectiveness.
- This automation process is triggered by the growth of virtualization products and cloud computing services.

Parallax for VM Storage Management

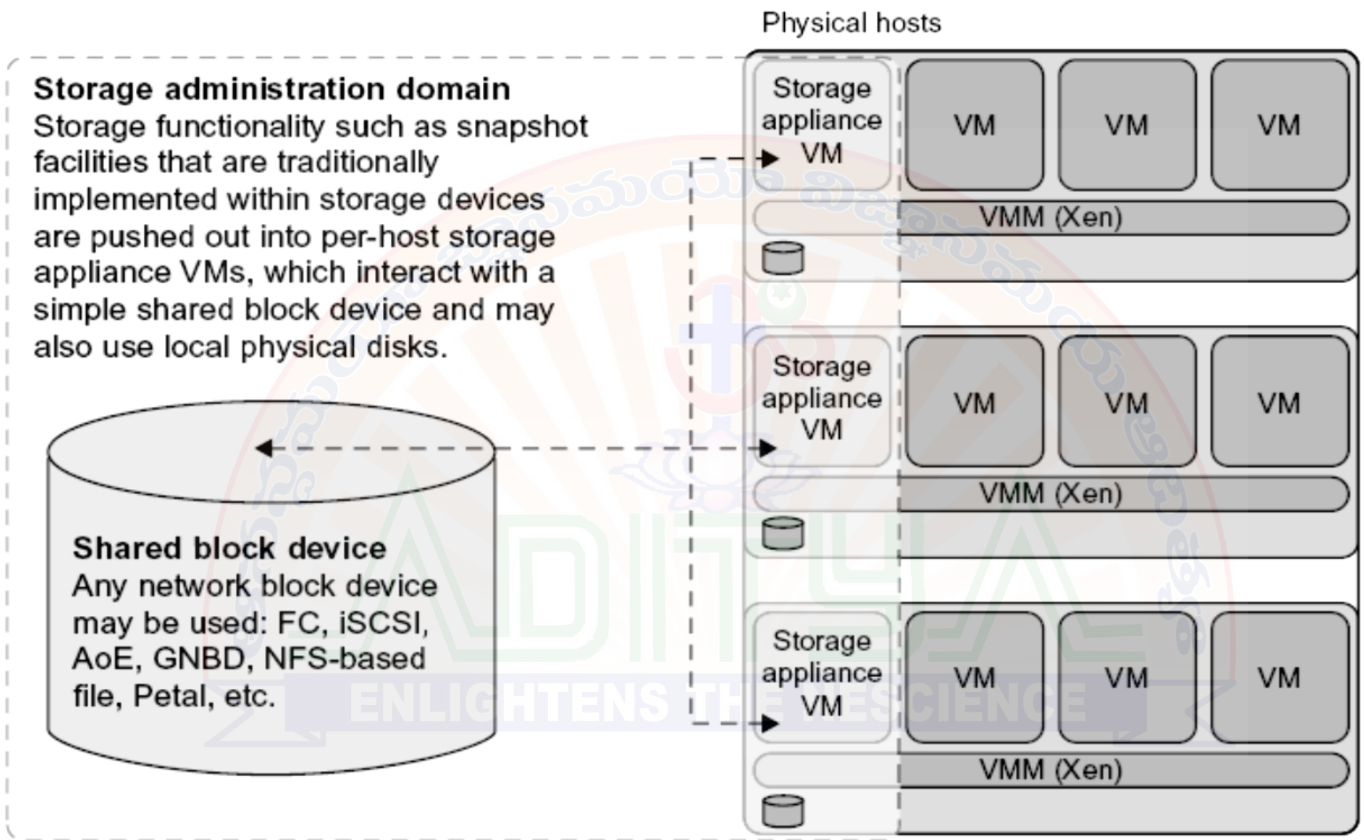


FIGURE 3.26

Parallax is a set of per-host storage appliances that share access to a common block device and presents virtual disks to client VMs.

(Courtesy of D. Meyer, et al. [43])

Cloud OS for Building Private Clouds

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

Eucalyptus : An Open-Source OS for Setting Up and Managing Private Clouds

Eucalyptus is an open source software system (Figure 3.27) intended mainly for supporting Infrastructure as a Service (IaaS) clouds. The system primarily supports virtual networking and the management of VMs; virtual storage is not supported. Its purpose is to build private clouds that can interact with end users through Ethernet or the Internet. The system also supports interaction with other private clouds or public clouds over the Internet. The system is short on security and other desired features for general-purpose grid or cloud applications.

- **Instance Manager** controls the execution, inspection, and terminating of VM instances on the host where it runs.
- **Group Manager** gathers information about and schedules VM execution on specific instance managers, as well as manages virtual instance network.
- **Cloud Manager** is the entry-point into the cloud for users and administrators. It queries node managers for information about resources, makes scheduling decisions, and implements them by making requests to group managers.

Eucalyptus : An Open-Source OS for Setting Up and Managing Private Clouds

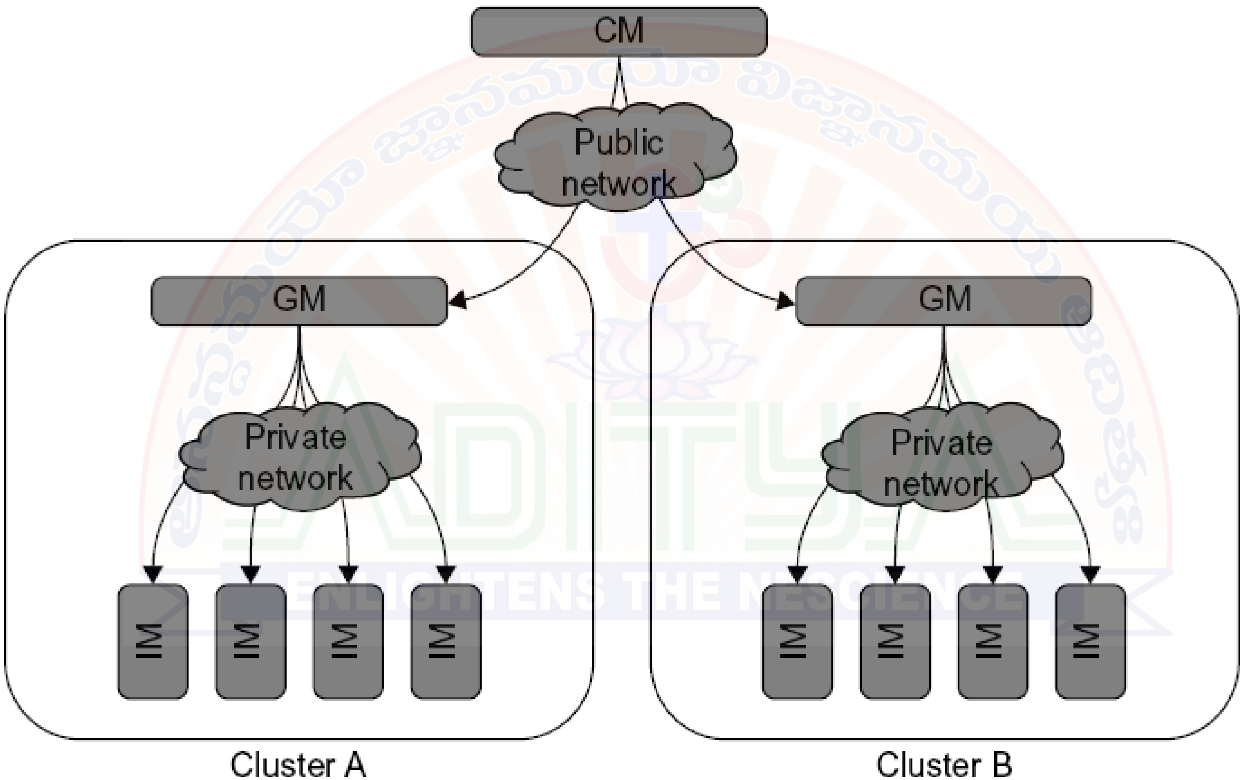
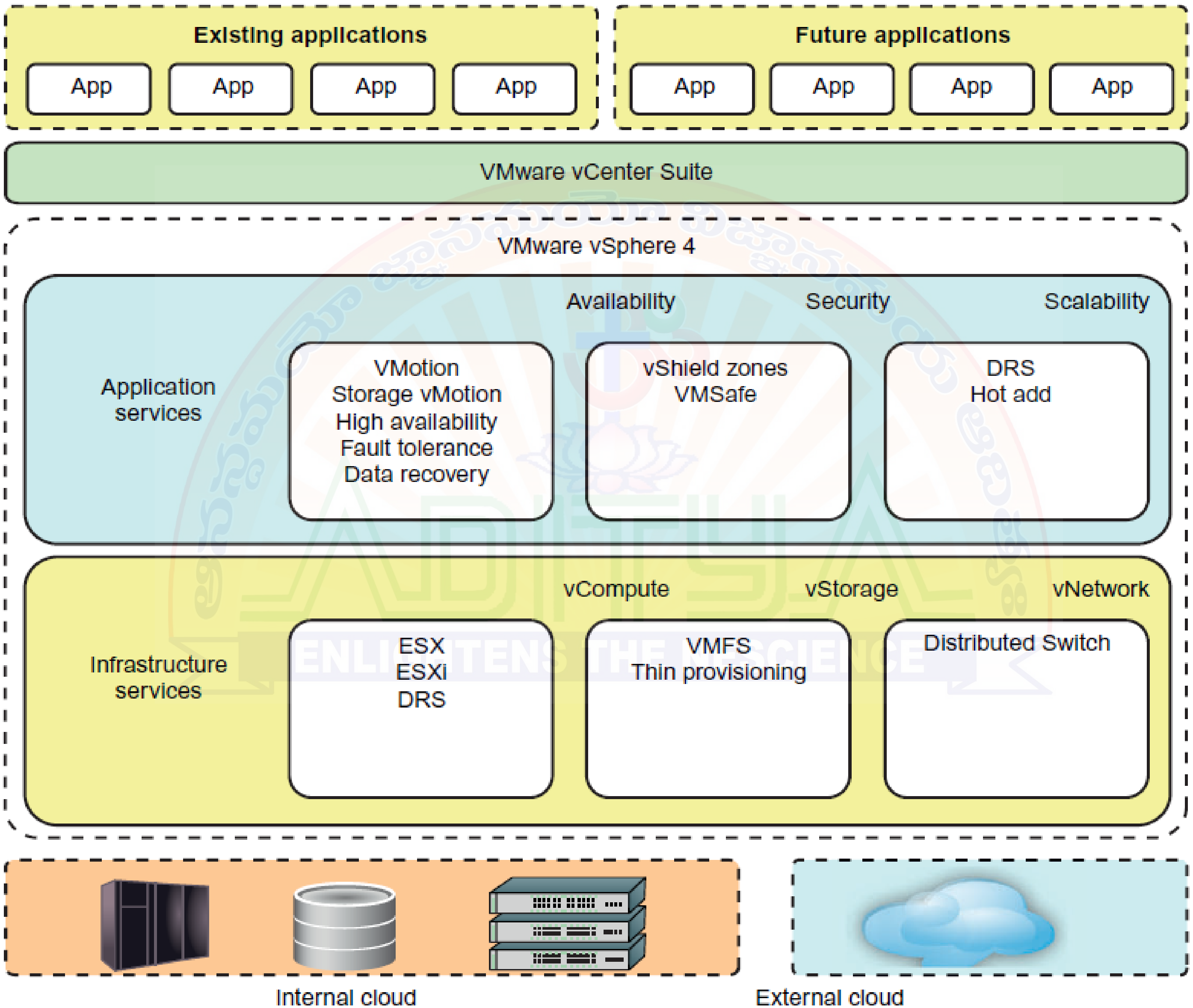


FIGURE 3.27

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

vSphere/4, a Cloud Operating System



Trusted Zones for VM Insulation

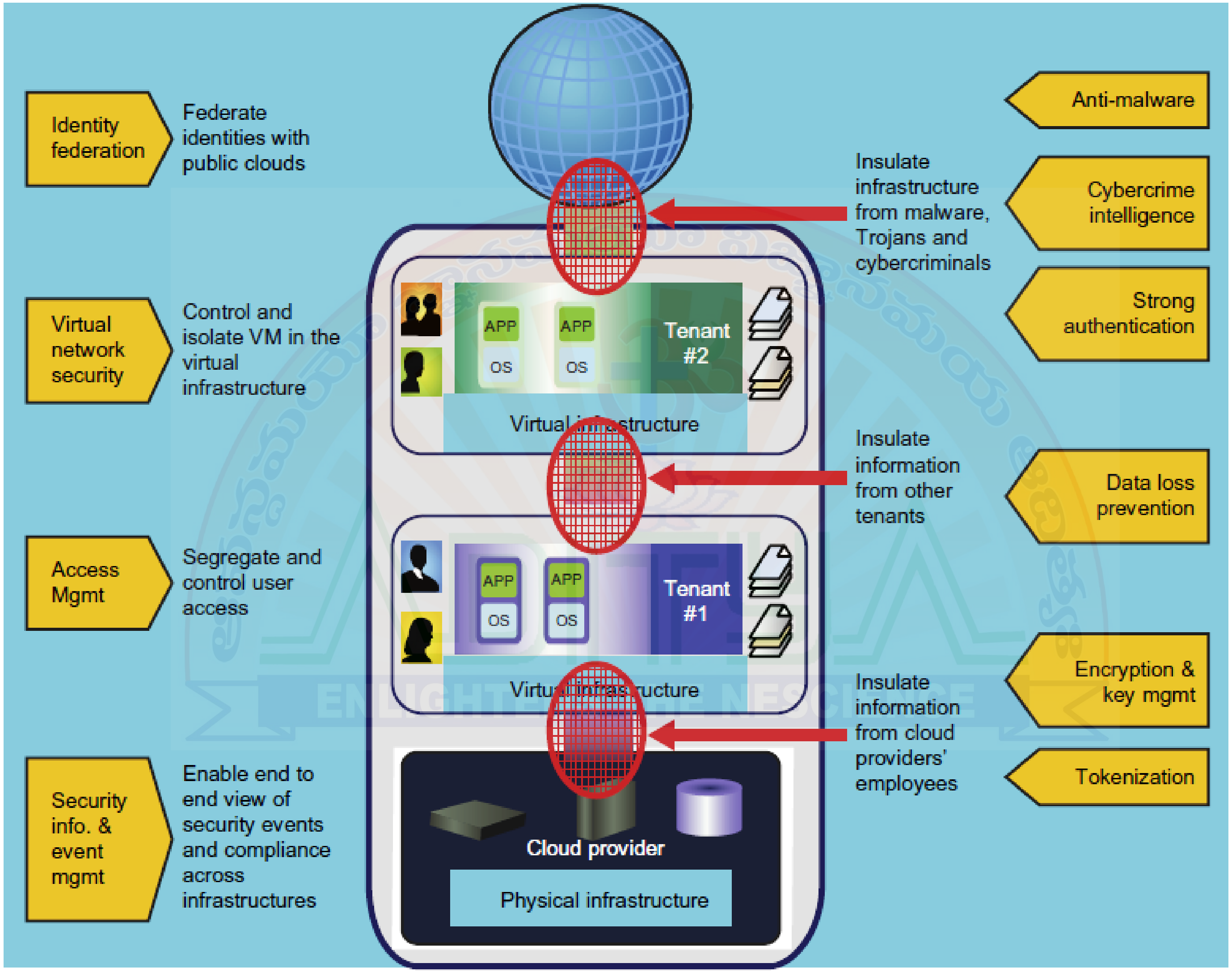


FIGURE 3.30

Techniques for establishing trusted zones for virtual cluster insulation and VM isolation.

UNIT-III

Cloud Platform Architecture

CLOUD COMPUTING AND SERVICE MODELS

Public, Private & Hybrid Clouds

- The concept of cloud computing has evolved from cluster, grid, and utility computing.
- Cluster and grid computing leverage the use of many computers in parallel to solve problems of any size.
- Utility and Software as a Service (SaaS) provide computing resources as a service with the notion of pay per use.
- Cloud computing leverages dynamic resources to deliver large numbers of services to end users.
- Cloud computing is a high-throughput computing (HTC) paradigm whereby the infrastructure provides the services through a large data center or server farms.
- The cloud computing model enables users to share access to resources from anywhere at any time through their connected devices.

Public, Private & Hybrid Clouds

- The cloud will free users to focus on user application development and create business value by outsourcing job execution to cloud providers.
- In this scenario, the computations (programs) are sent to where the data is located, rather than copying the data to millions of desktops as in the traditional approach.
- Cloud computing avoids large data movement, resulting in much better network bandwidth utilization.
- The cloud offers significant benefit to IT companies by freeing them from the low-level task of setting up the hardware (servers) and managing the system software.
- Cloud computing applies a virtual platform with elastic resources put together by on-demand provisioning of hardware, software, and data sets, dynamically.
- The main idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at data centers.

Public, Private & Hybrid Clouds

Centralized versus Distributed Computing

- Some people argue that cloud computing is centralized computing at data centers and others claim that cloud computing is the practice of distributed parallel computing over data-center resources. These represent two opposite views of cloud computing.
- All computations in cloud applications are distributed to servers in a data center.
- These are mainly virtual machines (VMs) in virtual clusters created out of data-center resources. In this sense, cloud platforms are systems distributed through virtualization.
- Both public clouds and private clouds are developed in the Internet.
- Clouds are generated by commercial providers or by enterprises in a distributed manner, they will be interconnected over the Internet to achieve scalable and efficient computing services.
- Commercial cloud providers such as Amazon, Google, and Microsoft created their platforms to be distributed geographically.

Public, Private & Hybrid Clouds

Public Clouds

- Public cloud is built over the Internet and can be accessed by any user who has paid for the service.
- Public clouds are owned by service providers and are accessible through a subscription.
- Many public clouds are available, including Google App Engine (GAE), Amazon Web Services (AWS), Microsoft Azure, IBM Blue Cloud, and Salesforce.com's Force.com.
- The providers of the aforementioned clouds are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.
- A public cloud delivers a selected set of business processes.
- The application and infrastructure services are offered on a flexible price-per-use basis.

Public, Private & Hybrid Clouds

Private Clouds

- A private cloud is built within the domain of an intranet owned by a single organization.
- Therefore, it is client owned and managed, and its access is limited to the owning clients and their partners.
- Its deployment was not meant to sell capacity over the Internet through publicly accessible interfaces.
- Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains.
- A private cloud is supposed to deliver more efficient and convenient cloud services.
- It may impact the cloud standardization, while retaining greater customization and organizational control.

Public, Private & Hybrid Clouds

Hybrid Clouds

- A hybrid cloud is built with both public and private clouds.
- Private clouds can also support a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud.
- For Eg. the Research Compute Cloud (RC2) is a private cloud, built by IBM, that interconnects the computing and IT resources at 8 IBM Research Centers scattered throughout the US, Europe, and Asia.
- A hybrid cloud provides access to clients, the partner network, and third parties.
- In summary, public clouds promote standardization, preserve capital investment, and offer application flexibility.
- Private clouds attempt to achieve customization and offer higher efficiency, resiliency, security, and privacy.
- Hybrid clouds operate in the middle, with many compromises in terms of resource sharing.

Public, Private & Hybrid Clouds

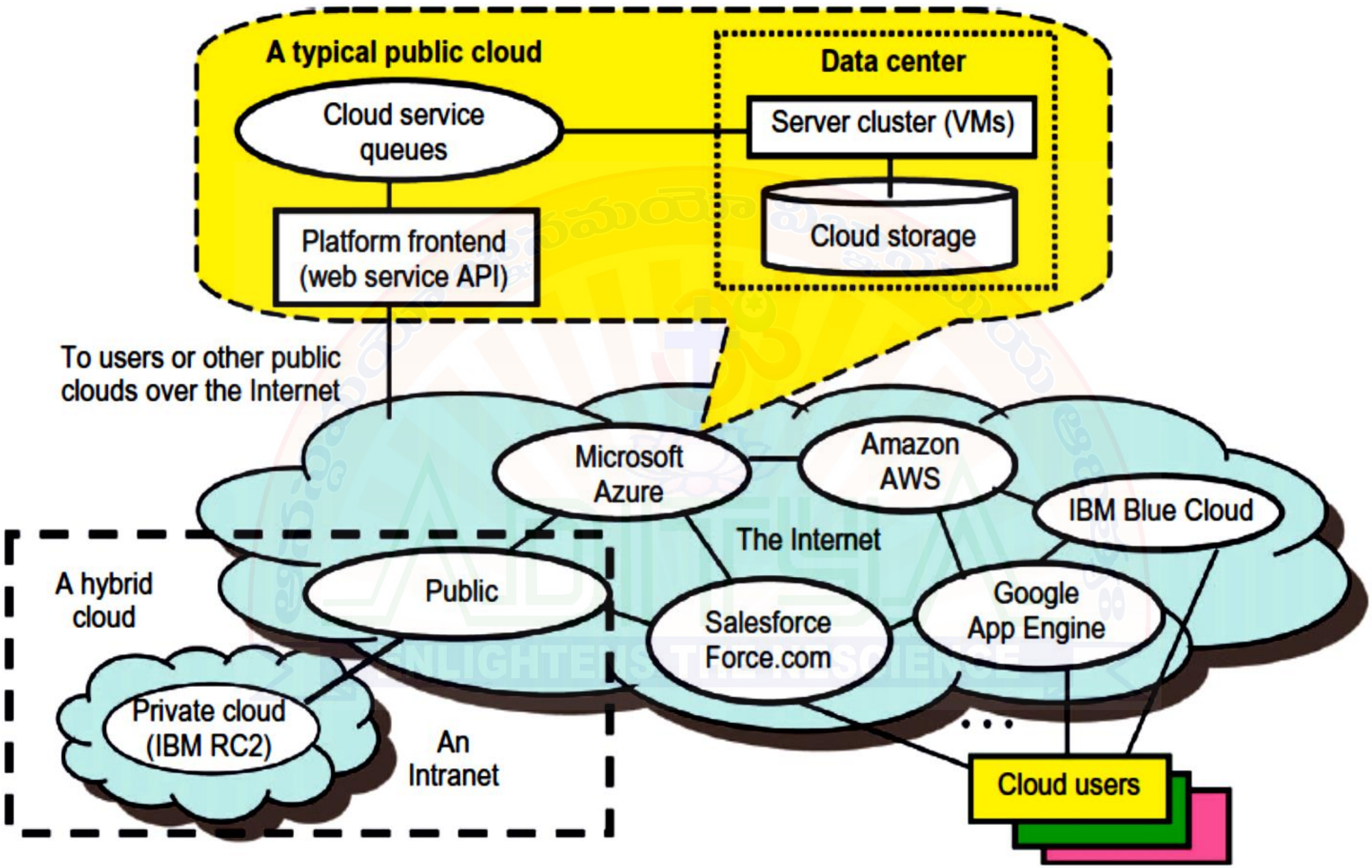


FIGURE 4.1

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

Public Clouds vs. Private Clouds

Characteristics	Public clouds	Private clouds
Technology leverage and ownership	Owned by service providers	Leverage existing IT infrastructure and personnel; owned by individual organization
Management of provisioned resources	Creating and managing VM instances within proprietary infrastructure; promote standardization, preserves capital investment, application flexibility	Client managed; achieve customization and offer higher efficiency
Workload distribution methods and loading policies	Handle workload without communication dependency; distribute data and VM resources; surge workload is off-loaded	Handle workload dynamically, but can better balance workloads; distribute data and VM resources
Security and data privacy enforcement	Publicly accessible through remote interface	Access is limited; provide pre-production testing and enforce data privacy and security policies
Example platforms	Google App Engine, Amazon AWS, Microsoft Azure	IBM RC2

Public, Private & Hybrid Clouds

Data-Center Networking Structure

- The core of a cloud is the server cluster (or VM cluster). Cluster nodes are used as compute nodes.
- A few control nodes are used to manage and monitor cloud activities.
- The scheduling of user jobs requires that you assign work to virtual clusters created for users.
- The gateway nodes provide the access points of the service from the outside world.
- These gateway nodes can be also used for security control of the entire cloud platform.
- In physical clusters and traditional grids, users expect static demand of resources.

Public, Private & Hybrid Clouds

Data-Center Networking Structure

- Data-center networks are mostly IP-based commodity networks, such as the 10 Gbps Ethernet network, which is optimized for Internet access.
- Figure 4.2 shows a multilayer structure for accessing the Internet. The server racks are at the bottom Layer 2, and they are connected through fast switches (S) as the hardware core.
- The data center is connected to the Internet at Layer 3 with many ccess routers (ARs) and border routers (BRs).

Public, Private & Hybrid Clouds

Data-Center Networking Structure

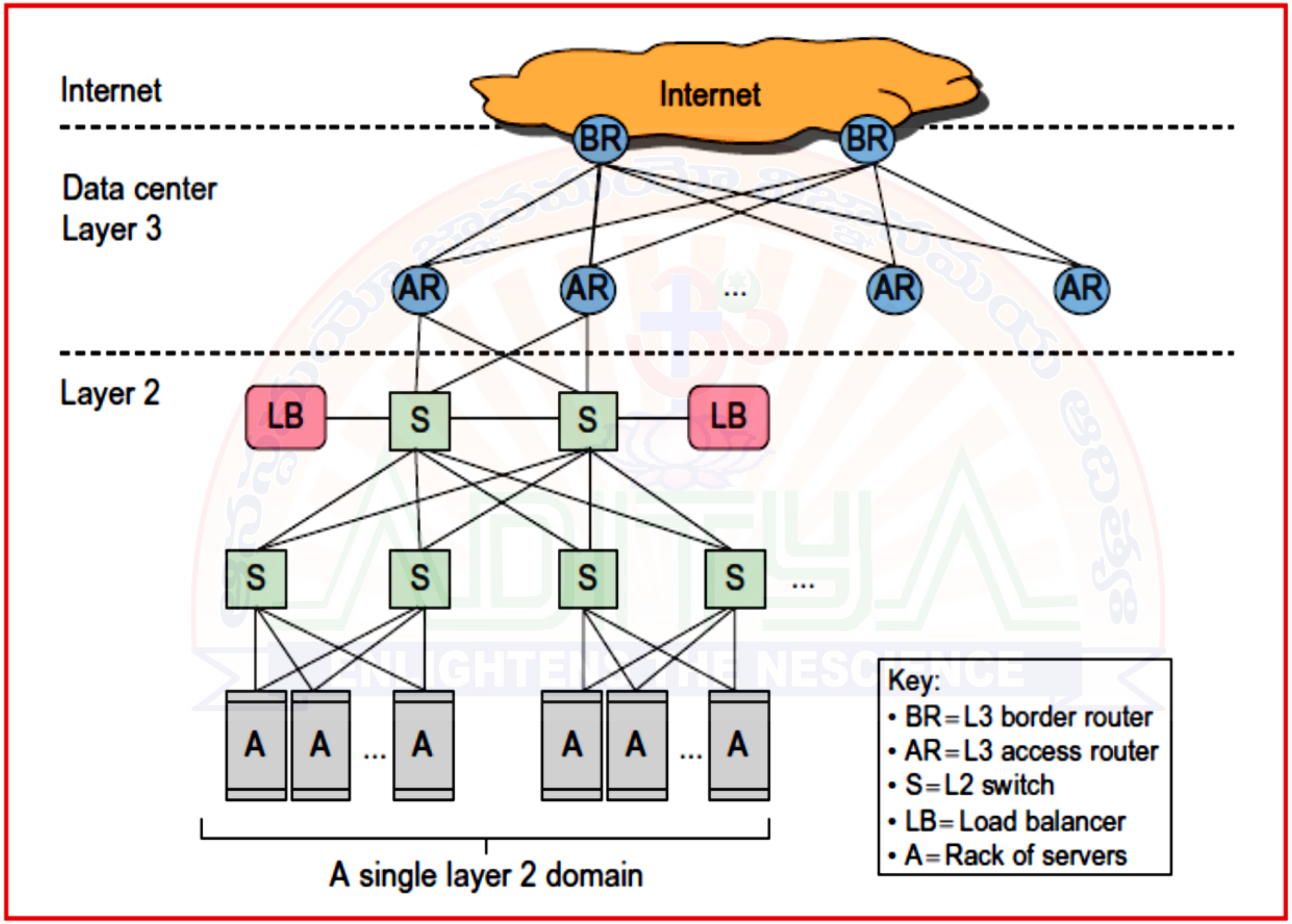


FIGURE 4.2

Standard data-center networking for the cloud to access the Internet.

Public, Private & Hybrid Clouds

Data-Center Networking Structure

- Private clouds are easier to secure and more trustworthy within a company or organization.
- Once private clouds become mature and better secured, they could be open or converted to public clouds.
- Therefore, the boundary between public and private clouds could be blurred in the future.
- For Eg. An e-mail application can run in the service-access nodes and provide the user interface for outside users.
- The application can get the service from the internal cloud computing services.
- There are also some service nodes designed to support the proper functioning of cloud computing clusters.
- There will be some independent service nodes that provide independent services for other nodes in the cluster. For eg, a news service need geographical information under service-access nodes.

Cloud Ecosystem and Enabling Technologies

- Cloud computing platforms differ from conventional computing platforms in many aspects.
- The traditional computing model is specified below by the process on the left, which involves buying the hardware, acquiring the necessary system software, installing the system, testing the configuration, and executing the application code and management of resources.
- What is even worse is that this cycle repeats itself in about every 18 months, meaning the machine we bought becomes obsolete every 18 months.
- computing model follows a payas-you-go model.

Cloud Ecosystem and Enabling Technologies

- Therefore the cost is significantly reduced, because we simply rent computer resources without buying the computer in advance.
- All hardware and software resources are leased from the cloud provider without capital investment on the part of the users.
- Only the execution phase costs some money.
- The experts at IBM have estimated that an 80 percent to 95 percent saving results from cloud computing, compared with the conventional computing paradigm.
- This is very much desired, especially for small businesses, which requires limited computing power and thus avoid the purchase of expensive computers or servers repeatedly every few years.

Cloud Ecosystem and Enabling Technologies

Classical Computing	**Cloud Computing**
(Repeat the following cycle every 18 months)	(Pay as you go per each service provided)
Buy and own	**Subscribe**
Hardware, system software, applications to meet peak needs	- - - -
Install, configure, test, verify, evaluate, manage	**Use** (Save about 80-95% of the total cost)
- - - -	- - - -
Use	(Finally)
- - - -	**$ - Pay for what you use**
Pay $$$$$ (High cost)	based on the QoS

Cloud Ecosystem and Enabling Technologies

Cloud Design Objectives

- **Shifting computing from desktops to data centers** Computer processing, storage, and software delivery is shifted away from desktops and local servers and toward data centers over the Internet.
- **Service provisioning and cloud economics** Providers supply cloud services by signing SLAs with consumers and end users. The services must be efficient in terms of computing, storage, and power consumption. Pricing is based on a pay-as-you-go policy.
- **Scalability in performance** The cloud platforms and software and infrastructure services must be able to scale in performance as the number of users increases.

Cloud Ecosystem and Enabling Technologies

Cloud Design Objectives

- **Data privacy protection** Can you trust data centers to handle your private data and records? This concern must be addressed to make clouds successful as trusted services.

- **High quality of cloud services** The QoS of cloud computing must be standardized to make clouds interoperable among multiple providers.

- **New standards and interfaces** This refers to solving the data lock-in problem associated with data centers or cloud providers. Universally accepted APIs and access protocols are needed to provide high portability and flexibility of virtualized applications.

Cloud Ecosystem and Enabling Technologies

Cost Model

- In traditional IT computing, users must acquire their own computer and peripheral equipment as capital expenses.
- In addition, they have to face operational expenditures in operating and maintaining the computer systems, including personnel and service costs.
- The fixed cost is the main cost, and that it could be reduced slightly as the number of users increases.
- However, the operational costs may increase sharply with a larger number of users.
- Therefore, the total cost escalates quickly with massive numbers of users.
- On the other hand, cloud computing applies a pay-per-use business model, in which user jobs are outsourced to data centers.

Cloud Ecosystem and Enabling Technologies

Cost Model

- Overall, cloud computing will reduce computing costs significantly for both small users and large enterprises.
- Computing economics does show a big gap between traditional IT users and cloud users.
- The savings in acquiring expensive computers up front releases a lot of burden for startup companies.
- The fact that cloud users only pay for operational expenses and do not have to invest in permanent equipment is especially attractive to massive numbers of small users.
- This is a major driving force for cloud computing to become appealing to most enterprises and heavy computer users.

Cloud Ecosystem and Enabling Technologies

Cost Model

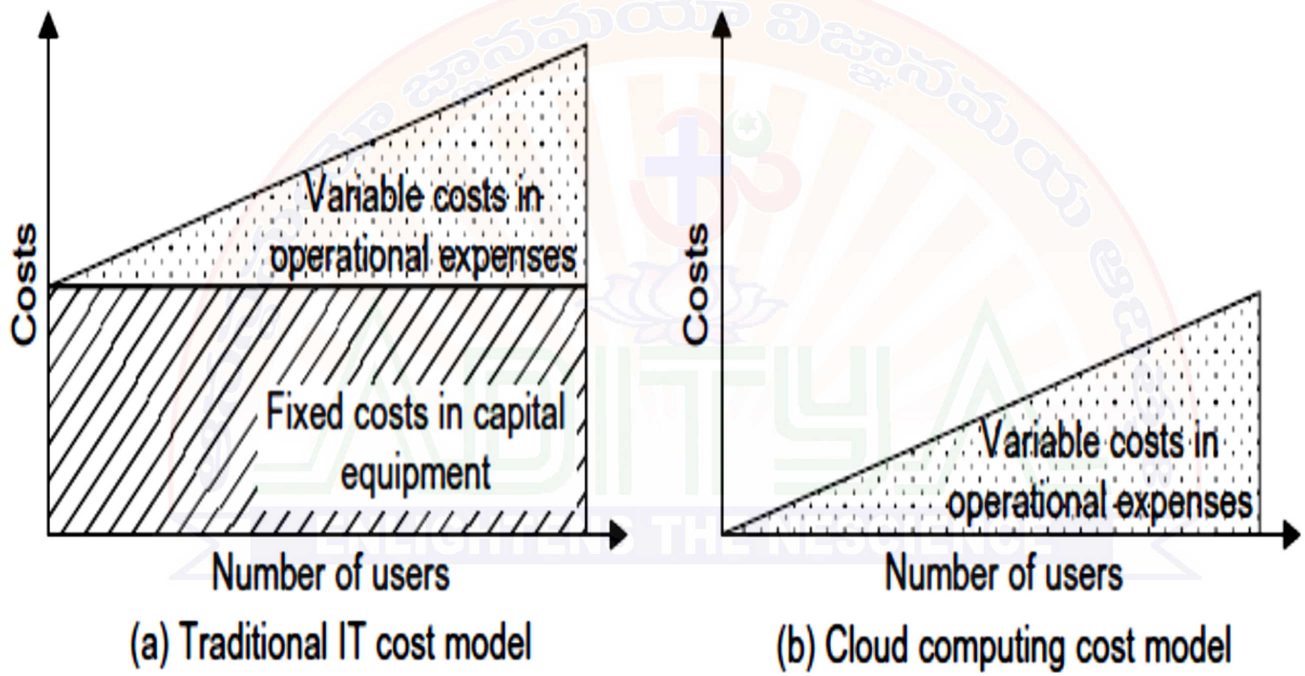


FIGURE 4.3

Computing economics between traditional IT users and cloud users.

Cloud Ecosystem and Enabling Technologies

Cloud Ecosystems

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

Cloud Ecosystem and Enabling Technologies

Cloud Ecosystems

- An ecosystem was suggested by Sotomayor et al. for building private clouds.
- They suggested four levels of ecosystem development in a private cloud.
- At the user end, consumers demand a flexible platform.
- At the cloud management level, the cloud manager provides virtualized resources over an IaaS platform.
- At the virtual infrastructure (VI) management level, the manager allocates VMs over multiple server clusters. Finally, at the VM management level, the VM managers handle VMs installed on individual host machines.
- An ecosystem of cloud tools attempts to span both cloud management and VI management.
- Integrating these two layers is complicated by the lack of open and standard interfaces between them.

Cloud Ecosystem and Enabling Technologies

Cloud Ecosystems

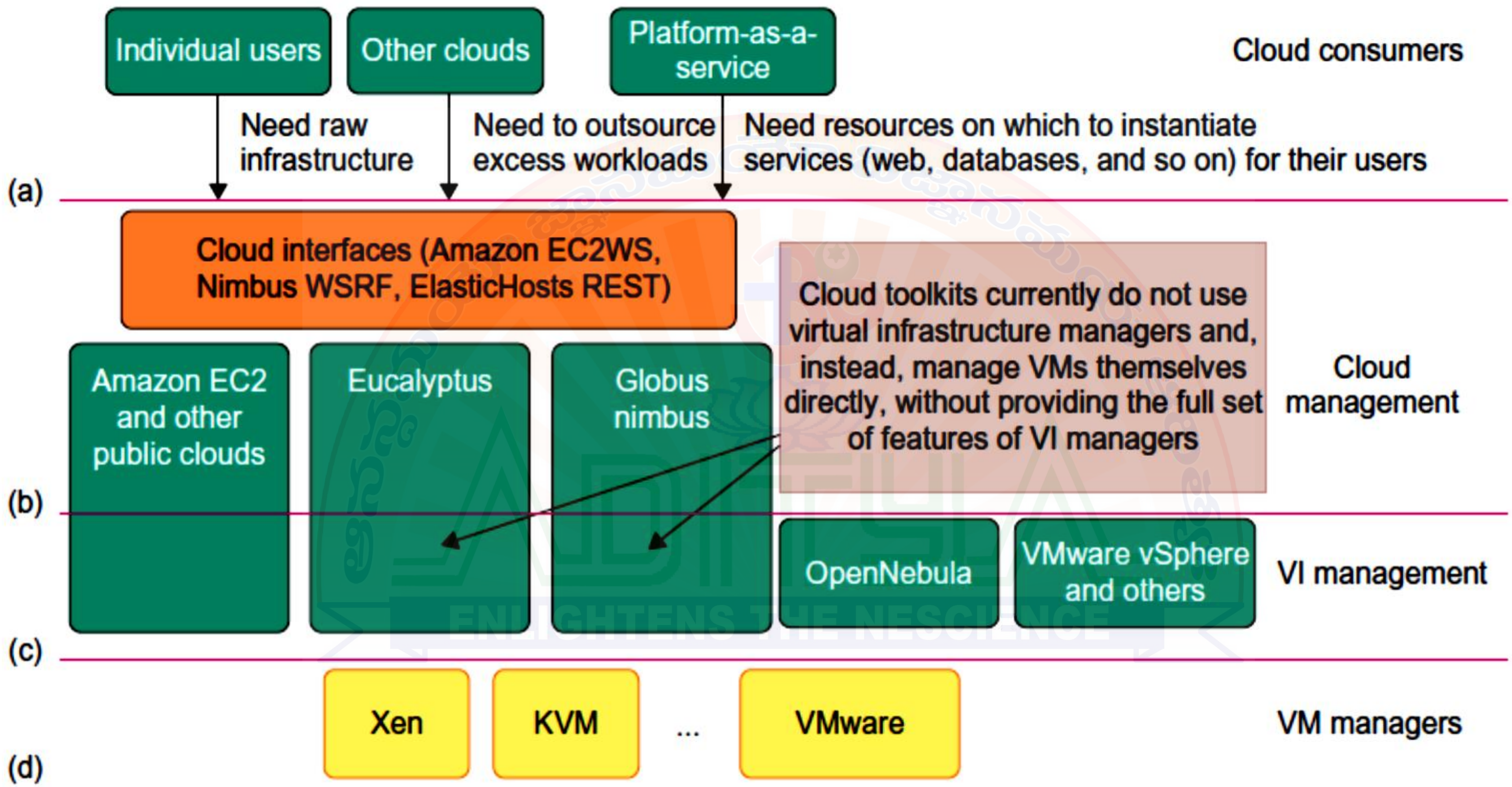


FIGURE 4.4

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.

Cloud Ecosystem and Enabling Technologies

Cloud Ecosystems

- An increasing number of startup companies are now basing their IT strategies on cloud resources, spending little or no capital to manage their own IT infrastructures.
- We desire a flexible and open architecture that enables organizations to build private/hybrid clouds. VI management is aimed at this goal.
- Example VI tools include oVirt, vSphere/4 from VMWare, and VM Orchestrator from platform Computing.
- These tools support dynamic placement and VM management on a pool of physical resources, automatic load balancing, server consolidation, and dynamic infrastructure resizing and partitioning.
- In addition to public clouds such as Amazon EC2, Eucalyptus and Globus Nimbus are open source tools for virtualization of cloud infrastructure.

Cloud Ecosystem and Enabling Technologies

Surge of Private Clouds

- In general, private clouds leverage existing IT infrastructure and personnel within an enterprise or government organization.
- Both public and private clouds handle workloads dynamically.
- However, public clouds should be designed to handle workloads without communication dependency.
- Both types of clouds distribute data and VM resources. However, private clouds can balance workloads to exploit IT resources more efficiently within the same intranet.
- Private clouds can also provide preproduction testing and enforce data privacy and security policies more effectively. In a public cloud, the surge workload is often offloaded.

Cloud Ecosystem and Enabling Technologies

Surge of Private Clouds

- Most companies start with virtualization of their computing machines to lower the operating costs.
- Companies such as Microsoft, Oracle, and SAP may want to establish policy-driven management of their computing resources, mainly to improve QoS to their employees and customers.
- By integrating virtualized data centers and company IT resources, they offer IT as a service to improve the agility of their company operations.
- This approach avoids replacement of a large number of servers every 18 months.
- As a result, these companies can upgrade their IT efficiency significantly.

Infrastructure-as-a-Service (IaaS)

Infrastructure as a Service

- Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-based services in a pay-as-you-go model to consumers.
- The services provided over the cloud can be generally categorized into three different service models: namely
- IaaS, Platform as a Service (PaaS), and Software as a Service (SaaS).
- These form the three pillars on top of which cloud computing solutions are delivered to end users.
- All three models allow users to access services over the Internet, relying entirely on the infrastructures of cloud service providers.

Infrastructure-as-a-Service (IaaS)

Infrastructure as a Service

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

Infrastructure-as-a-Service (IaaS)

Infrastructure as a Service

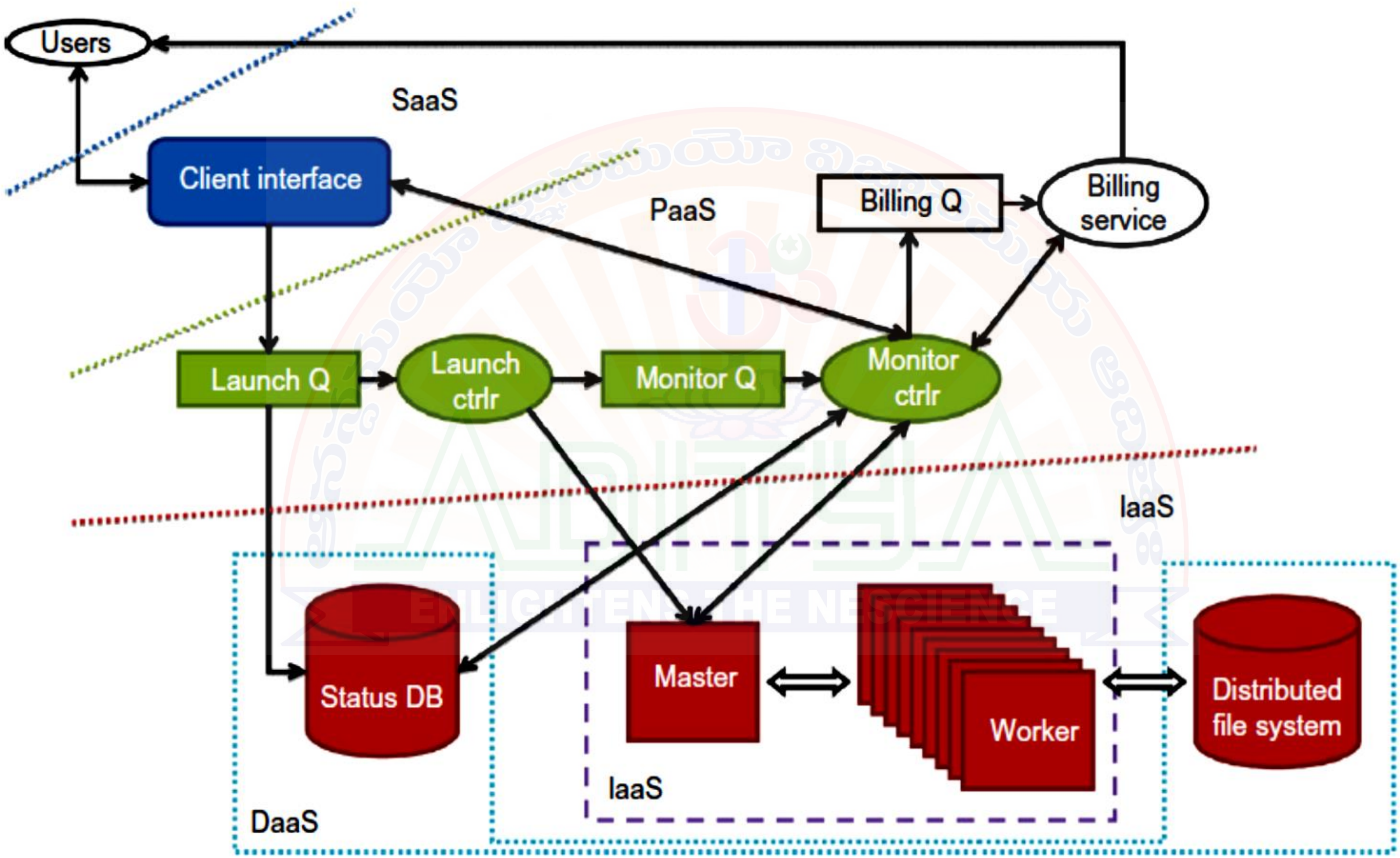


FIGURE 4.5

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

Infrastructure-as-a-Service (IaaS)

Infrastructure as a Service

Table 4.1 Public Cloud Offerings of IaaS [10,18]

Cloud Name	VM Instance Capacity	API and Access Tools	Hypervisor, Guest OS
Amazon EC2	Each instance has 1–20 EC2 processors, 1.7–15 GB of memory, and 160–1.69 TB of storage.	CLI or web Service (WS) portal	Xen, Linux, Windows
GoGrid	Each instance has 1–6 CPUs, 0.5–8 GB of memory, and 30–480 GB of storage.	REST, Java, PHP, Python, Ruby	Xen, Linux, Windows
Rackspace Cloud	Each instance has a four-core CPU, 0.25–16 GB of memory, and 10–620 GB of storage.	REST, Python, PHP, Java, C#, .NET	Xen, Linux
FlexiScale in the UK	Each instance has 1–4 CPUs, 0.5–16 GB of memory, and 20–270 GB of storage.	web console	Xen, Linux, Windows
Joyent Cloud	Each instance has up to eight CPUs, 0.25–32 GB of memory, and 30–480 GB of storage.	No specific API, SSH, Virtual/Min	OS-level virtualization, OpenSolaris

Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)

Platform as a Service (PaaS)

- To be able to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment.
- Such a platform includes operating system and runtime library support.
- This has triggered the creation of the PaaS model to enable users to develop and deploy their user applications.
- The platform cloud is an integrated computer system consisting of both hardware and software infrastructure.
- The user application can be developed on this virtualized cloud platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET).

Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)

Platform as a Service (PaaS)

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

Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)

Software as a Service (SaaS)

- The SaaS model provides software applications as a service.
- As a result, on the customer side, there is no upfront investment in servers or software licensing.
- On the provider side, costs are kept rather low, compared with conventional hosting of user applications.
- Customer data is stored in the cloud that is either vendor proprietary or publicly hosted to support PaaS and IaaS.
- The best examples of SaaS services include Google Gmail and docs, Microsoft SharePoint, and the CRM software from Salesforce.com.
- They are all very successful in promoting their own business or are used by thousands of small businesses in their day-to-day operations.

PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

Public Clouds and Service Offerings

- Cloud services are demanded by computing and IT administrators, software vendors, and end users.
- At the top level, individual users and organizational users demand very different services.
- The application providers at the SaaS level serve mainly individual users.
- Most business organizations are serviced by IaaS and PaaS providers.
- The infrastructure services (IaaS) provide compute, storage, and communication resources to both applications and organizational users.
- The cloud environment is defined by the PaaS or platform providers.

Public Clouds and Service Offerings

- Cloud services rely on new advances in machine virtualization, SOA, grid infrastructure management, and power efficiency.
- Consumers purchase such services in the form of IaaS, PaaS, or SaaS.
- Also, many cloud entrepreneurs are selling value-added utility services to massive numbers of users.
- The cloud industry leverages the growing demand by many enterprises and business users to outsource their computing and storage jobs to professional providers.

Public Clouds and Service Offerings

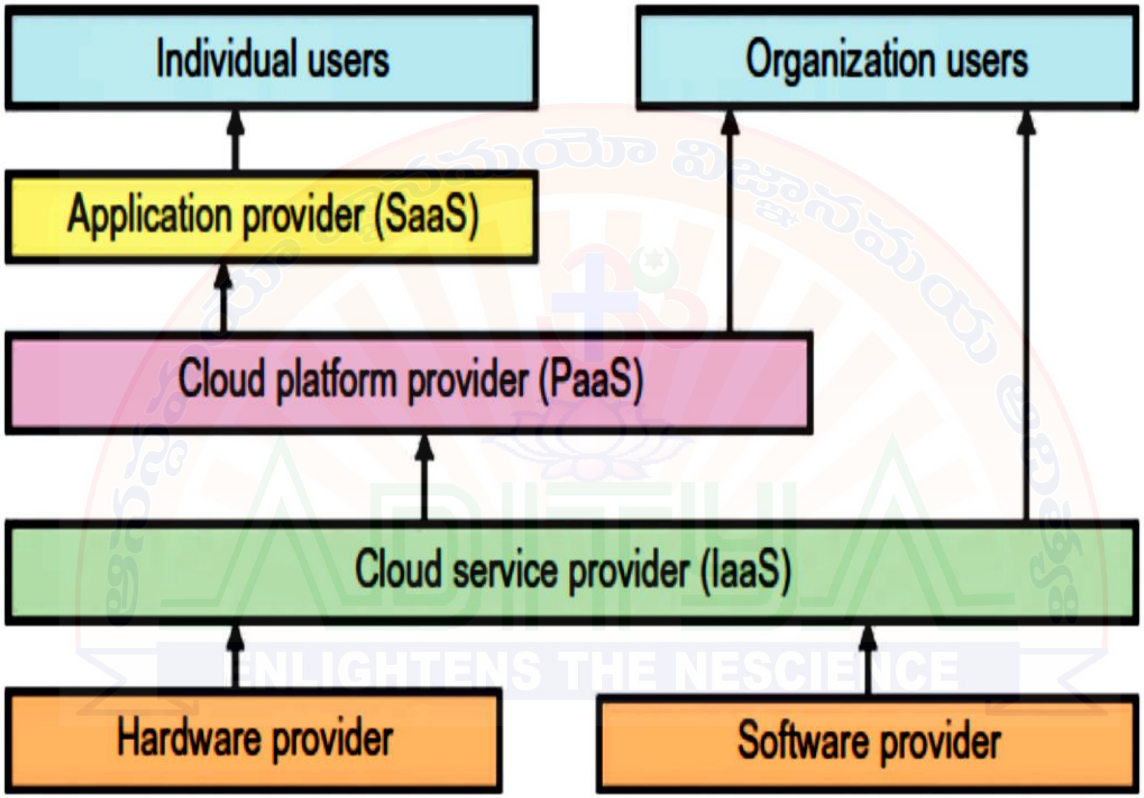


FIGURE 4.19

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

Public Clouds and Service Offerings

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.*

Google App Engine (GAE)

- Google has the world's largest search engine facilities.
- The company has extensive experience in massive data processing that has led to new insights into data-center design and novel programming models that scale to incredible sizes.
- The Google platform is based on its search engine expertise, but with MapReduce, this infrastructure is applicable to many other areas.
- Google has hundreds of data centers and has installed more than 460,000 servers worldwide.
- For example, 200 Google data centers are used at one time for a number of cloud applications.

Google App Engine (GAE)

Google Cloud Infrastructure

- Google has pioneered cloud development by leveraging the large number of data centers it operates.
- For example, Google pioneered cloud services in Gmail, Google Docs, and Google Earth, among other applications.
- These applications can support a large number of users simultaneously with HA.
- Notable technology achievements include the Google File System (GFS), MapReduce, BigTable, and Chubby (lock service).
- In 2008, Google announced the GAE web application platform which is becoming a common platform for many small cloud service providers.

Google App Engine (GAE)

GAE Architecture

- GFS is used for storing large amounts of data.
- MapReduce is for use in application program development.
- Chubby is used for distributed application lock services.
- BigTable offers a storage service for accessing structured data.
- Users can interact with Google applications via the web interface provided by each application.
- Third-party application providers can use GAE to build cloud applications for providing services.
- The applications all run in data centers under tight management by Google engineers.
- Inside each data center, there are thousands of servers forming different clusters.

Google App Engine (GAE)

GAE Architecture

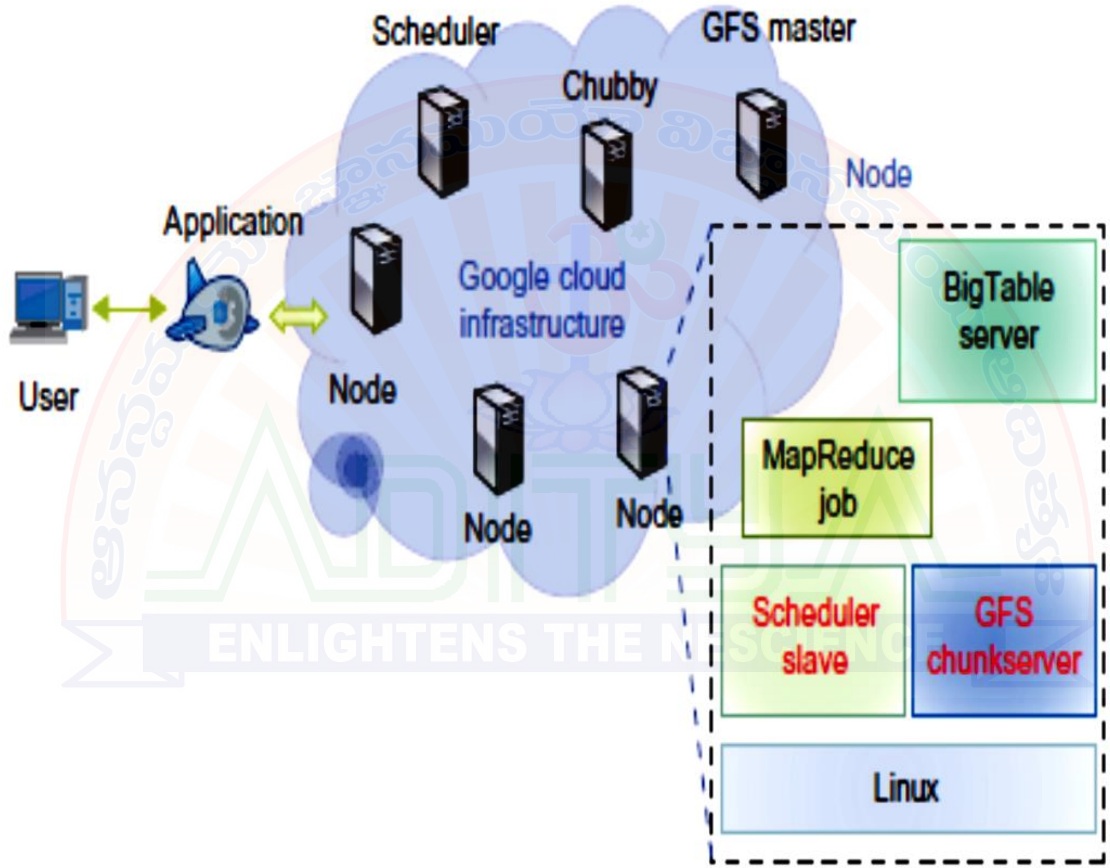


FIGURE 4.20

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

Google App Engine (GAE)

Functional Modules of GAE

- The GAE is not an infrastructure platform, but rather an application development platform for users. The GAE platform comprises the following five major components.
- **The datastore** offers object-oriented, distributed, structured data storage services based on BigTable techniques. The datastore secures data management operations.
- **The application runtime environment** offers a platform for scalable web programming and execution. It supports two development languages: Python and Java.

Google App Engine (GAE)

Functional Modules of GAE

- **The software development kit (SDK)** is used for local application development. The SDK allows users to execute test runs of local applications and upload application code.
- **The administration console** is used for easy management of user application development cycles, instead of for physical resource management.
- **The GAE web service** infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources by GAE.

Google offers essentially free GAE services to all Gmail account owners. You can register for a GAE account or use your Gmail account name to sign up for the service. The service is free within a quota. If you exceed the quota, the page instructs you on how to pay for the sevice.

Google App Engine (GAE)

GAE Applications

- Well-known GAE applications include the Google Search Engine, Google Docs, Google Earth, and Gmail.
- These applications can support large numbers of users simultaneously.
- Users can interact with Google applications via the web interface provided by each application.
- Third-party application providers can use GAE to build cloud applications for providing services.
- The applications are all run in the Google data centers.
- Web applications built on top of GAE can use the APIs authenticating users and sending e-mail using Google accounts.

Amazon Web Services (AWS)

- VMs can be used to share computing resources both flexibly and safely. Amazon has been a leader in providing public cloud services (http://aws.amazon.com/). Amazon applies the IaaS model in providing its services.

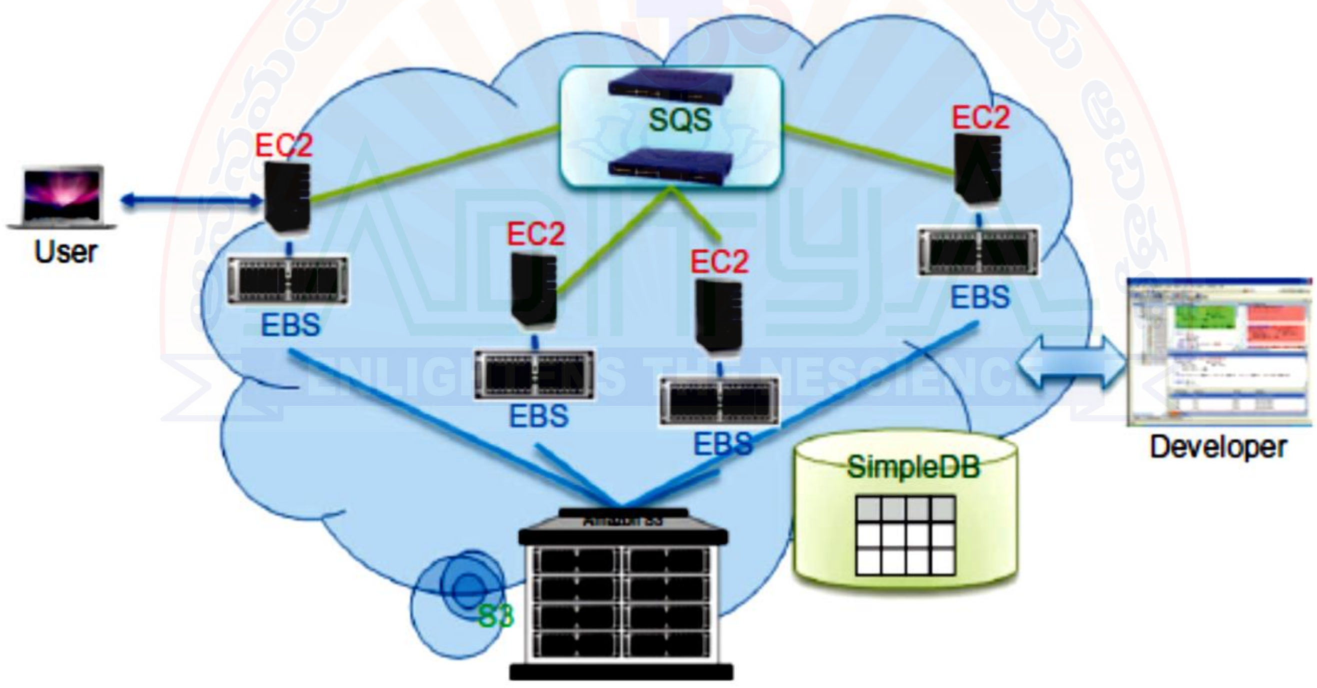


FIGURE 4.21

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

Amazon Web Services (AWS)

- EC2 provides the virtualized platforms to the host VMs where the cloud application can run.
- S3 (Simple Storage Service) provides the object-oriented storage service for users.
- EBS (Elastic Block Service) provides the block storage interface which can be used to support traditional applications.
- SQS stands for Simple Queue Service, and its job is to ensure a reliable message service between two processes.
- The message can be kept reliably even when the receiver processes are not running.
- Users can access their objects through SOAP with either browsers or other client programs which support the SOAP standard.

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

(Courtesy of Amazon, http://aws.amazon.com [3])

Microsoft Windows Azure

- In 2008, Microsoft launched a Windows Azure platform to meet the challenges in cloud computing.
- This platform is built over Microsoft data centers.
- The platform is divided into three major component platforms.
- Windows Azure offers a cloud platform built on Windows OS and based on Microsoft virtualization technology.
- Applications are installed on VMs deployed on the data-center servers.
- Azure manages all servers, storage, and network resources of the data center.
- On top of the infrastructure are the various services for building different cloud applications.

Microsoft Windows Azure

- Cloud-level services provided by the Azure platform are introduced below :
- **Live service** Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently.
- **.NET service** This package supports application development on local hosts and execution on cloud machines.
- **SQL Azure** This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud.
- **SharePoint service** This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services.
- **Dynamic CRM service** This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions.

Microsoft Windows Azure

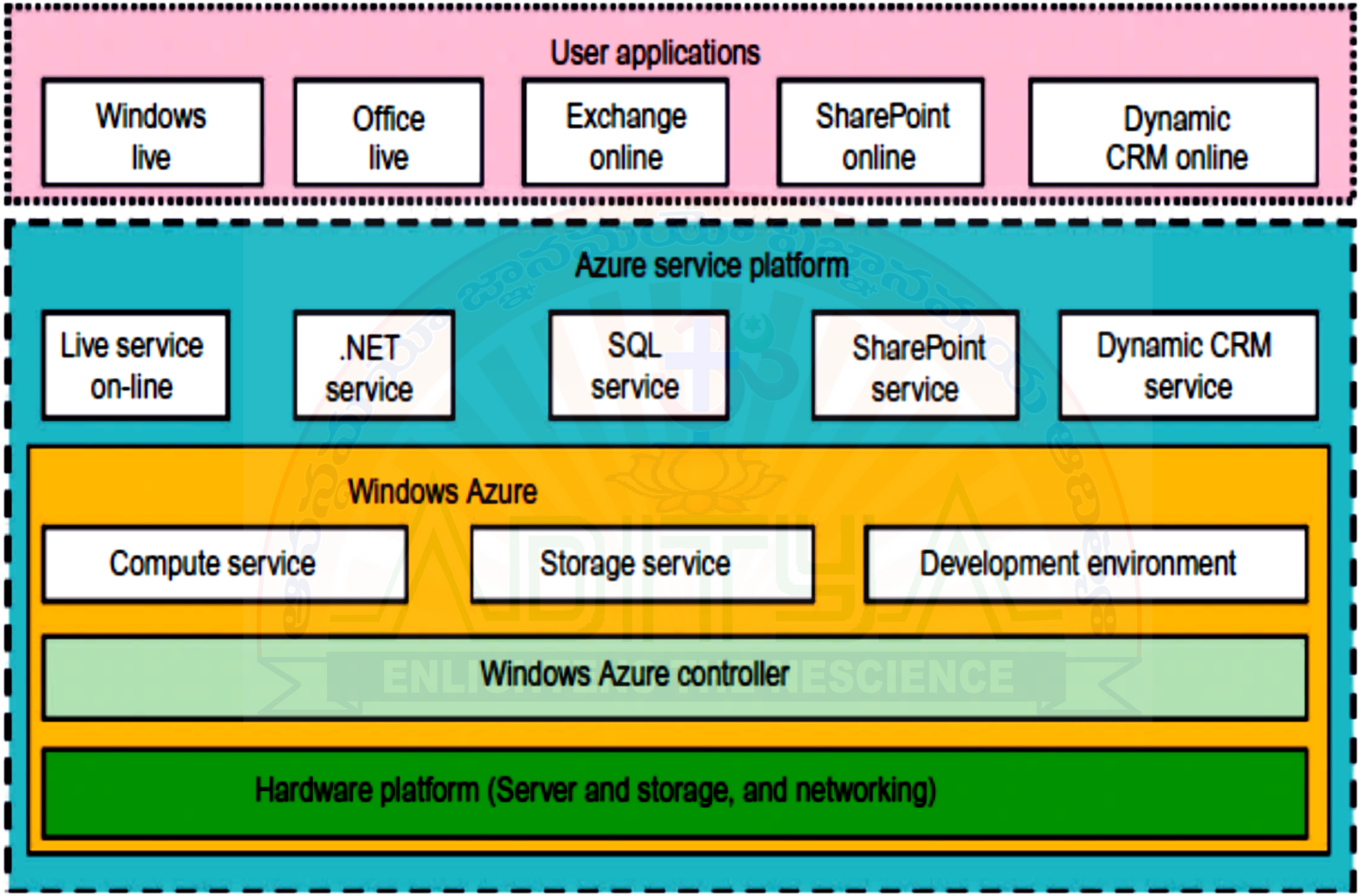


FIGURE 4.22

Microsoft Windows Azure platform for cloud computing.