CPET 581 Cloud Computing: Technologies and Enterprise IT Strategies

Lecture 1

Overview of Distributed and Cloud Computing System Models and Enabling Technologies

Based on the Chapter 1 Distributed System Models and Enabling Technologies of the Text Book: Distributed and Cloud Computing, by K. Hwang, G C. Fox, and J.J. Dongarra, published Elsevier/Morgan Kaufmann, 2012.

Spring 2015

A Specialty Course for Purdue University's M.S. in Technology Graduate Program: IT/Advanced Computer App Track

Paul I-Hai Lin, Professor

Dept. of Computer, Electrical and Information Technology

Purdue University Fort Wayne Campus

The Evolution of Computer Systems and Applications

- Computer History Museum, http://www.computerhistory.org/
  - Early computer companies
  - Analog computers
  - Mainframe computers
    - Time-sharing
    - Real-time computing
  - Supercomputers
  - Minicomputers
  - Networking
  - Personal computers
  - Mobile computing

The Evolution of Computer Systems and Applications (cont.)

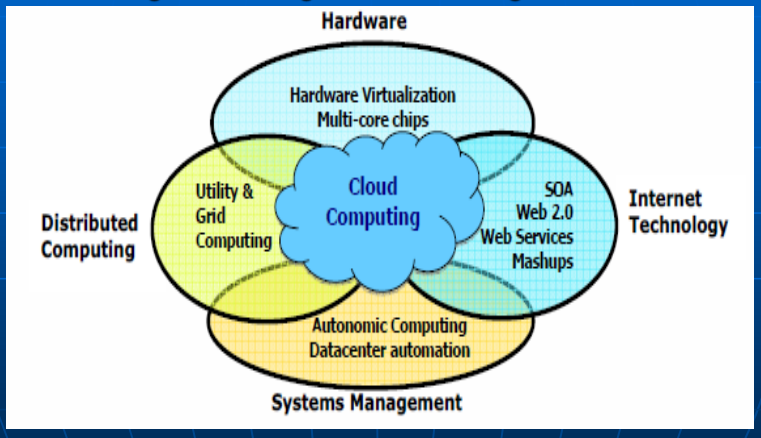
- Client-Server Computing
- Distributed Computing
- Virtualization and data centers
- Utility Computing
- Grid Computing
- Internet computing
- Web services
- Service-Oriented Computing (SOA)
- Mobile Computing
- Cloud Computing

Data Deluge

- The Coming Data Deluge, by Paul McFedries, Feb. 2011, IEEE Spectrum, http://spectrum.ieee.org/at-work/innovation/the-coming-data-deluge
  - Small data
  - Big data
  - Data deluge
  - Data-intensive science – a new, 4th paradigm for scientific exploration, e-Science
- Bridging the Data Deluge Gap, by Eric Savitz, 2012/8/23 – Forbes, http://www.forbes.com/sites/ciocentral/2012/08/23/bridging-the-data-deluge-gap/
  - Net information flow is growing faster than the IT investment required to "Store", "Transmit, "Analyze," and "Manage" it
  - Where are the these bottlenecks?

Coping with Data Deluge

- Data Deluge Enabling New Challenges



(Courtesy of Judy Qiu, Indiana University, 2011)

Interactions among 4 technical challenges
(source: Judy Qiu, Indiana University, 2011)

- Data Deluge
- Cloud Technology
- eScience,
  - Computational intensive science that is carried out in highly distributed network environments, or science that uses immense data sets that require Grid Computing. (source: http://en.wikipedia.org/wiki/E-Science)
  - Microsoft Research: eScience, http://research.microsoft.com/en-us/groups/escience/
- Multicore/Parallel Computing

(Courtesy of Judy Qiu, Indiana University, 2011)

1.1 Scalable Internet-based Computing

- General Computing Trend
  - Leverage shared web resources
  - Massive amount of data over the Internet
- High Performance Computing (HPC)
  - Supercomputers (massively parallel processors, MPP)
  - Clusters of cooperative computers; share computing resources
  - Physically connected in close range to one another
- High Throughput Computing (HTC)

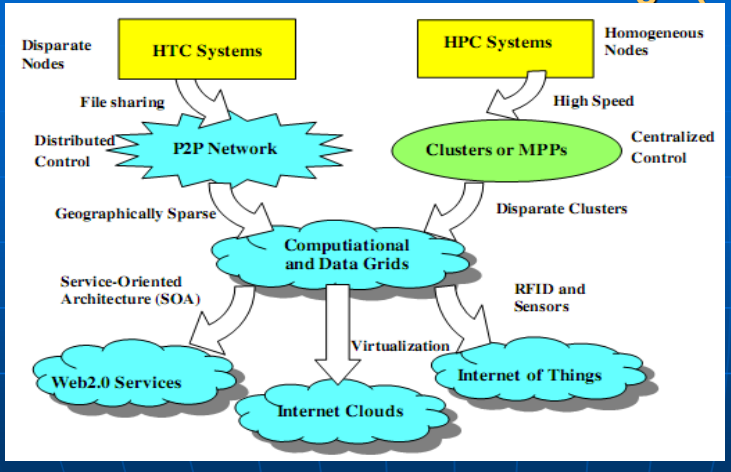
1.1 Scalable Internet-based Computing

- High Throughput Computing (HTC) & Applications
  - Peer-to-peer (P2P) networks – distributed file sharing and content delivery applications
  - Web service platforms
  - Cloud computing
- HTC Technologies
  - Improved batch processing speed
  - Address acute problems at many data and enterprise computing centers
    - Cost, Energy saving, Security, Reliability

1.1 Scalable Internet-based Computing

- Three New Computing Paradigms
  - Web 2.0 Services
  - Internet Clouds
  - Internet of Things
- Computing Paradigm Distinction
  - Centralized computing
  - Parallel computing
  - Distributed computing
  - Cloud computing

Cloud and Internet of Things (IOT)



Source: K. Hwang, G. Fox, and J. Dongarra, *Distributed and Cloud Computing,* Morgan Kaufmann, 2012.

HPC: High-Performance Computing

HTC: High-Throughput Computing

P2P: Peer to Peer

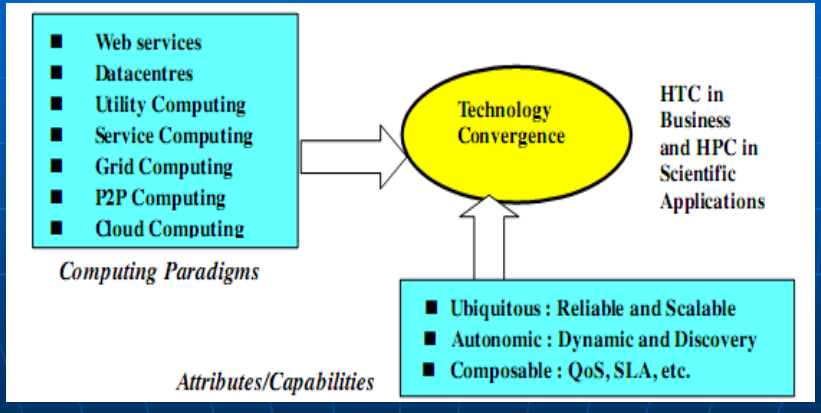
MPP: Massively Parallel Processors

1.1 Scalable Internet-based Computing

- Degrees of Parallelism
  - Bit-level parallelism (BLP)
  - Instruction-level parallelism (ILP)
    - Pipelining, Super scalar computing, VLIW (very long instruction word) architecture, Multithreading
  - Data-level parallelism (DLP
    - Single-instruction multiple data (SIMD)
  - Task-level parallelism (TLP)
    - Multicore processor and Chip Multiprocessors (CMPs)
  - Job-level parallelism (JLP)

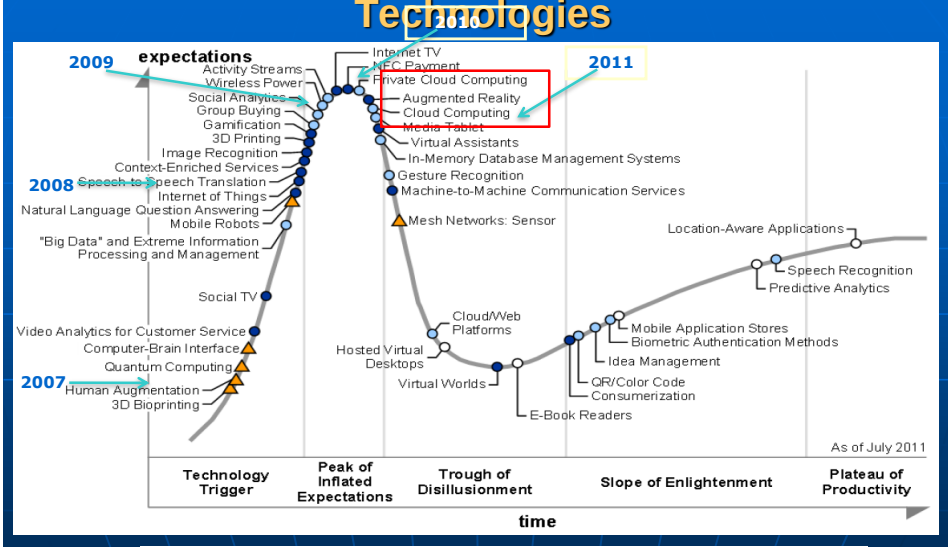
1.1 Scalable Internet-based Computing

- HPC for Science and HTC for Business Applications



(Courtesy of Raj Buyya, University of Melbourne, 2011)

2011 Gartner "IT Hype Cycle" for Emerging Technologies



13

1.2 Technologies for Network Based Systems

- Multicore CPUs and Multithreading Technologies
- GPU Computing (Graphics Co-processor)
- Memory, Storage, and Wide Area Networking
- Virtual Machines and Virtualization Middleware
- Data Center Virtualization for Cloud Computing

1.2 Technologies for Network Based Systems

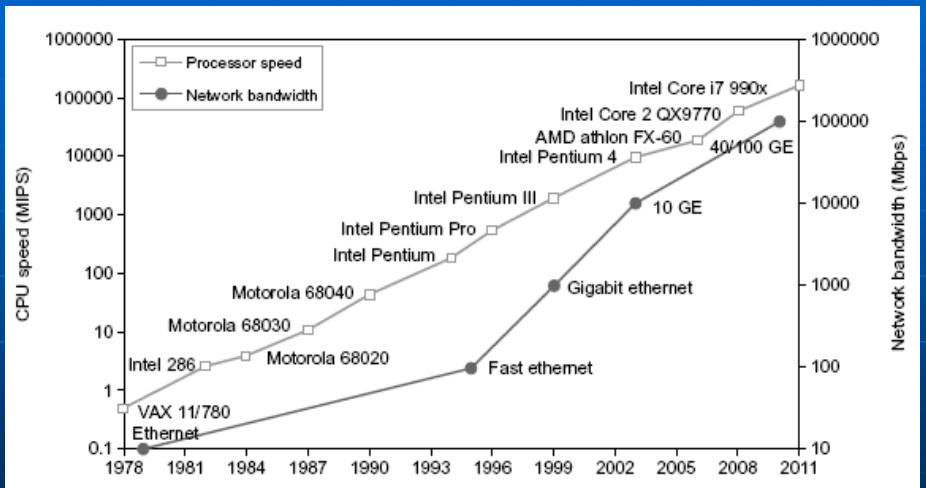


FIGURE 1.4

Improvement in processor and network technologies over 33 years.

Figure 1.5 Multicore Processor

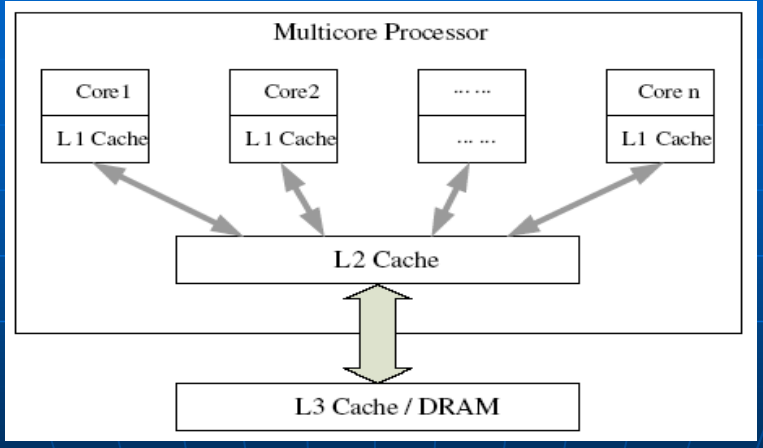


Figure 1.6 Multithreading Technologies

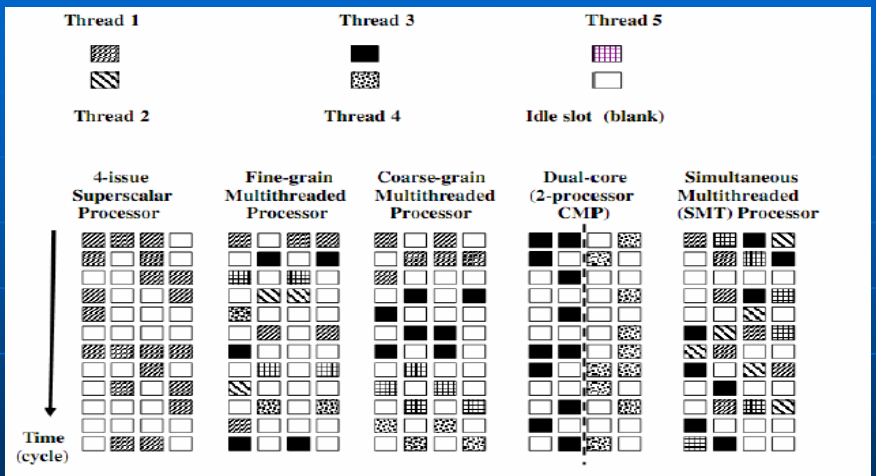


Figure 1. 8 Five micro-architectures that are current in use in modern processors that exploit both ILP and TLP supported by multicore and multithreading technologies

Figure 1.7 Architecture of a Many-Core Multiprocessor GPU Interacting with a CPU Processor

- CPU (Central Processor Unit)
- GPU (Graphics Processor Unit)

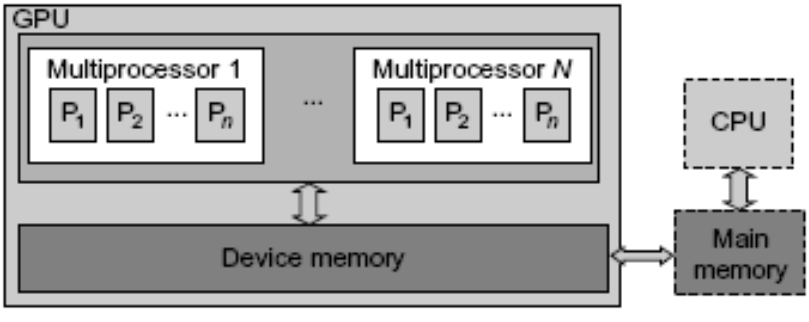
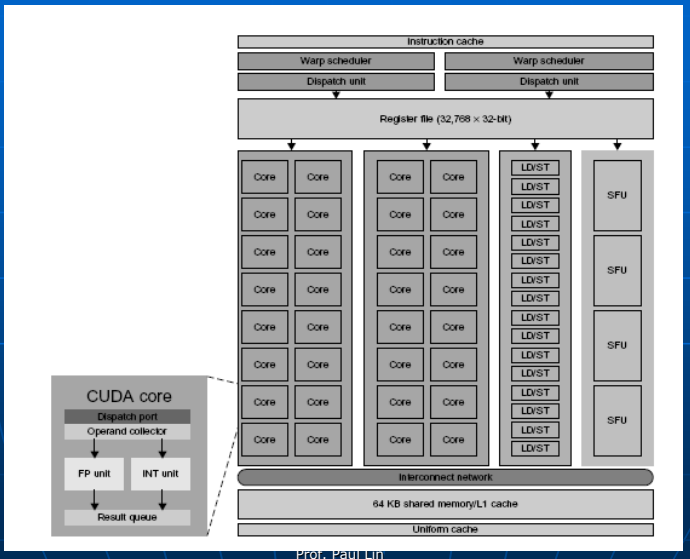


Figure 1.8 NVIDIA Fermi GPU built with 16 streaming multiprocessors (SMs) of 32 CUDA core each



Number Prefix Used in Computer Technologies

Number Prefix Used

Prefix	Symbol	Factor	
Yotta	Y	10^{24}	or E24
Zetta	Z	10^{21}	or E21
Exa	E	10^{18}	or E18
Peta	P	10^{15}	or E15
Tera	T	10^{12}	or E12
Giga	G	10^{9}	or E9
Mega	M	10^{6}	or E6
Kilo	k	10^{3}	or E3
hecto	h	10^{2}	or E2
deca	da	10^{1}	or E1
deci	d	10^{-1}	or E-1
centi	c	10^{-2}	or E-2
milli	m	10^{-3}	or E-3

Number Prefix Used in Computer Technologies

Number Prefix Used

Prefix	Symbol	Factor	
micro	μ	10^{-6}	or E-6
nano	n	10^{-9}	or E-9
pico	p	10^{-12}	or E-12
femto	f	10^{-15}	or E-15
atto	a	10^{-18}	or E-18
zepto	z	10^{-21}	or E-21
yocto	y	10^{-24}	or E-24

Figure 1.10 Improvement in memory and disk technologies over 33 years

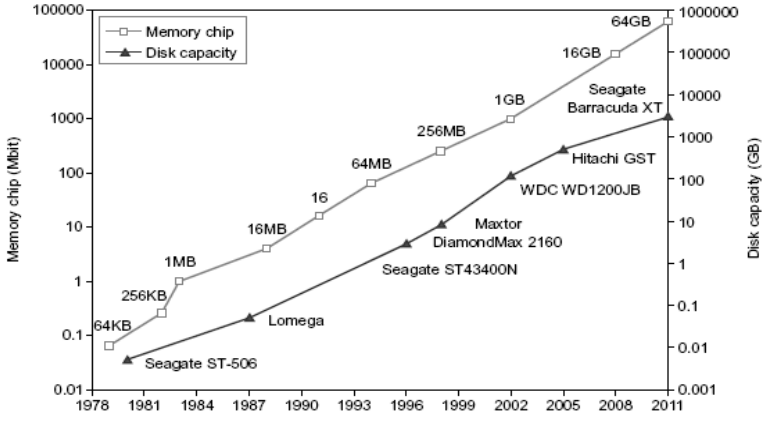


FIGURE 1.10

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

(Courtesy of Xiaosong Lou and Lizhong Chen of University of Southern California, 2011)

Figure 1.9 GPU, CPU Performance Comparison

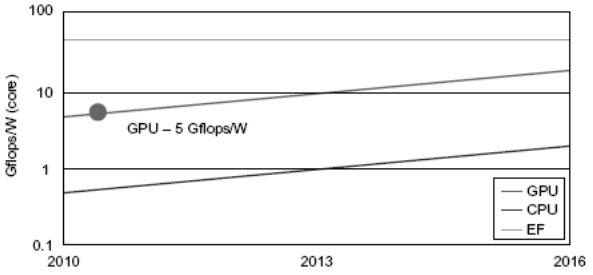
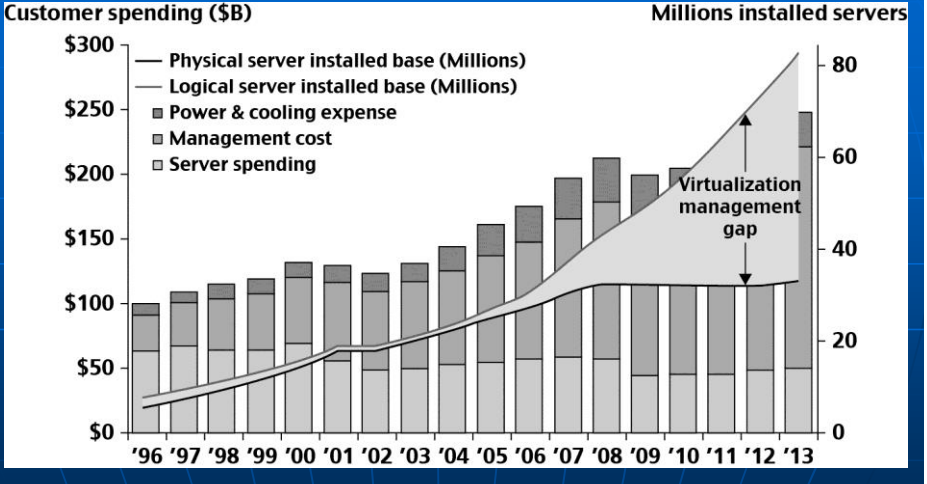


FIGURE 1.9

GPU and CPU performance in Gflops/Watt/core, compared with 60 Gflops/Watt/core projected in future Exascale systems.

Figure 1.14 Datacenter and server cost distribution



Low Cost Design: Datacenter

- IDC 2009 Datacenter Report
- Data Center Cost
  - 30% - purchasing IT equipment; 33% - Chillers
  - 18% - Uninterruptable power supply; 9% - computer room HVAC; 7% - power distribution, lighting, transformer costs
- Low-Cost Design Philosophy
  - About 60 percent of the cost is allocated to Management & Maintenance
  - The server purchase cost did not increase much with time
  - Use commodity switches and networks
  - Use commodity x86 servers
  - The software layer handles
    - Network traffic balancing
    - Fault tolerance
    - Expandability

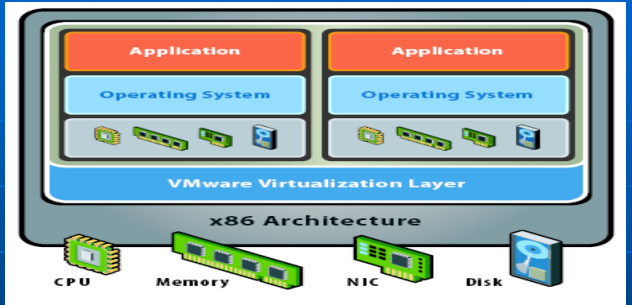
Datacenter Growth and Cost Breakdown

- U.S. Datacenter 2012-2016 Forecast (Doc # 237070)
  - From 2.94 million in 2012 to 2.89 million in 2016
  - From 611.4 million square feet in 2012 to more than 700 million square feet in 2016
- IDC Find Growth, Consolidation, and Changing Ownership Patterns in Worldwide Datacenter Forecast, Nov. 10, 2014, http://www.idc.com/getdoc.jsp?containerId=prUS25237514

Cloud Computing Enabling Technologies

- Convergence of Technologies
  1) Hardware virtualization and multi-core chips
  2) Utility and grid computing
  3) SOA (Service-Oriented Architecture), Web 2.0, and WS mashups (Web services)
  4) Atonomic computing and data center automation
- Microsoft Data Center, video, Microsoft Azure – Microsoft Cloud, 10:16 min, Feb. 16, 2014, https://www.youtube.com/watch?v=rUlDyhBc_Rg

Virtual Machine Architecture
(source VMWare, 2010)

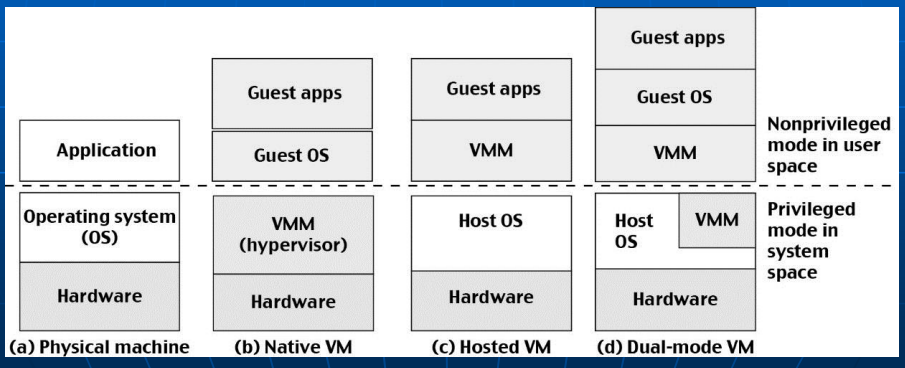


After Virtualization:

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

Figure 1.12 Three VM Architecture

- VMM – Virtual Machine Monitor
- (a) VMM (hypervisor) in the privileged mode; bare-metal VM



Primitive Operations in Virtual Machines

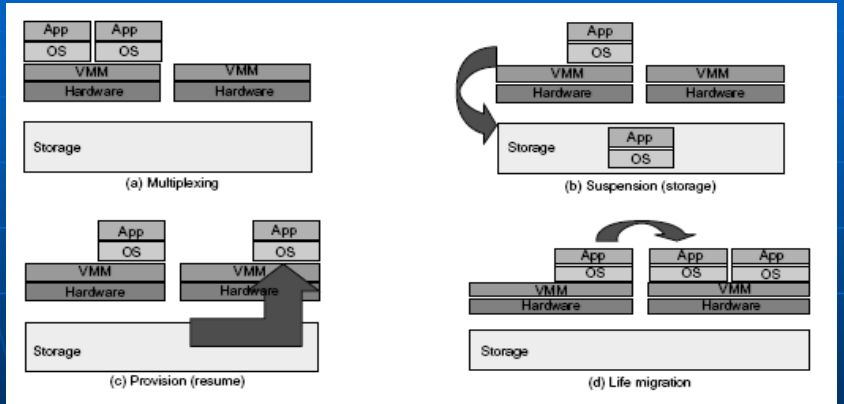


FIGURE 1.13
VM multiplexing, suspension, provision, and migration in a distributed computing environment.
(Courtesy of M. Rosenblum, Keynote address, ACM ASPLOS 2006 [41])

Concepts of Virtual Clusters

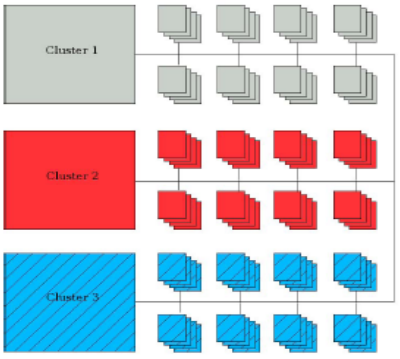


Fig. 1. A Campus Area Grid

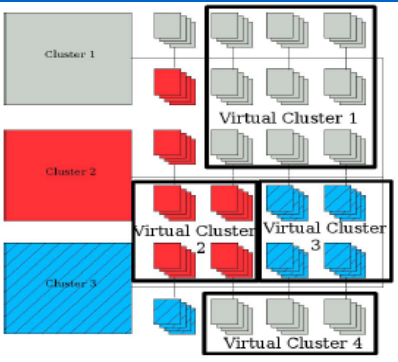


Fig. 2. Virtual machines in a cluster environment

(Source: W. Emeneker, et et al, "Dynamic Virtual Clustering with Xen and Moab, ISPA 2006, Springer-Verlag LNCS 4331, 2006, pp. 440-451)

1.3 System Models for Distributed and Cloud Computing Systems

- Four Groups of Massive Computer Systems: Clusters, P2P networks, computing grids, Internet clouds

Table 1.2 Classification of Distributed Parallel Computing Systems

Functionality, Applications	Multicomputer Clusters [27, 33]	Peer-to-Peer Networks [40]	Data/Computational Grids [6, 42]	Cloud Platforms [1, 9, 12, 17, 29]
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 datacenters via service-level agreement
Control and Resources Management	Homogeneous nodes with distributed control, running Unix or Linux	Autonomous client nodes, free in and out, with distributed self-organization	Centralized control, server oriented with authenticated security, and static resources	Dynamic resource provisioning of servers, storage, and networks over massive datasets
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 super-computing, global problem solving, and datacenter 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, and .NET	TeraGrid, GriPhyN, UK EGEE, D-Grid, ChinaGrid, etc.	Google App Engine, IBM Bluecloud, Amazon Web Service(AWS), and Microsoft Azure,

A Typical Cluster Architecture

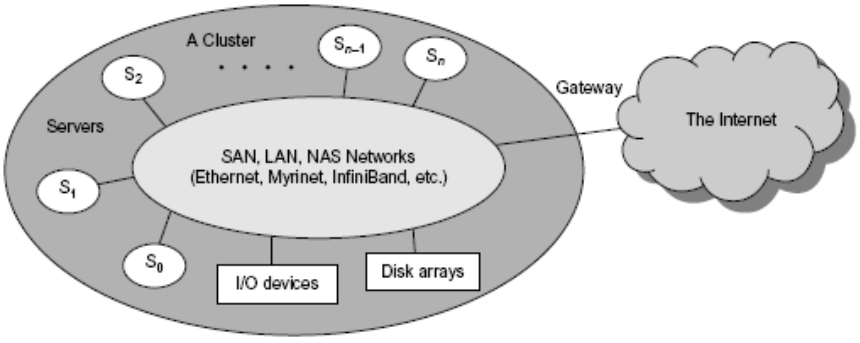


FIGURE 1.15

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

- SAN: Storage Area Network
- LAN: Local Area Network

A Computational Grid

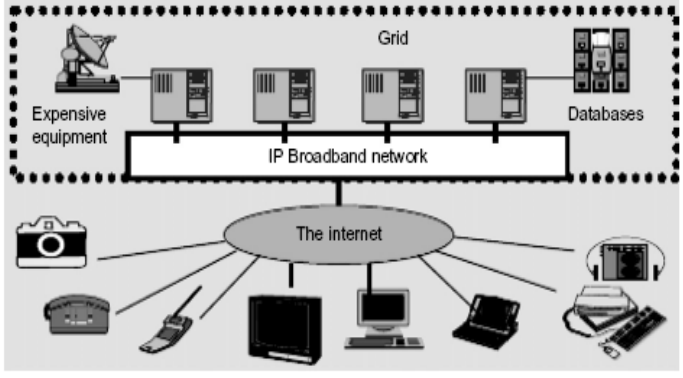


FIGURE 1.16

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

A Typical Computational Grid

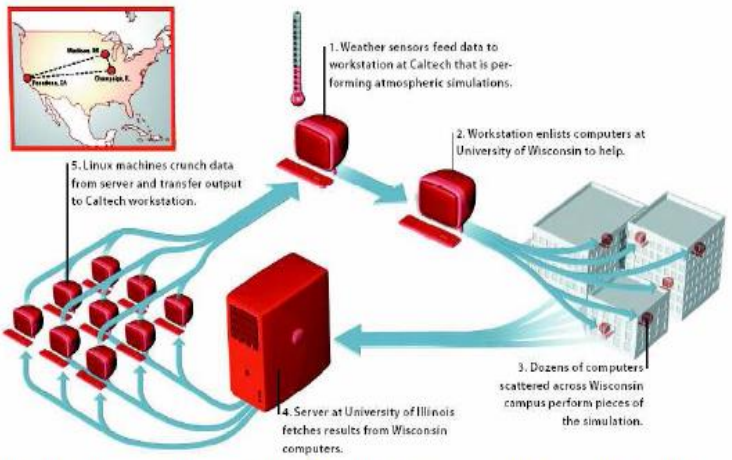


Figure 1.17 An example computational Grid built over specialized computers at three resource sites at Wisconsin, Caltech, and Illinois. (Courtesy of Michel Waldrop, "Grid Computing", IEEE *Computer Magazine*, 2000. [42])

P2P Structure

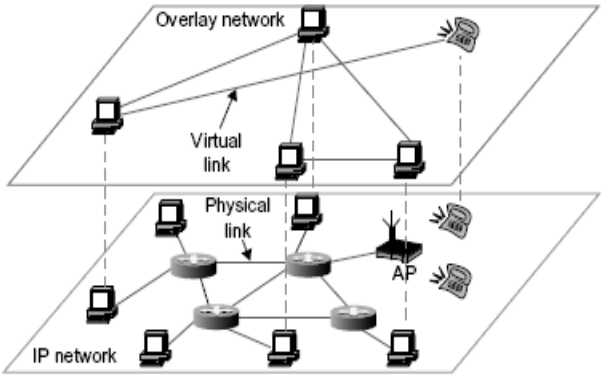


FIGURE 1.17

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

(Courtesy of Zhenyu Li, Institute of Computing Technology, Chinese Academy of Sciences, 2

P2P (Peer-to-Peer) Network 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.

Figure 1.18 Basic Concept of Internet Clouds

- Cloud Computing over the Internet
- Virtualized resource from data centers to form an Internet cloud, provisioning with hardware, software, storage, networks, and services for paid users to run their applications.

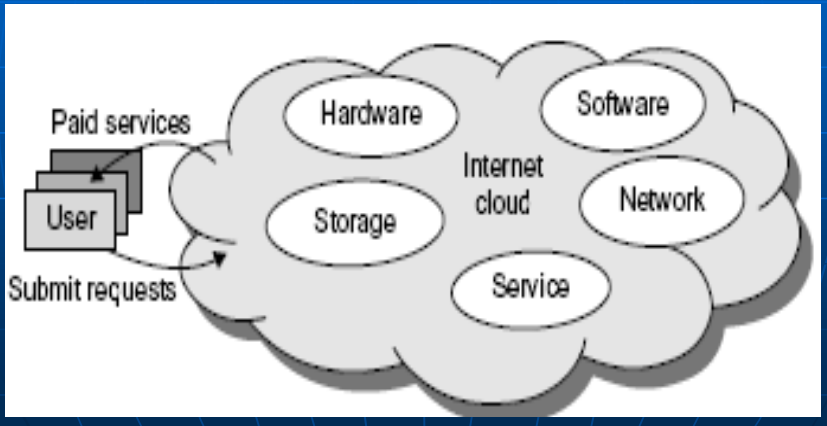


Figure 1.19 Three Cloud Service

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

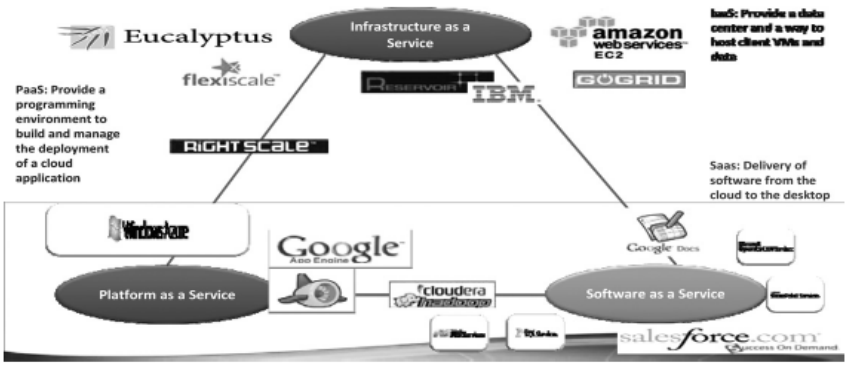


FIGURE 1.19

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

(*Courtesy of Dennis Gannon, keynote address at Cloudcom2010 [19]*)

Eight Reasons to Adapt to the Cloud for Upgraded Internet Applications and Web Services, page 36

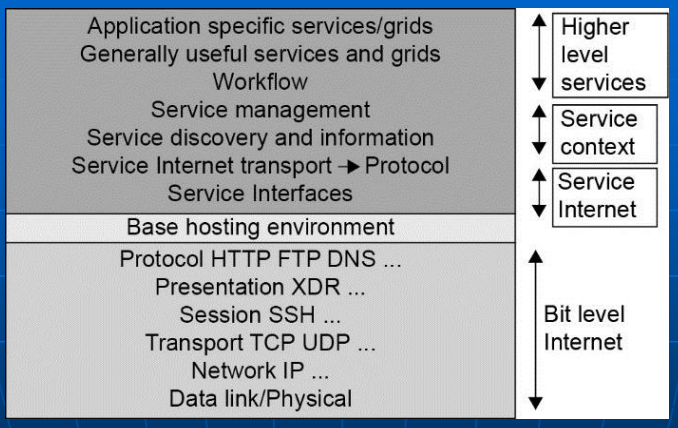
1. Desired locations in areas with **protected space and high energy efficiency**
2. Sharing of peak-load capacity among a large pool of users, improving overall utilization
3. Separation of infrastructure maintenance duties from domain-specific application development
4. Significant reduction in cloud 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

Cloud Computing Challenges: Dealing with too many issues (courtesy of R. Buyya)

1.4 Software Environments for Distributed Systems and Clouds

- Service-Oriented Architecture (SOA)
  - Layered architecture for web services and grids
  - Web services and tools:
    - XML
    - Web services: SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language)
    - REST (Representational State Transfer)
- Trends toward Distributed Operating Systems
- Parallel and Distributed Programming Models
  - Massage-Passing Interface (MPI)
  - MapReduce
  - Hadoop
- Performance, Security, and Energy Efficiency

Figure 1.20 Layered Architecture for Web Services and the Grids



HTTP – Hypertext Transfer Protocol, FTP – File Transfer Protocol
DNS – Domain Name Service
XDR – External Data Representation (FRC-1014)

Figure 1.21 The Evolution of SOA (ss – sensor services, fs – filter or transforming service)

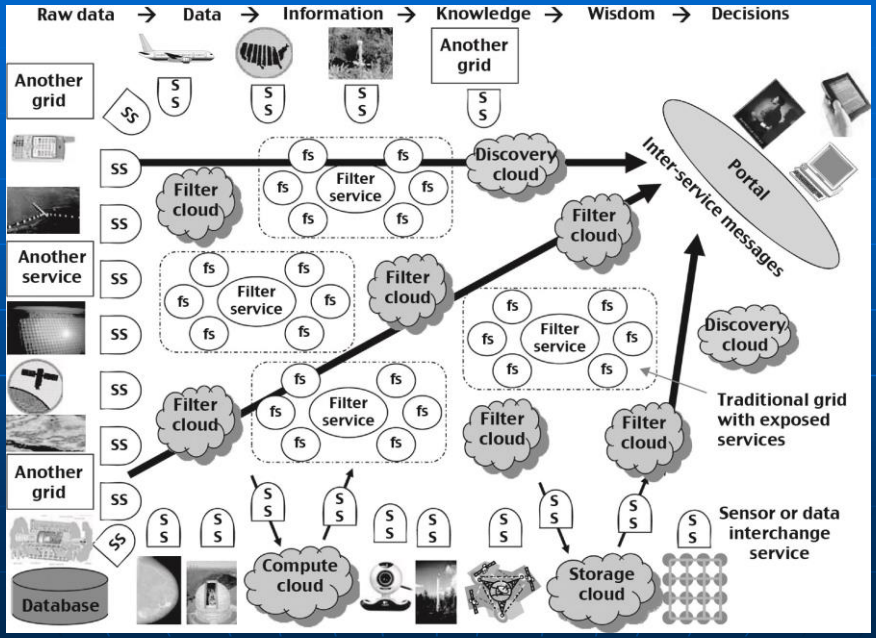


Figure 1.22 A Ideal Model for Cloud Computing: A transparent computing environment that separates the user data, application, OS, and hardware in time and space

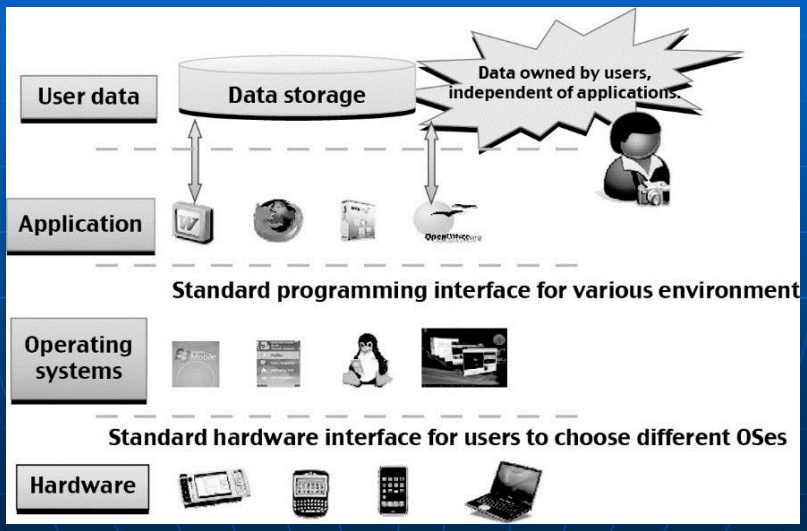


Table 1.6 Feature Comparison of Three Distributed Operating Systems

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

Parallel and Distributed Programming

Table 1.7 Parallel and Distributed Programming Models and Tool Sets

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

Grid Standards and Middleware

Table 1.9 Grid Standards and Toolkits for scientific and Engineering Applications

Grid Standards	Major Grid Service Functionalities	Key Features and Security Infrastructure
OGSA Standard	Open Grid Service Architecture offers common grid service standards for general public use	Support heterogeneous distributed environment, bridging CA, multiple trusted intermediaries, dynamic policies, multiple security mechanisms, etc.
Globus Toolkits	Resource allocation, Globus security infrastructure (GSI), and generic security service API	Sign-in multi-site authentication with PKI, Kerberos, SSL, Proxy, delegation, and GSS API for message integrity and confidentiality
IBM Grid Toolbox	AIX and Linux grids built on top of Globus Toolkit, autonomic computing, Replica services	Using simple CA, granting access, grid service (ReGS), supporting Grid application for Java (GAF4J), GridMap in IntraGrid for security update.

Figure 1.26 Four Operations Layers of Distributed Computing

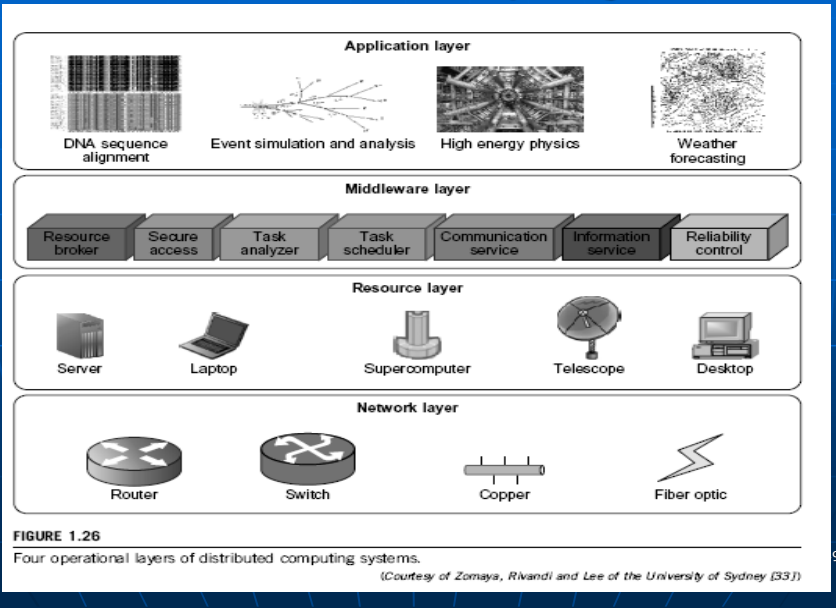


FIGURE 1.26
Four operational layers of distributed computing systems.
(Courtesy of Zomaya, Rivandi and Lee of the University of Sydney [33])

Performance Metrics

- Network bandwidth (Mbps – Million bits per second)
- CPU Speed – MIPS (Million Instructions Per Second)
- System throughput - Performance Metrics
  - MIPS (Million Instruction Per Second) – CPU speed
  - Tflops (Tera floating-point operations per second; Tera = 10^{12})
  - TPSs (Transactions per Second)
  - Job response time
  - Network latency
  - System overhead: OS boot time, compile time, I/O data rate, run-time support system
- QoS – Internet and Web services
- System availability, dependability, security resilience

Scalability Analysis

- Dimensions of Scalability
  - **Size Scalability**
    - Achieve higher performance by increasing machine size
    - Machine size (Ex: 512 processors in 1997 => 65,000 processors)
  - **Software Scalability**
    - Upgrade OS, Compilers Libraries, new software, etc.,
  - **Application Scalability**
    - Problem size
    - Matching “Problem size” with “Machine size”
  - **Technology Scalability**
    - Time (generation scalability), Space (packaging and energy concerns), and Heterogeneity (use hardware components or software packages from different vendors)

System Scalability vs. OS. Multiplicity

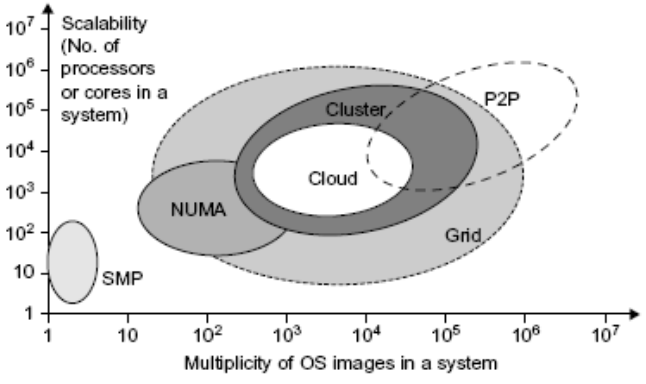


FIGURE 1.23

System scalability versus multiplicity of OS images based on 2010 technology.

Example 1-2. Energy efficiency in distributed power management

- **DVFS (Dynamic Voltage & Frequency Scaling) Method for Energy Efficiency**
- Reducing clock frequency or voltage during slack time (idle time)
- Relationship between Energy and Voltage Frequency in CMOS circuit: v = voltage, Ceff = circuit switching capacity, K = technology dependent factor, vt = threshold voltage; t = the execution time under clock frequency f.

$$\begin{cases} E = C_{eff} f v^2 t \\ f = K \dfrac{(v - v_t)^2}{v} \end{cases}$$

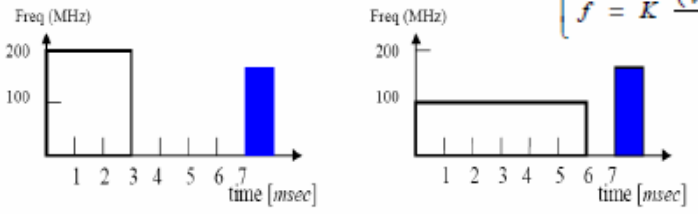


Figure 1.30 DVFS technique (right) original task (left) voltage-frequency scaled task (Courtesy of R.Ge, et al, "Performance Constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters", *Proc. of ACM Supercomputing Conf.*, 2005 [18].)

Amdahl's Laws

- Amdahl's Law – a law governing the speed up of using parallel processors on a problem
- Execute a given program on a uniprocessor computer with a total execution time of T minutes.
- Parallel Computing
  - Ignore all system, I/O time, exception handling or communication overhead. The program is now parallelized or partitioned for parallel execution on a cluster of "many processing nodes"
  - Assume that a fraction "**Alpha α**" of the code must be executed sequentially – Sequential Bottleneck
  - **(1 - α)** of the code can be compiled for execution by "N processors"
  - The total execution time of the program (T) = Sequential Execution Time + Parallel Execution Time

$$\alpha T + (1 - \alpha)T/N$$

Amdahl's Laws (cont.)

- Amdahls' law states that "the speedup factor of using the N-processor system over the use of a single processor is expressed by

$$Speedup = S = \frac{T}{\left[\alpha T + \frac{(1-\alpha)T}{N}\right]} = 1/[\alpha + \frac{1-\alpha}{N}]$$

  - Fixed workload speedup
  - Maximum speedup: if α ≈ 0 , or the code is fully parallelizable
  - As cluster size increases N ≈ Infinity; S ≈ 1/α
    - Upper bound is independent of cluster size N
    - If α = 0.25, or 1 – α = 0.75, the maximum speedup achieved is 4. (Even if hundreds of processors is used)
  - We should make α as small as possible. Increase the cluster size alone may not result in a good speedup

Problem with Fixed Workload

- To execute a fixed workload on **n processors**, parallel processing may lead to a system efficiency defined as:

$$E = \frac{S}{N} = \frac{\left\{\frac{1}{\left[\alpha + \frac{1-\alpha}{N}\right]}\right\}}{N} = 1/[\alpha N + 1 - \alpha]$$

  - α= 0.25 N = 256 nodes, E = 1/[0.25*256 + 0.75] = 1.5%
  - Only a few processors (say, 4) are kept busy, while the majority of the nodes are left idling.
  - Very often the system efficiency is very low, especially when the cluster size is very large.

Gustafson's Law

- Scaled-workload speed up – John Gustafson (1988)
  - Scaling the problem size to match the "**cluster capacity**."
  - Let **W** = workload in a given program.
  - When using an N-processor system, the user scaled the workload to

$$W' = \alpha W + (1 - \alpha) N W$$

  - Only the parallelizable portion of the workload is scaled N times in the second term.

$$S' = \frac{W'}{W} = \frac{[\alpha W + (1 - \alpha) N W]}{W} = \alpha + (1 - \alpha) N$$

  - By fixing the parallel execution time at level W, the following efficiency expression is obtained:

$$E' = S'/N = \frac{\alpha}{N} + (1 - \alpha)$$

Improved efficiency: a 256 node cluster, α = 0.25,

E' = 0.25/256 + 0.75 = 0.751

Fault Tolerance and System Availability

- System Availability

  = MTTF/(MTTF + MTTR)
- A system is highly available if it has
  - a long "Mean Time To Failure (MTTF)" and
  - a short "Mean Time to Repair (MTTR)"

System Availability vs. Configuration Size

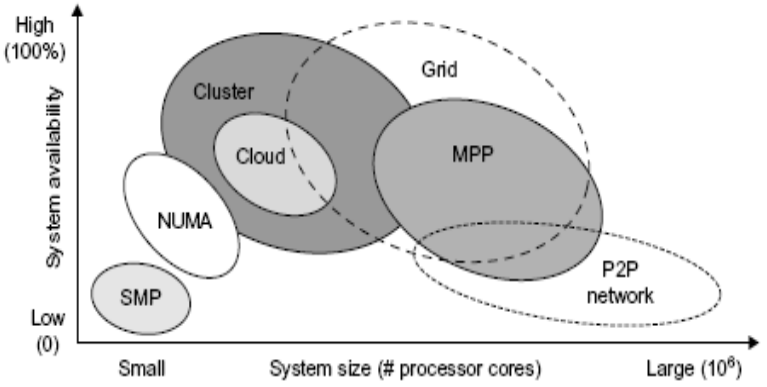


FIGURE 1.24

Estimated system availability by system size of common configurations in 2010.

System Attacks & Network Threats

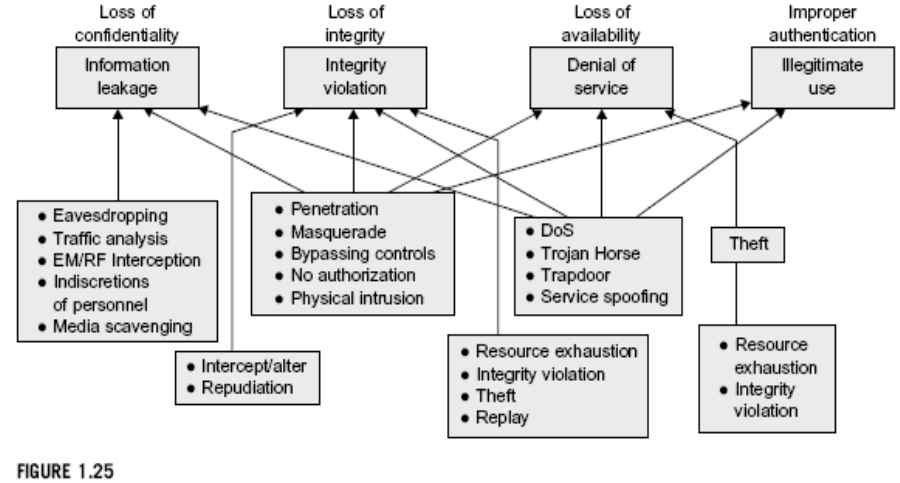


FIGURE 1.25

Various system attacks and network threats to the cyberspace.

Figure 1.22 Transparent Cloud Computing Environment

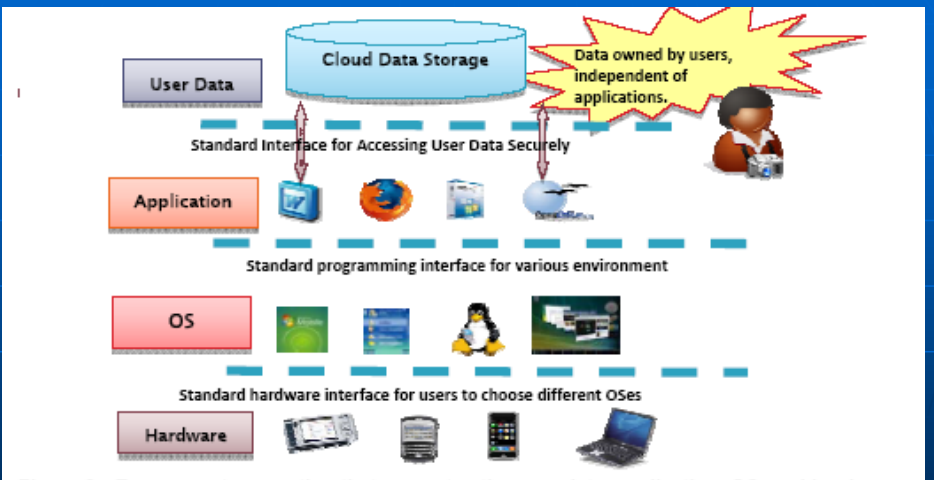


Figure 3 Transparent computing that separates the user data, application, OS, and hardware in time and space – an ideal model for future Cloud platform construction

Summary & Conclusion