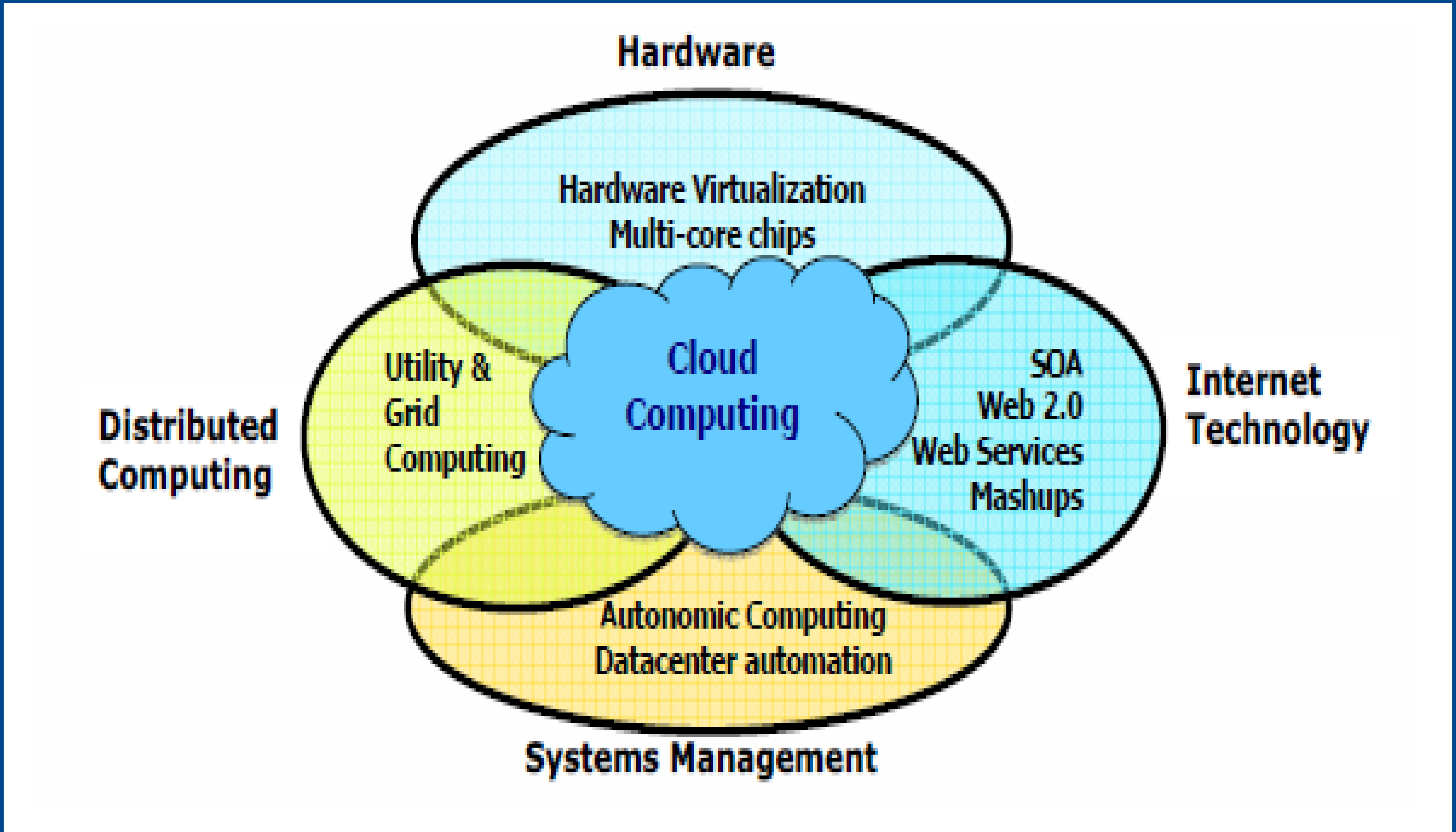
Distributed and Cloud Computing

K. Hwang, G. Fox and J. Dongarra

Lecture 1: Enabling Technologies and Distributed System Models

Data Deluge Enabling New Challenges



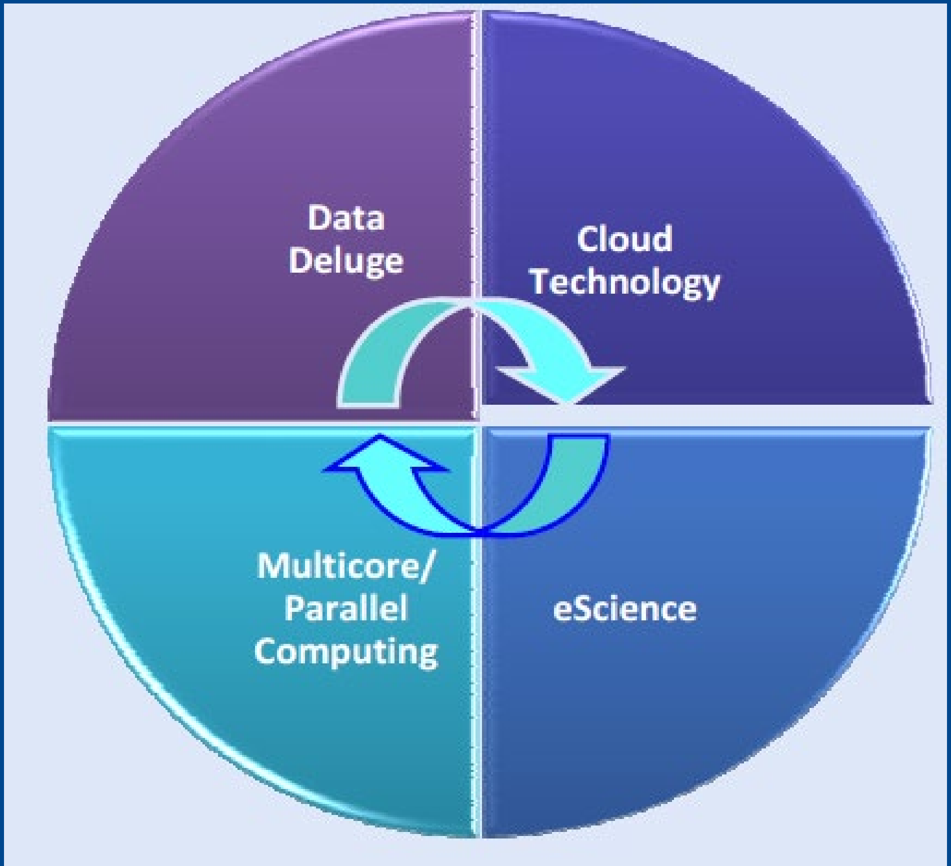
(Courtesy of Judy Qiu, Indiana University, 2011)

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

- HPC moving from centralized supercomputers to geographically distributed desktops, desksides, 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

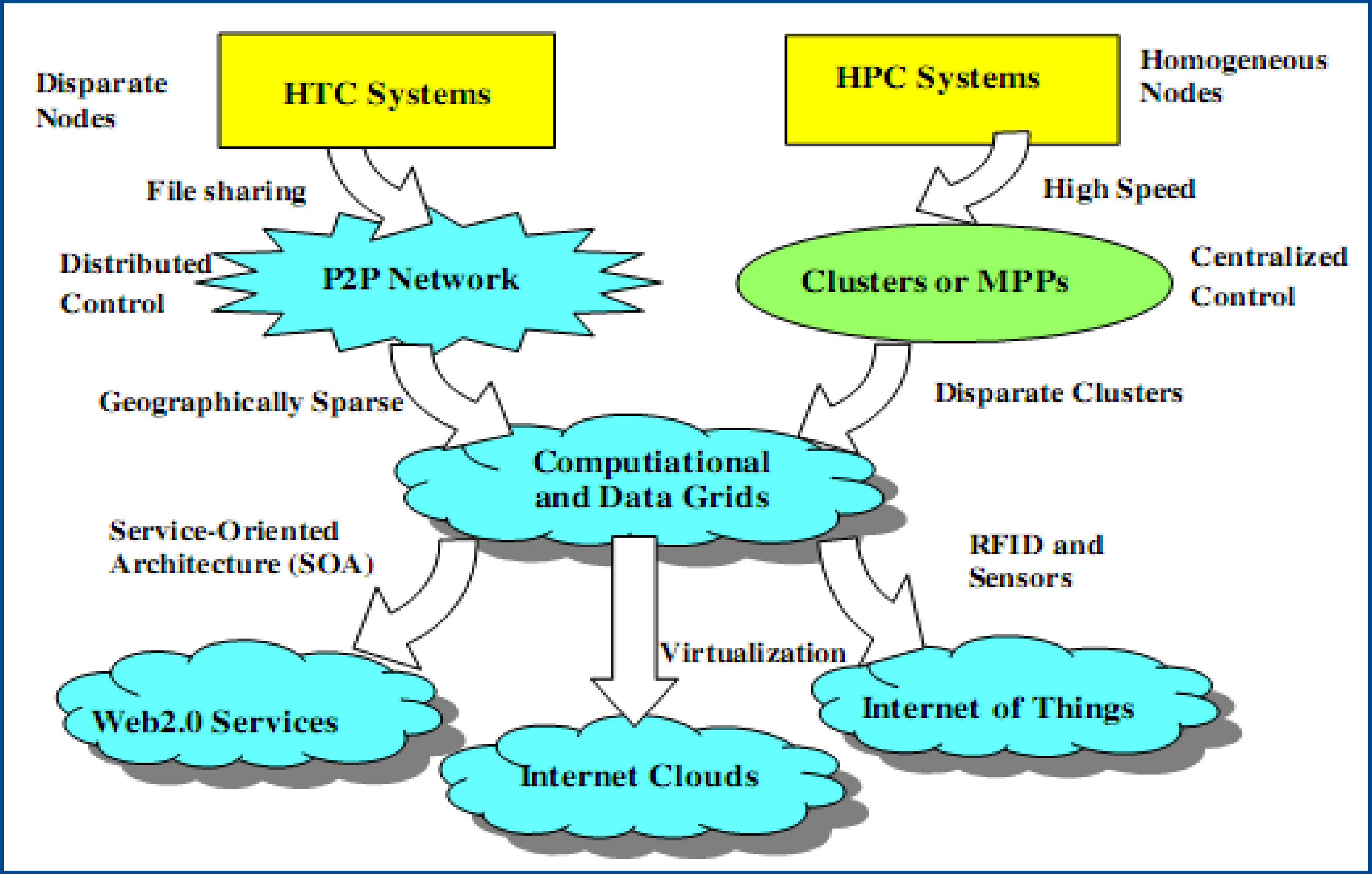
Interactions among 4 technical challenges:

Data Deluge, Cloud Technology, eScience, and Multicore/Pareallel Computing



(Courtesy of Judy Qiu, Indiana University, 2011)

Evolutionary Trend
toward Clouds and Internet of Things



HPC: High-Performance Computing

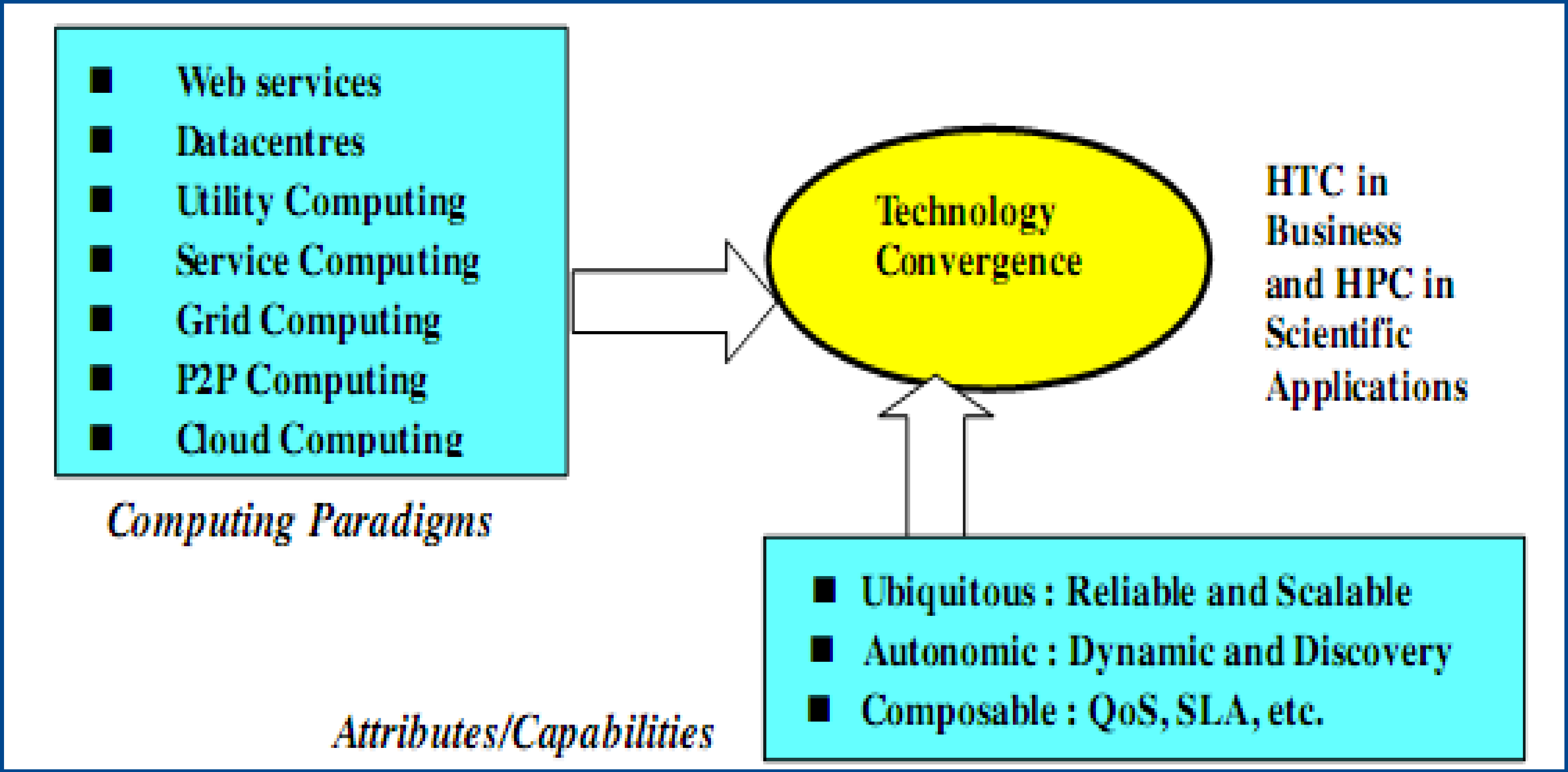
HTC: High-Throughput Computing

P2P: Peer to Peer

MPP: Massively Parallel Processors

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

Technology Convergence toward HPC for Science and HTC for Business

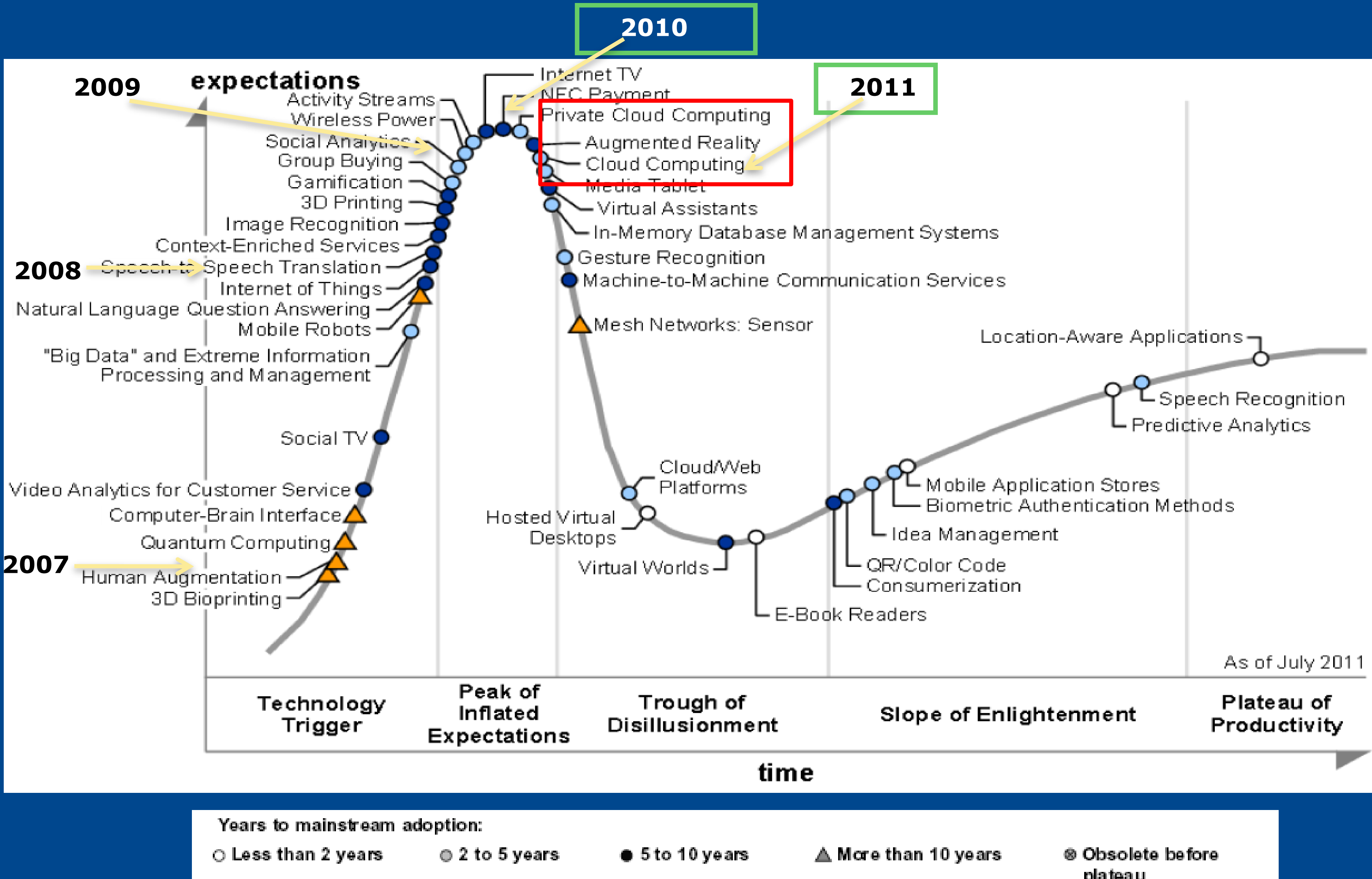


(Courtesy of Raj Buyya, University of Melbourne, 2011)

Major technological challenges
to build distributed system

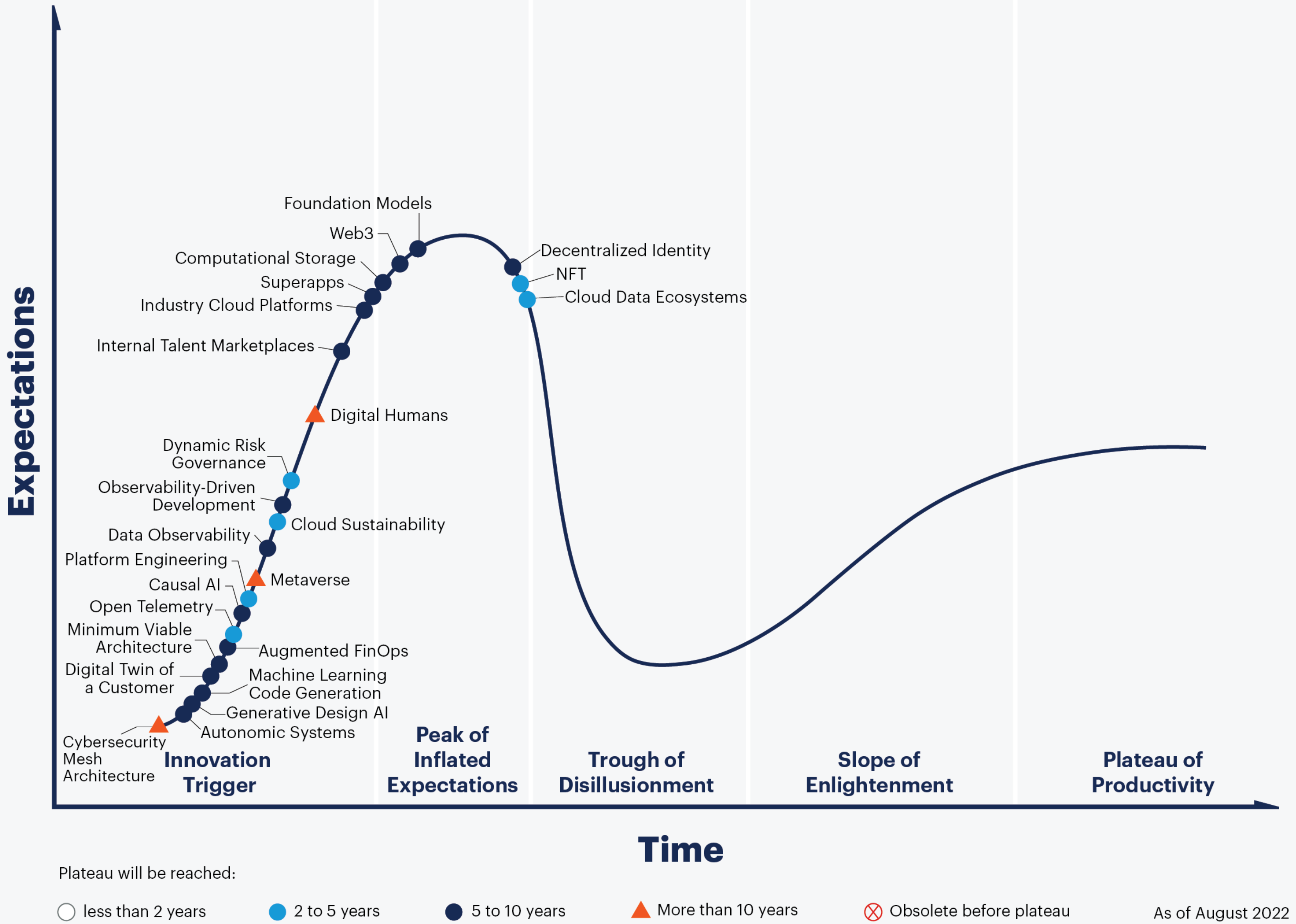
1. New network-efficient processors
2. Scalable memory and storage schemes
3. Distributed OSes
4. Middleware for machine virtualization
5. New programming models
6. Efficient resource management
7. Application program development

2011 Gartner "IT Hype Cycle" for Emerging Technologies



What is Gartner Hype cycle? https://en.wikipedia.org/wiki/Gartner_hype_cycle

Hype Cycle for Emerging Tech, 2022



gartner.com

Internet of Things (IoT)

- Introduced in 1999 at MIT
- What is IoT?
  - The networked interconnection of everyday objectes, tools, devices, or computers.
  - A wireless network of sensors that interconnect all things in our daily life.
- Idea of IoT
  - To tag every object using RFID or a related sensor or electronic technology such as GPS.
  - With IPv6 protocol, 2^{128} IP addresses are available to distinguish all the objects on Earth, including all computers and pervasive devices

Internet of Things (IoT)

- Requirements
  - Track 100 trillion static or moving object simultaneously.
  - Need universal addressability of all of the objects or things.
  - To reduce the complexity of identification, search, and storage, one can set the threshold to filter out fine-grain objects.
- All the objects and devices:
  - Instrumented, interconnected, and interacted with each other intelligently.

Internet of Things (IoT)

- Communication Patterns
  - H2H (human-to-human)
  - H2T (human-to-thing)
  - T2T (thing-to-thing)
- What to achieve: a smart Earth
  - Intelligent cities
  - Clean water
  - Efficient power
  - Convenient transportation
  - Good food supplies
  - Responsible banks
  - Fast telecommunications
  - Green IT
  - Better schools
  - Good health care
  - Abundant resource
  - and so on

Cyber-Physical Systems (CPS)

- CPS
  - Integrates “**cyber**” (heterogeneous, asynchronous) with “**physical**” (concurrent and information-dense) objects
  - Merges the “**3C**” technologies of *computation*, *communication*, and *control* into an intelligent closed feedback system between the physical world and the information world.
- Difference: IoT vs. CPS
  - IoT emphasizes various **networking connections** among physical objects.
  - CPS emphasizes **exploration of virtual reality (VR) applications** in the physical world **to interact with the physical world**.

Multicore CPUs

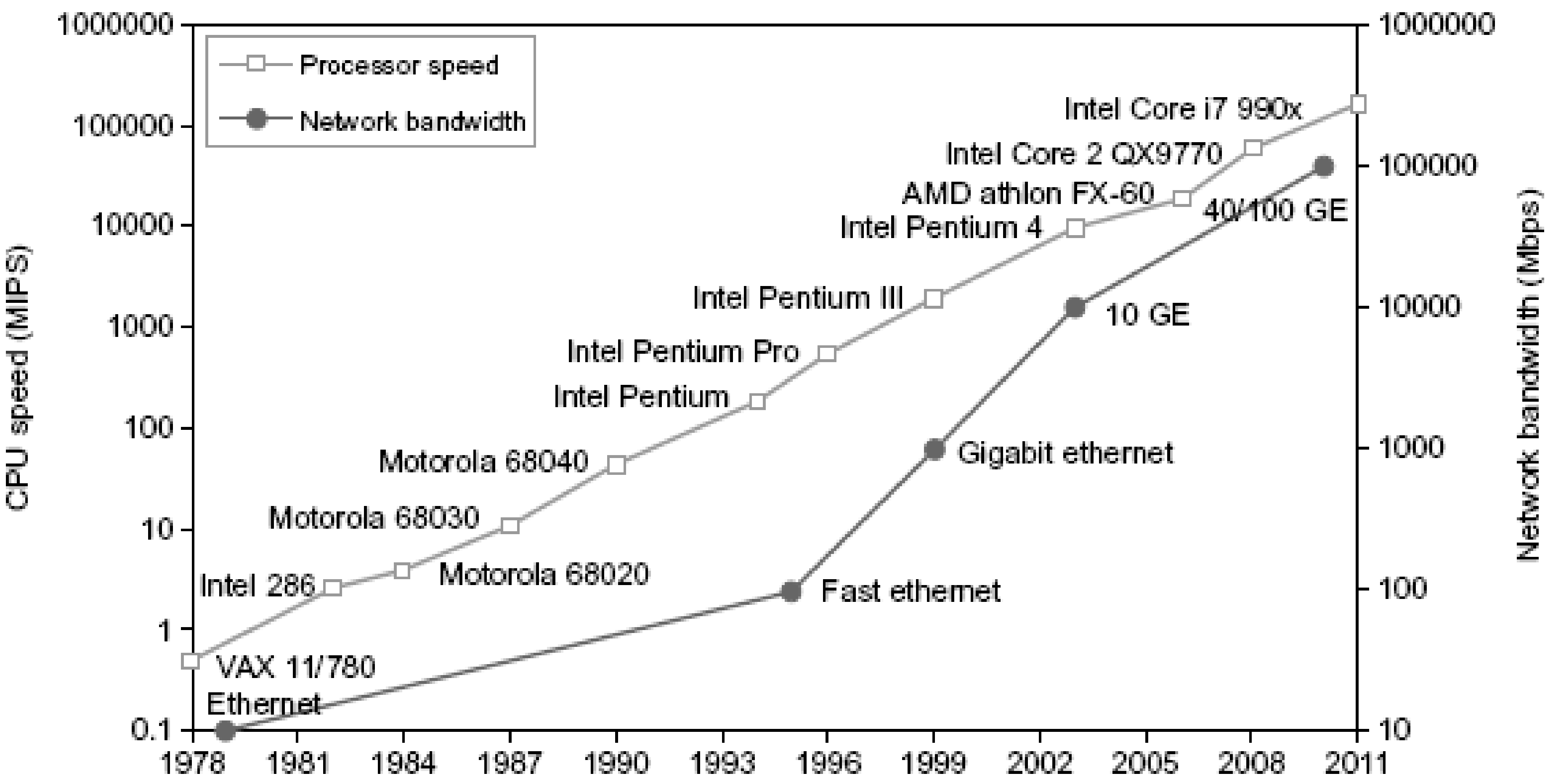


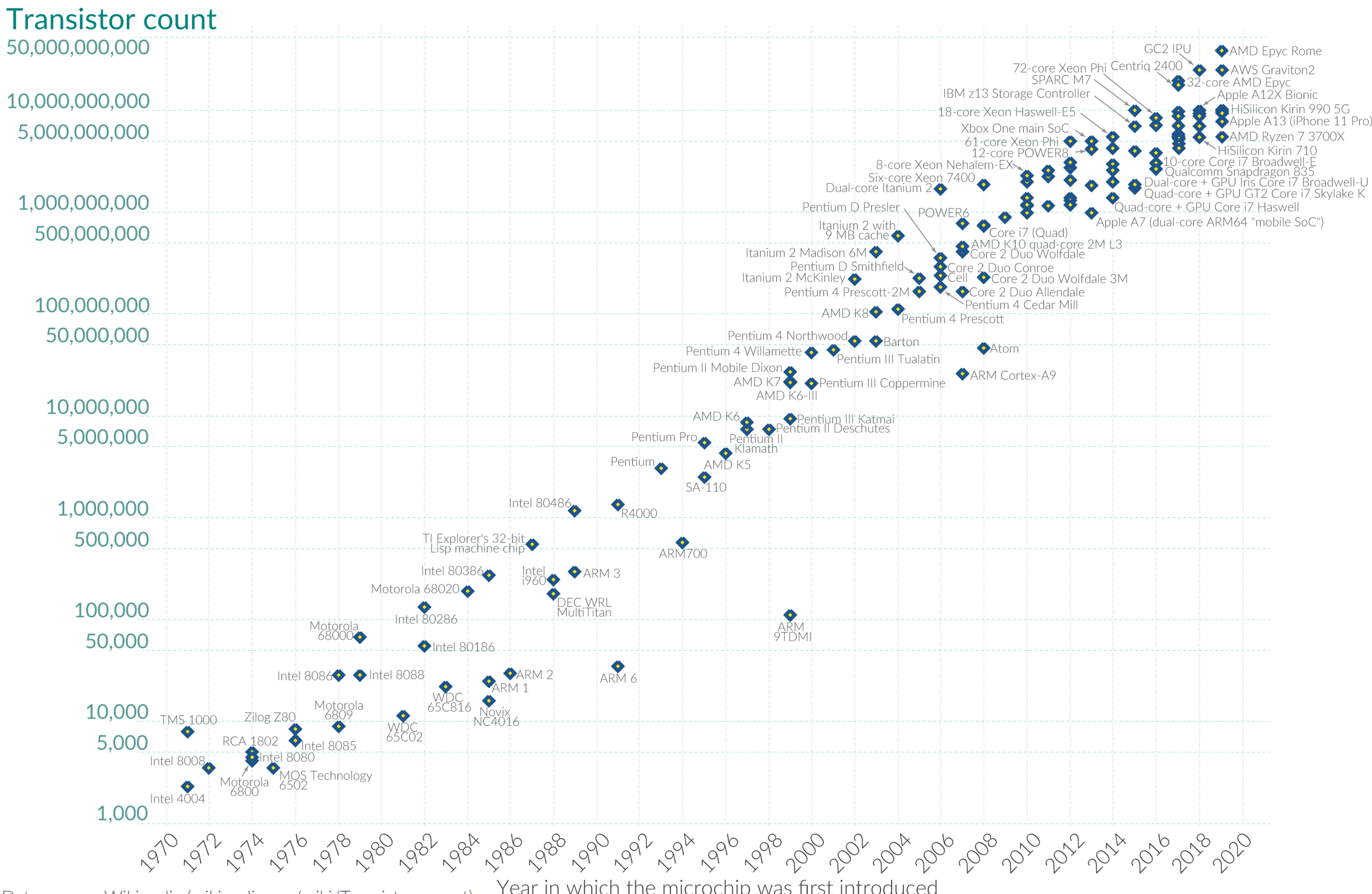
FIGURE 1.4

Improvement in processor and network technologies over 33 years.

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

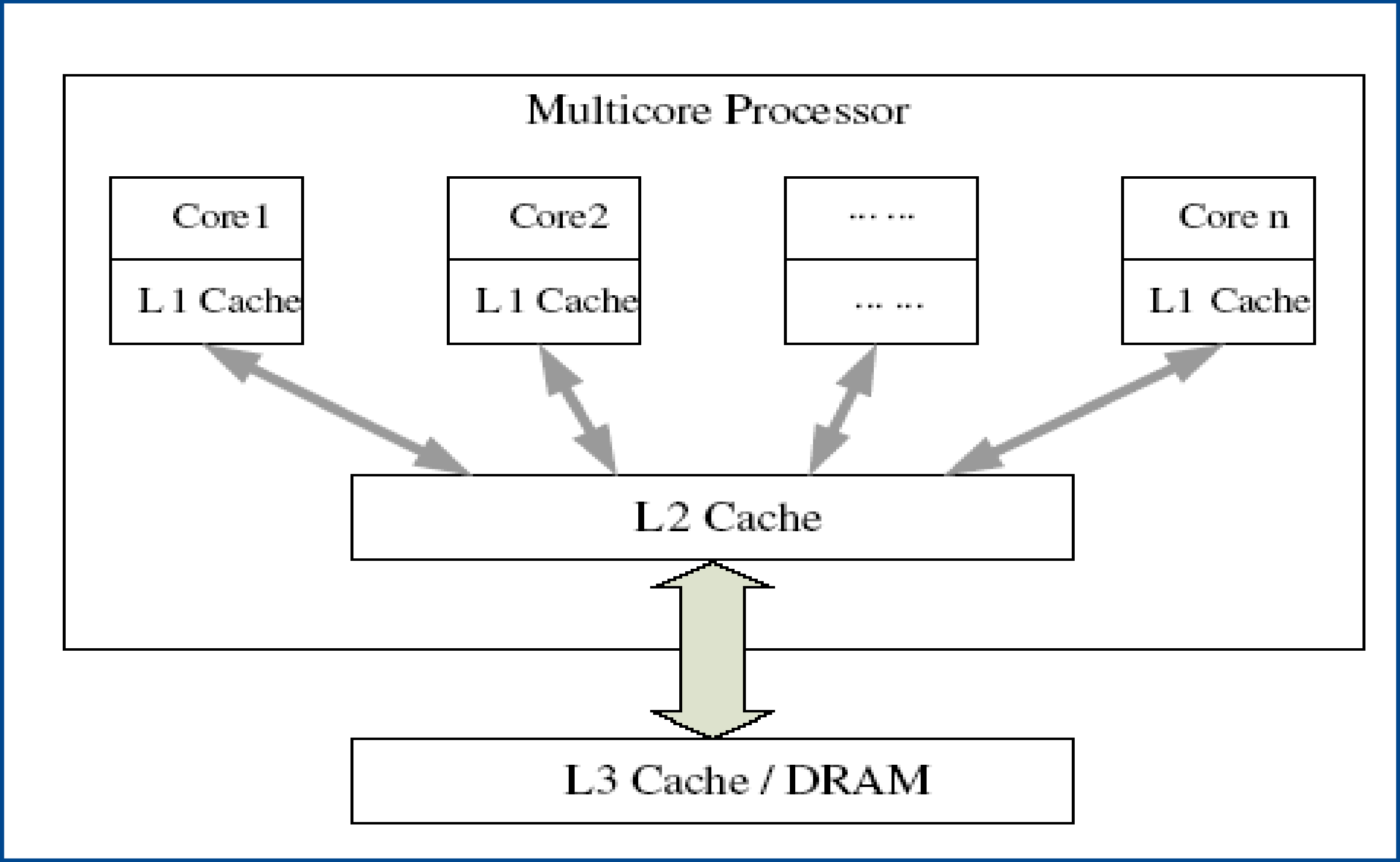
Our World in Data



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Image Source: https://upload.wikimedia.org/wikipedia/commons/0/00/Moore%27s_Law_Transistor_Count_1970-2020.png

Multicore CPU chip using a hierarchy of caches



- L1 cache is private to each core
- On-chip L2 cache is shared
- L3 cache or DRAM is off the chip

Multicore CPUs and Multithreading Technologies

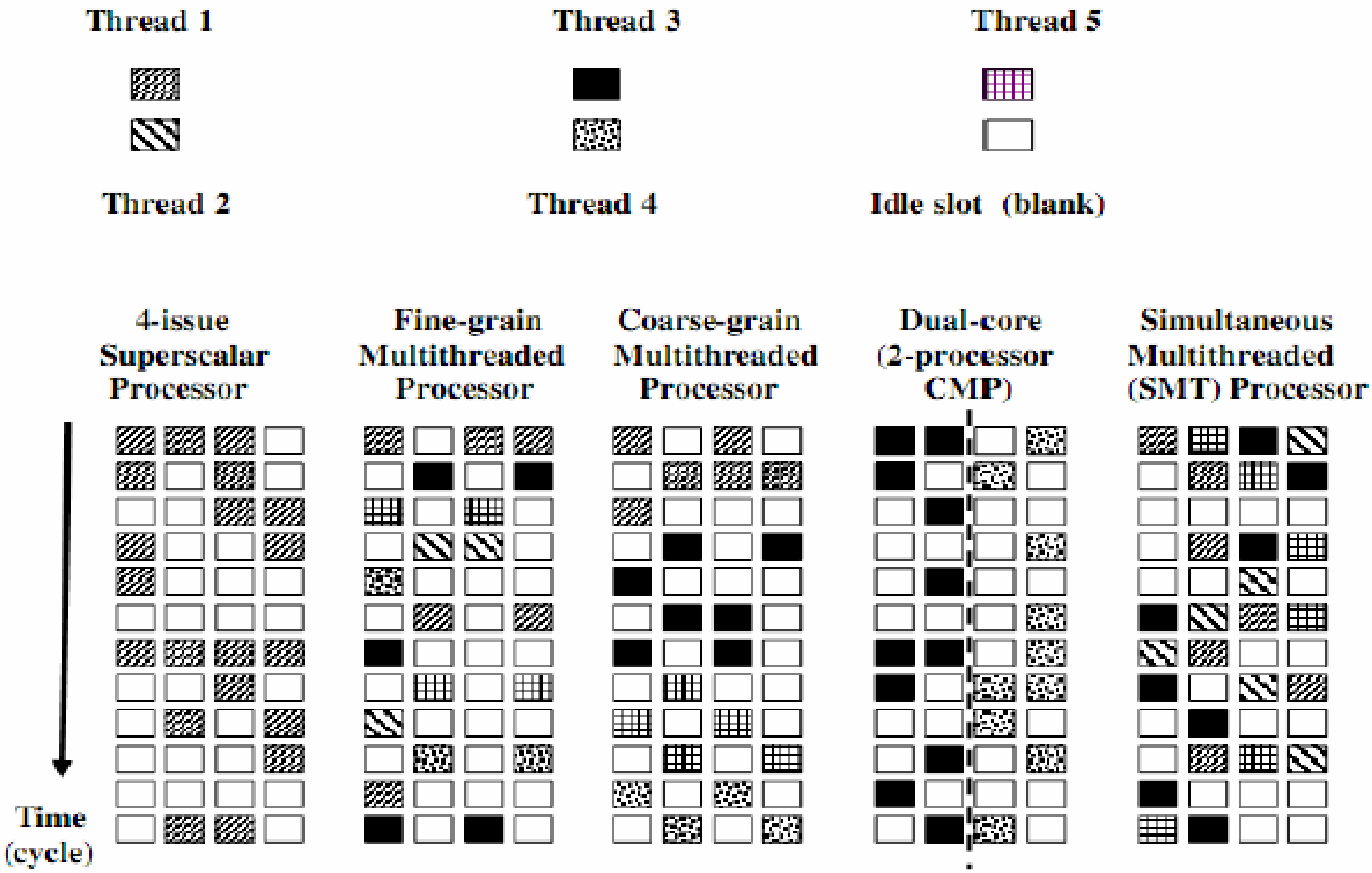
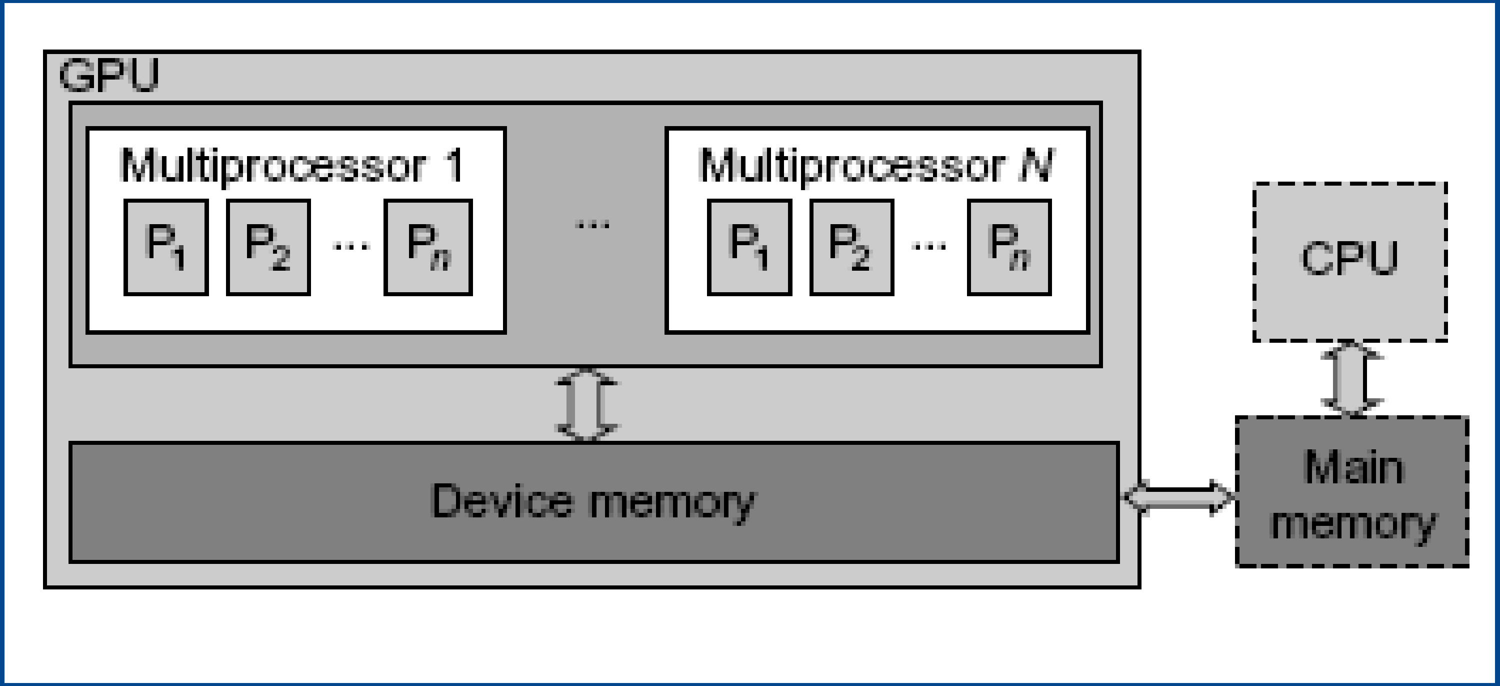


Figure 1. 8 Five micro-architectures that are current in use in modern processors that exploit both ILP and TLP supported by multicore and multithreading technologies

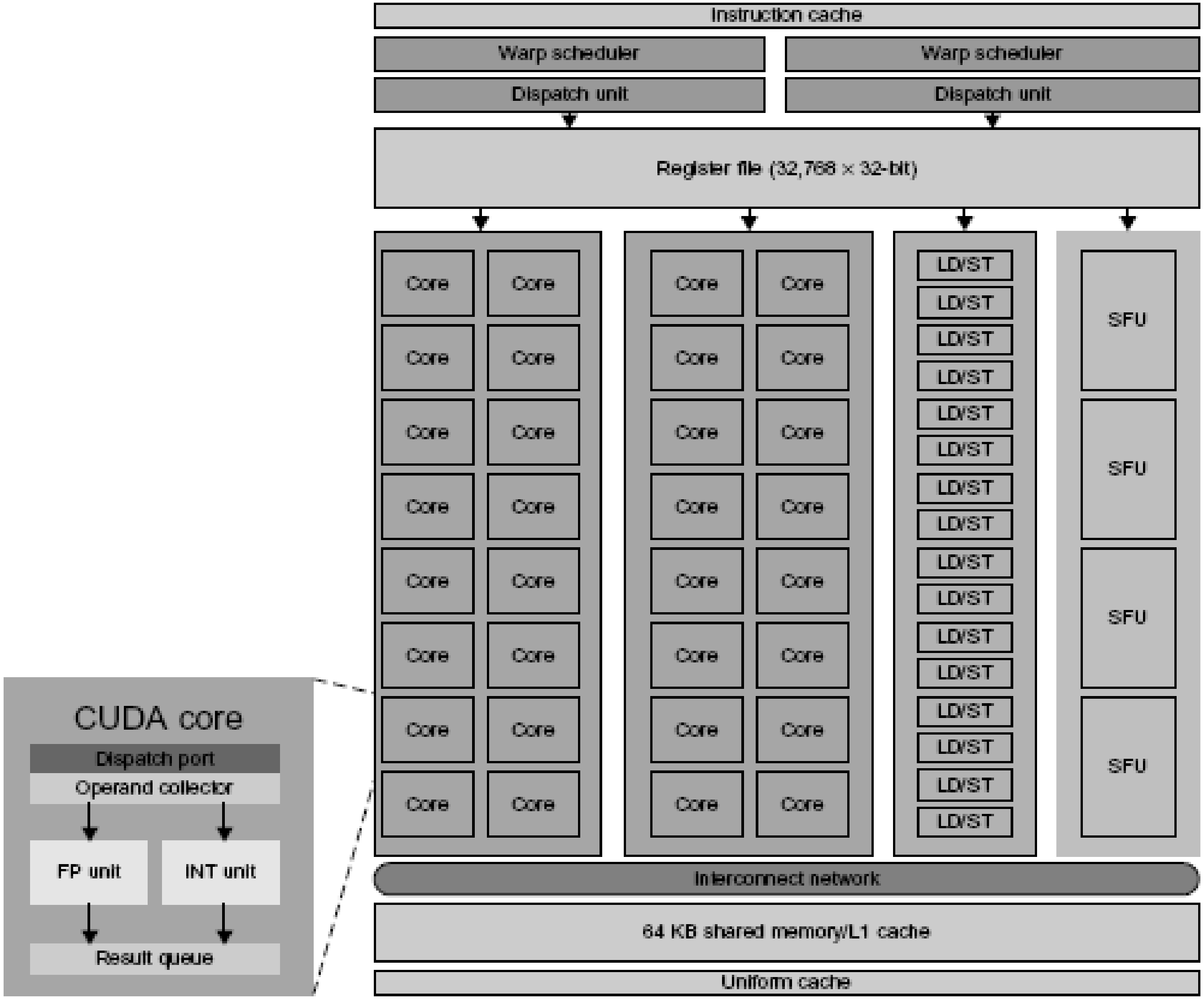
Graphics Processing Unit (GPU)

- GPU
  - graphics coprocessor or accelerator mounted on a computer's graphics card of video card.
- *General-purpose computing on GPU* (**GPGPU**)
  - NVIDIA's CUDA model was for HPC using GPGPUs.
- Modern GPUs
  - To power supercomputers with **massive parallelism** at **multicore** and **multithreading** levels.

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



For massively parallel execution in 100s or 1000s of processing



NVIDIA Femi GPU built with 16 streaming multiprocessors (SMs) of 32 CUDA cores each; only one SM is shown.

GPU: (ex) NVIDIA TITAN V
as of 2022

Graphics Professing Clusters	6
Streaming Multiprocessors	89
CUDA Core (single precision)	5120
Texture Units	320
Base Clock (MHz)	1200 MHz
Boost Clock (MHz)	1455 MHz

Comparison of the CPU and GPU

in performance/power ratio

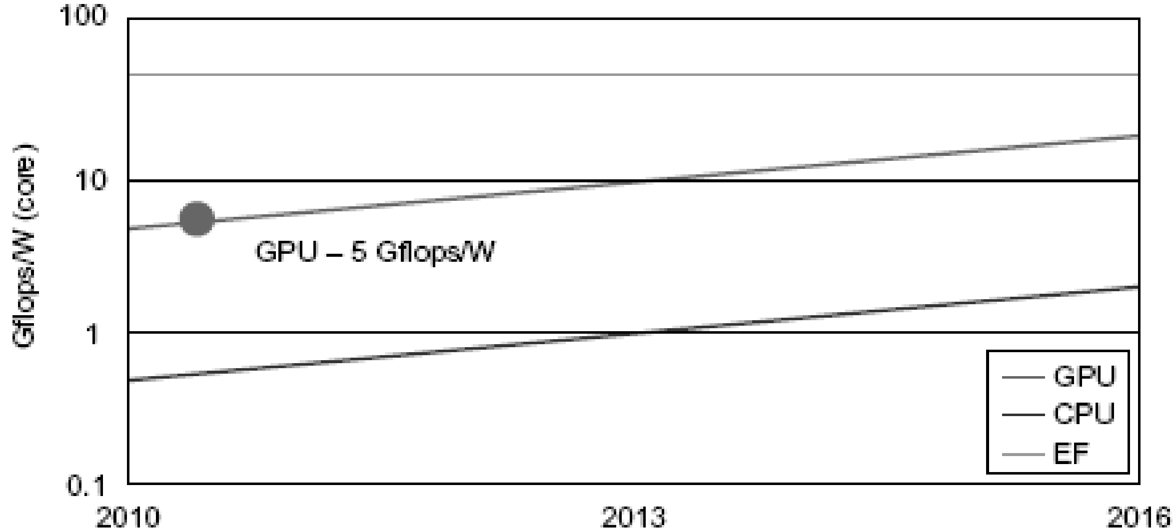
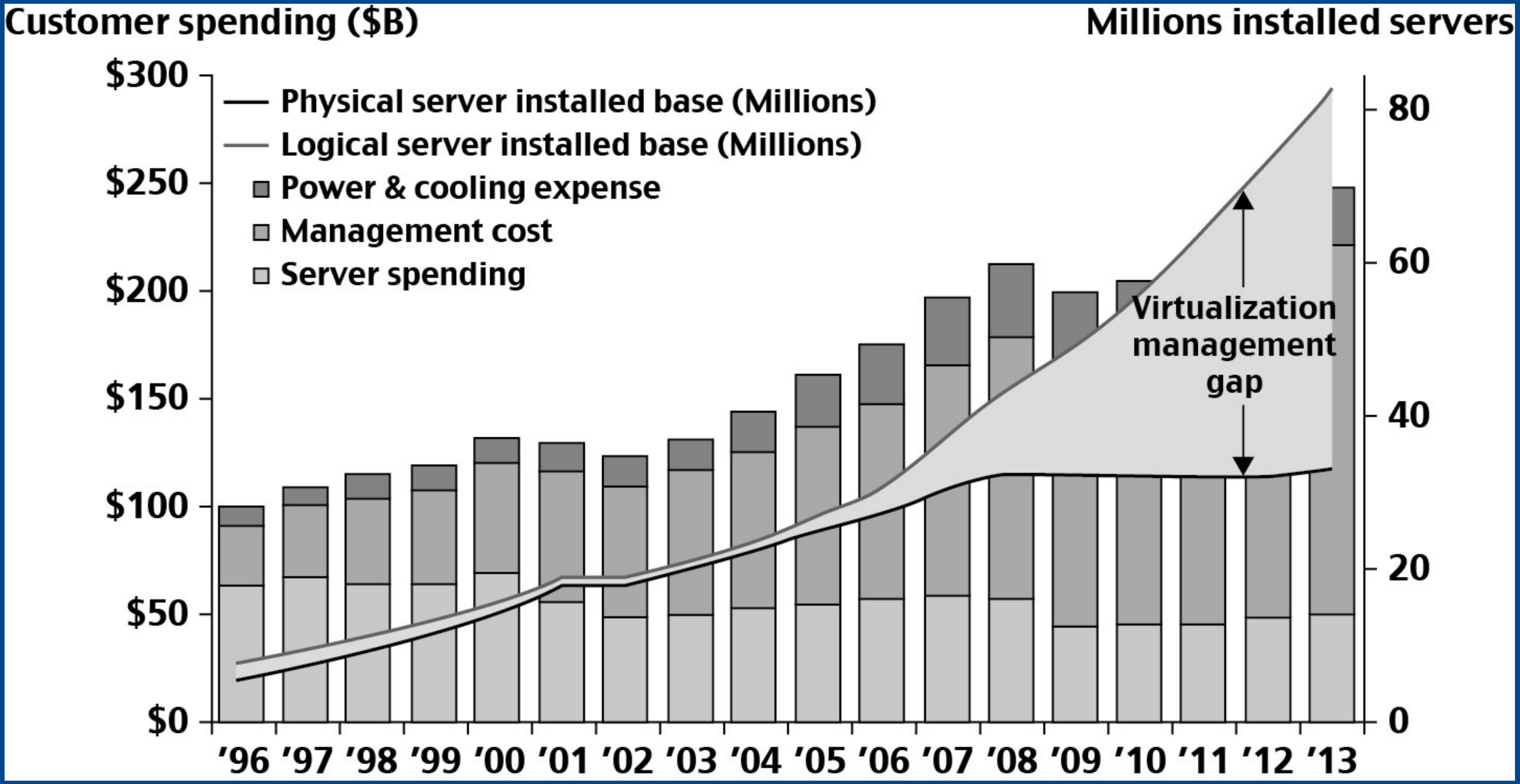


FIGURE 1.9

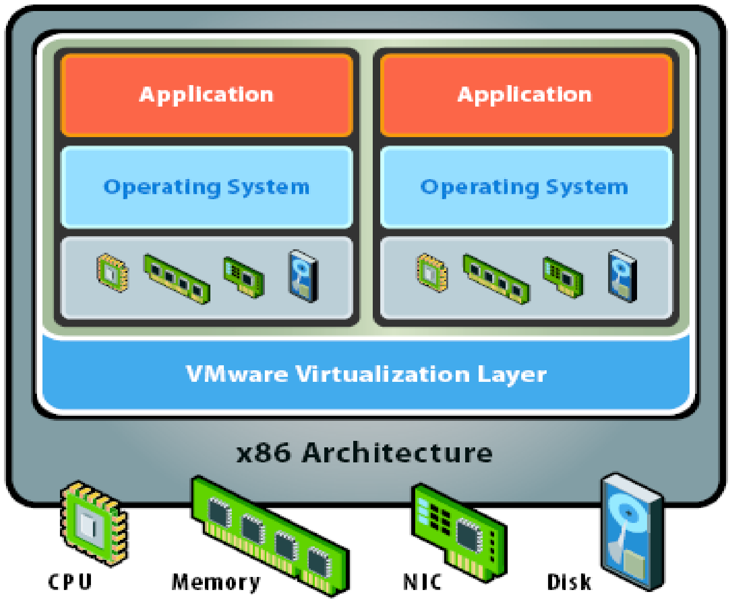
GPU and CPU performance in Gflops/Watt/core, compared with 60 Gflops/Watt/core projected in future Exascale systems.

Bill Dally of Stanford Univ. considers power and massive parallelism are major benefits of GPUs over CPUs for the future.

Datacenter and Server Cost Distribution



Virtual Machine Architecture



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

(Courtesy of VMWare, 2010)

Primitive Operations in Virtual Machines:

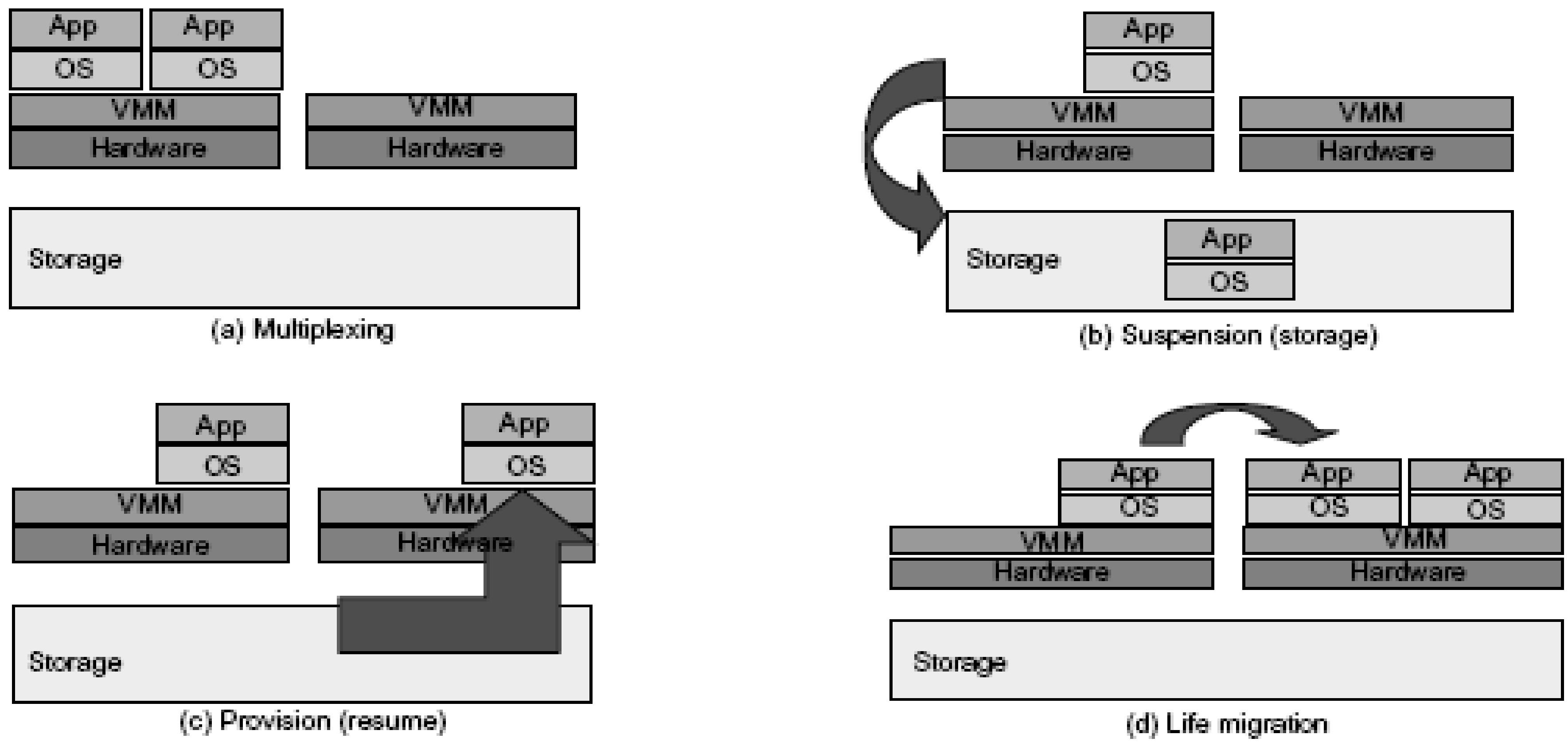


FIGURE 1.13

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

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

Concept of Virtual Clusters

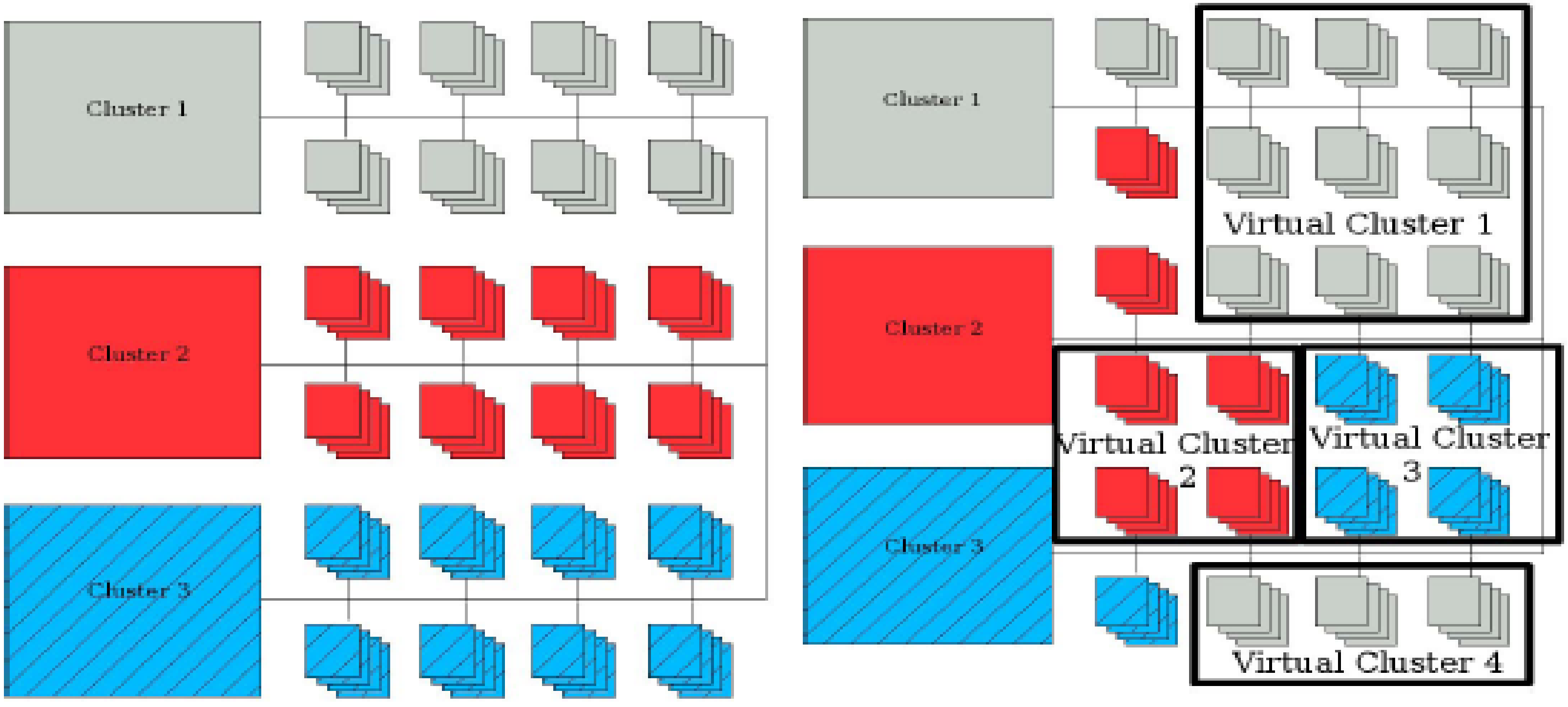


Fig. 1. A Campus Area Grid

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)

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

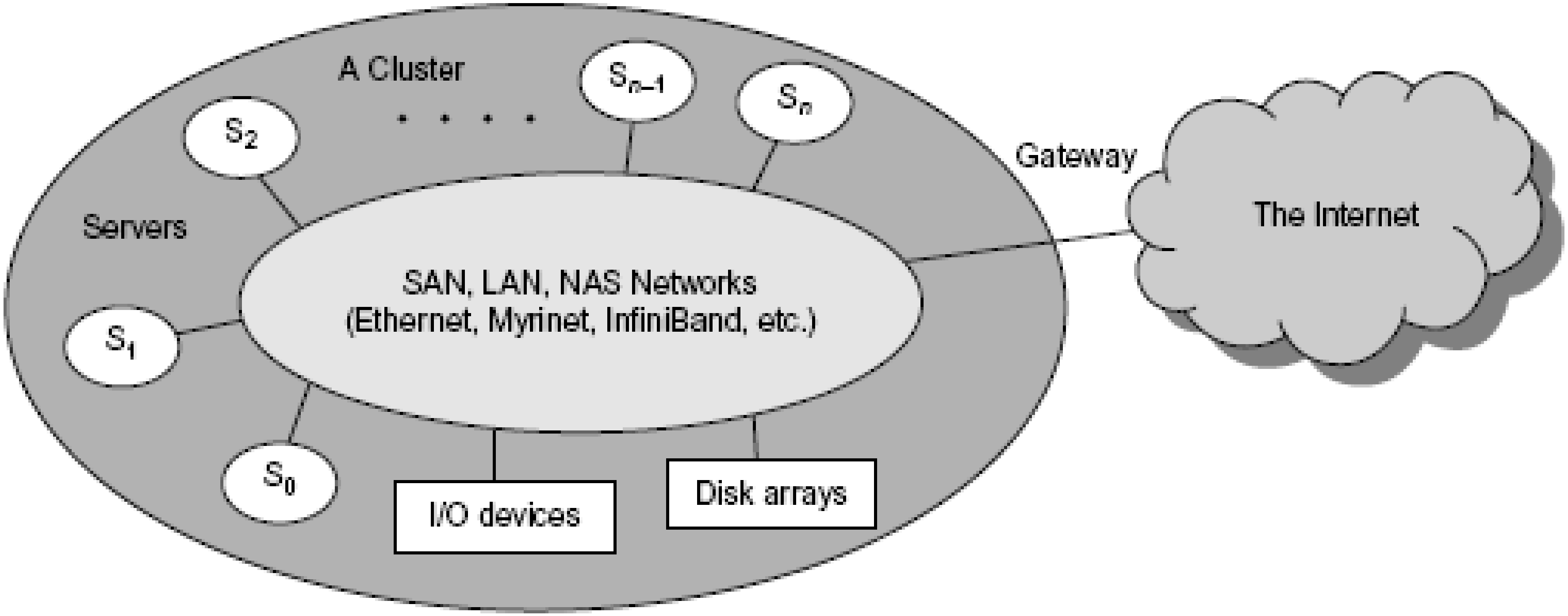


FIGURE 1.15

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.

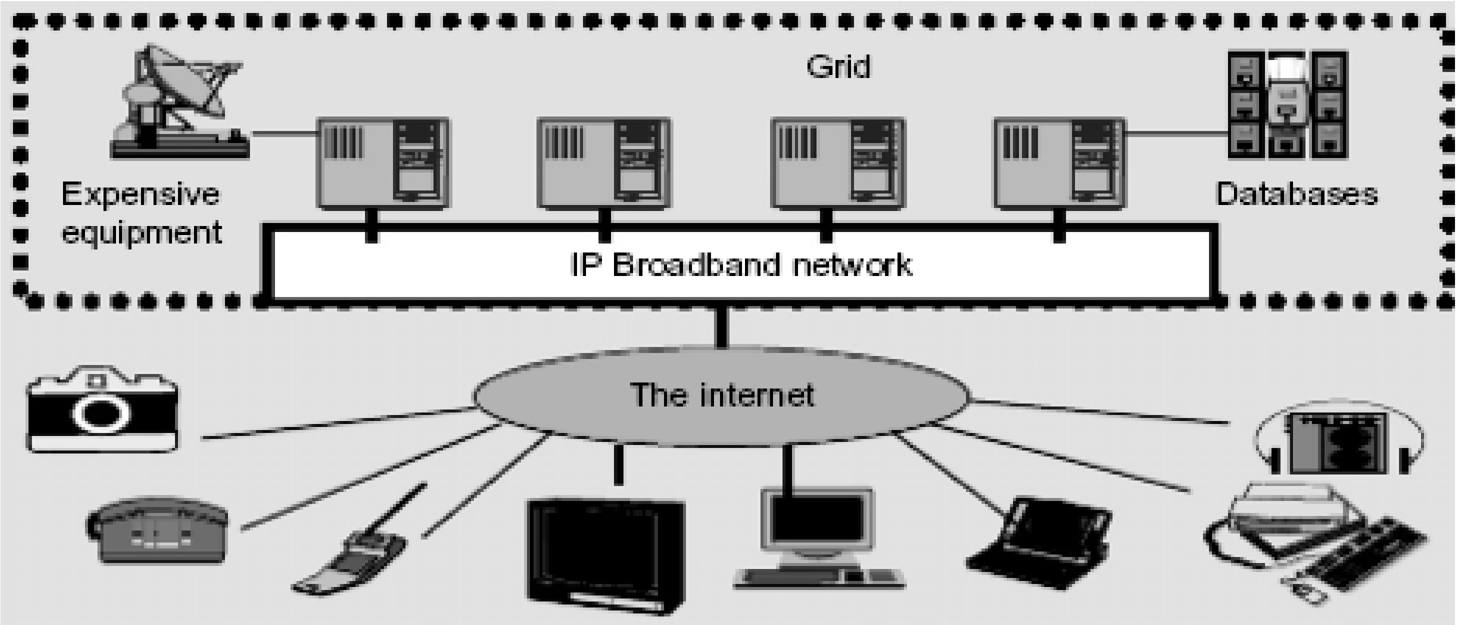


FIGURE 1.16

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

A Typical Computational Grid

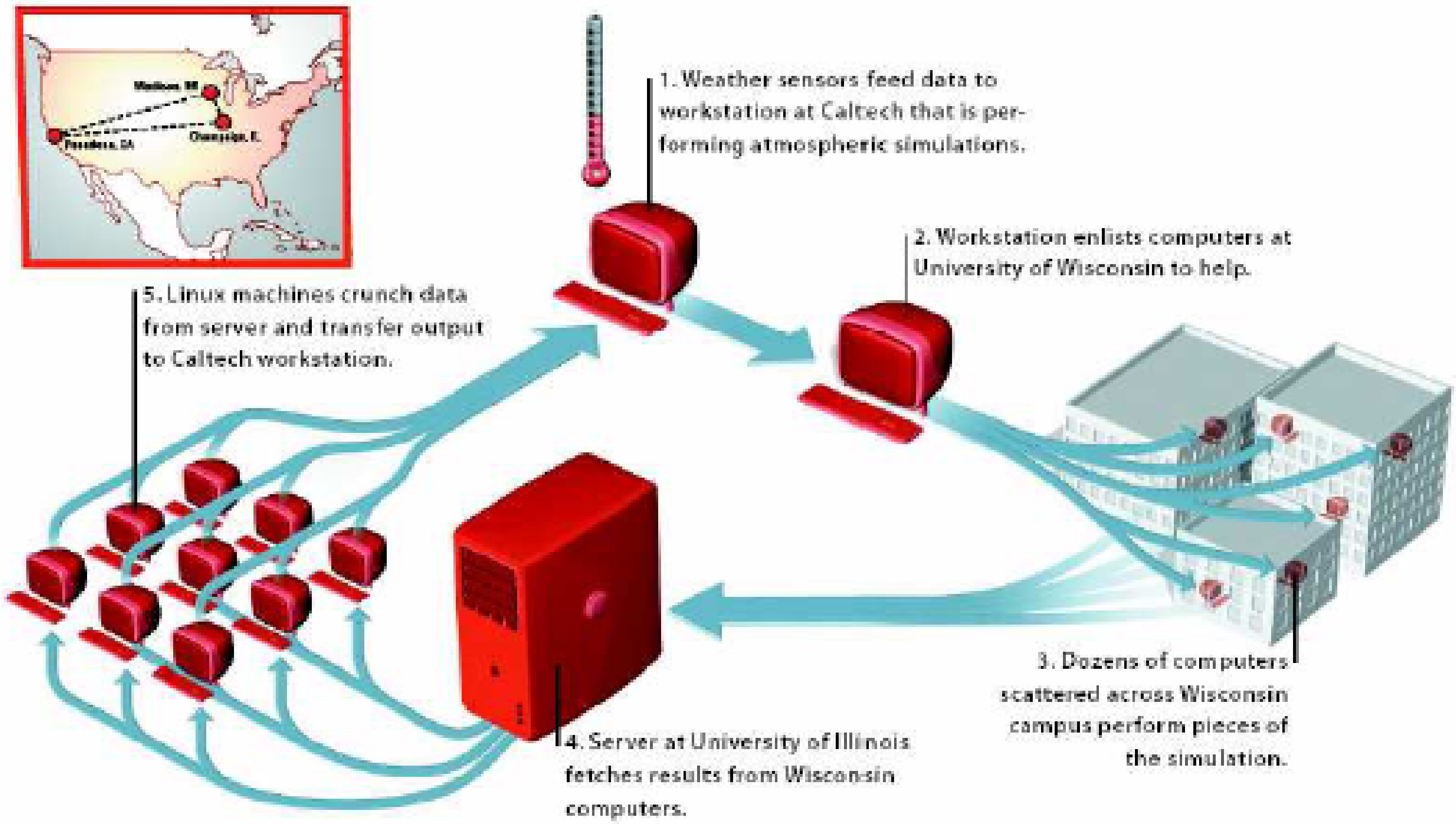


Figure 1.17 An example computational Grid built over specialized computers at three resource sites at Wisconsin, Caltech, and Illinois. (Courtesy of Michel Waldrop, "Grid Computing", IEEE *Computer Magazine*, 2000. [42])

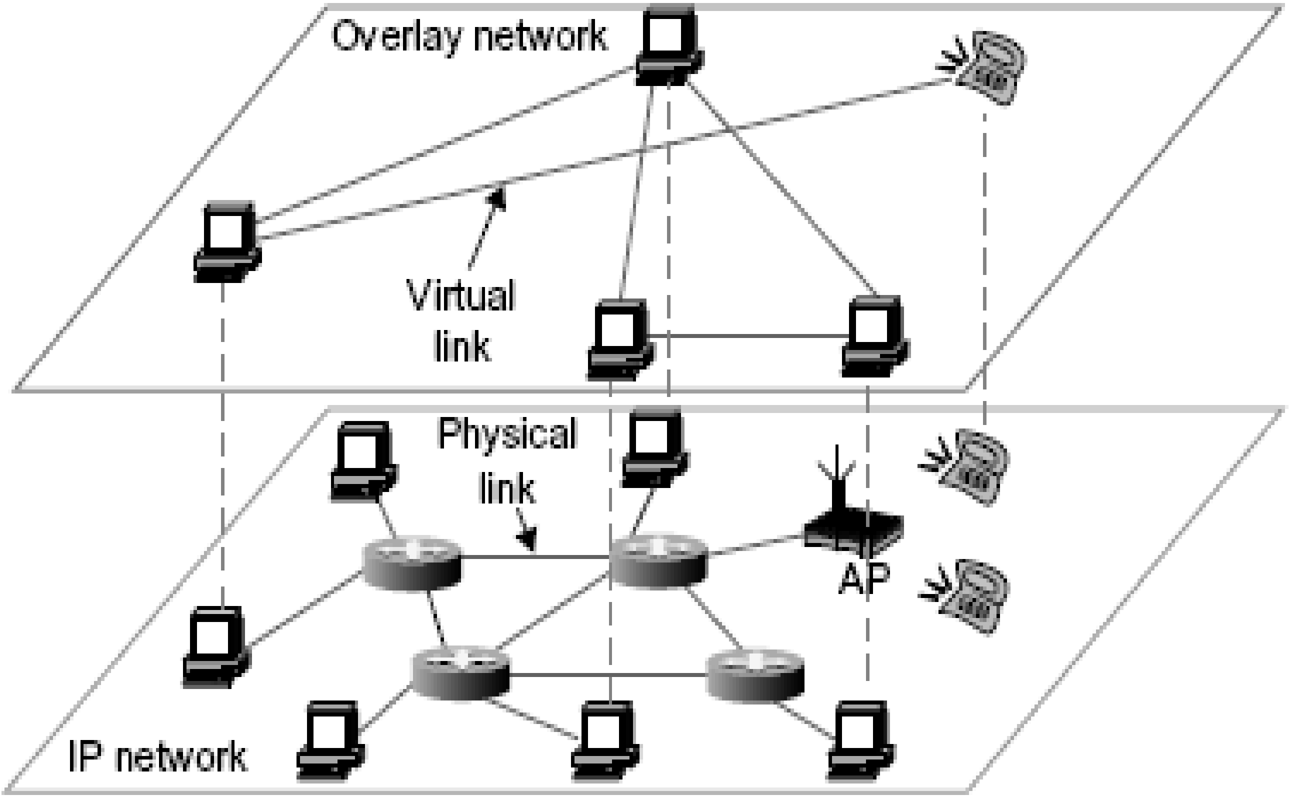


FIGURE 1.17

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

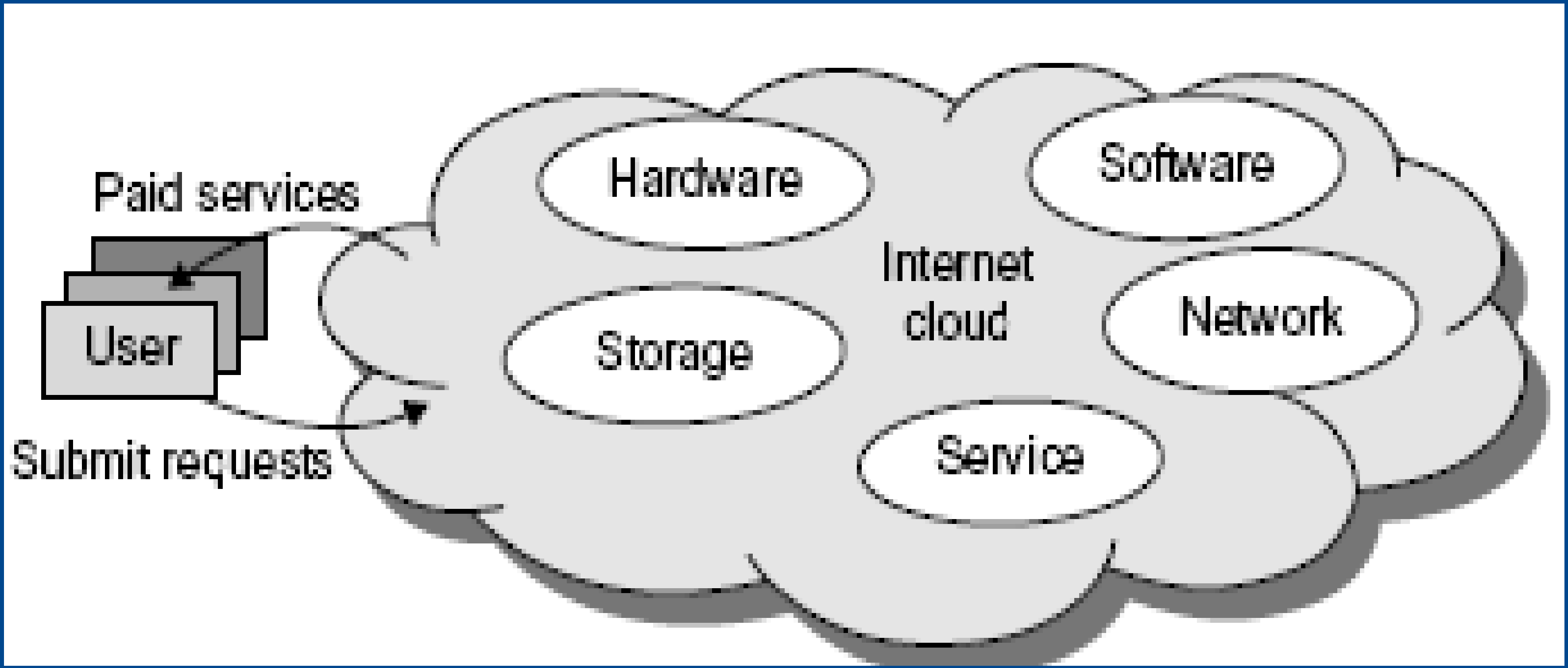
Table 1.5 Major Categories of P2P Network Families [42]

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

Basic Concept of Internet Clouds



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

(Courtesy of Raj Buyya, 2012)

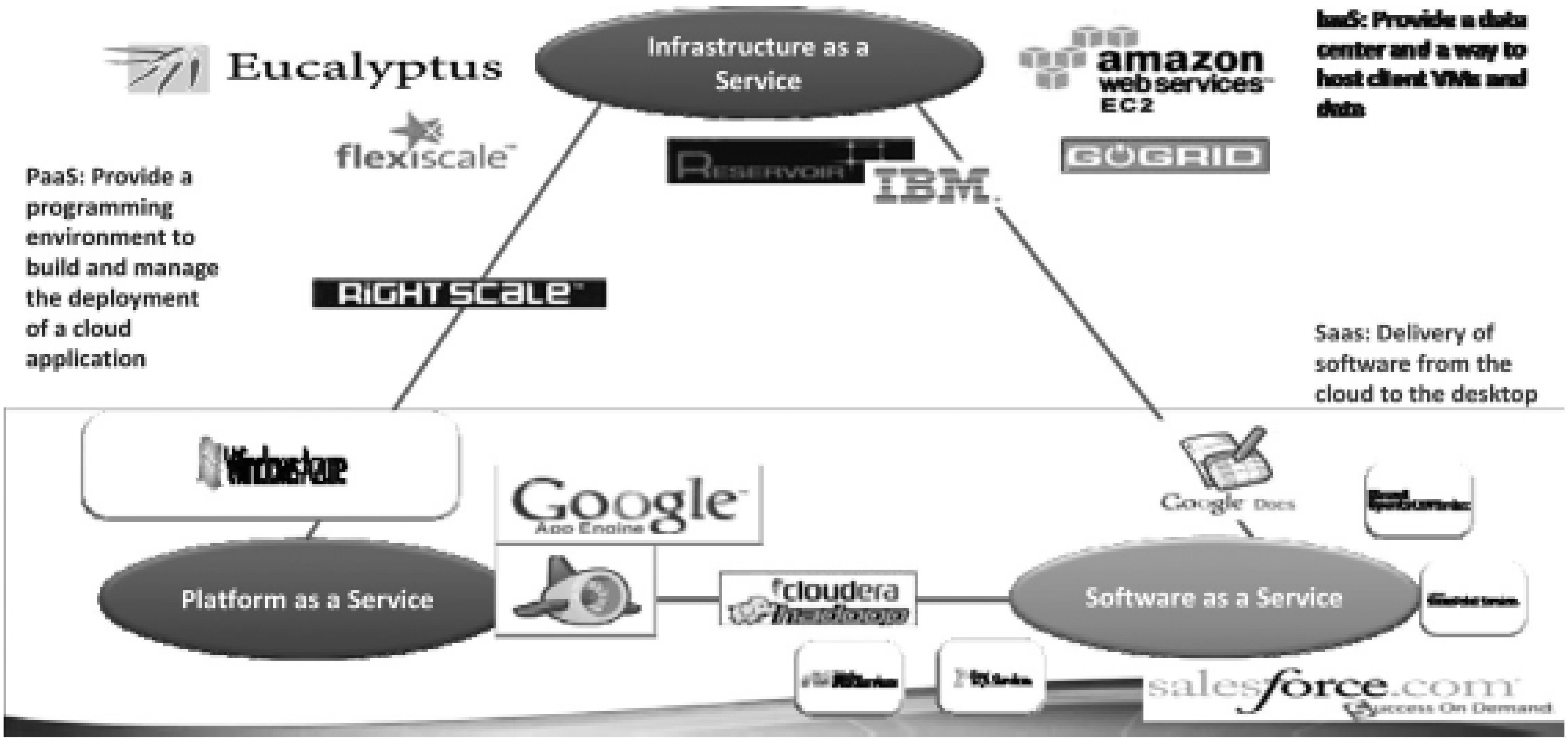


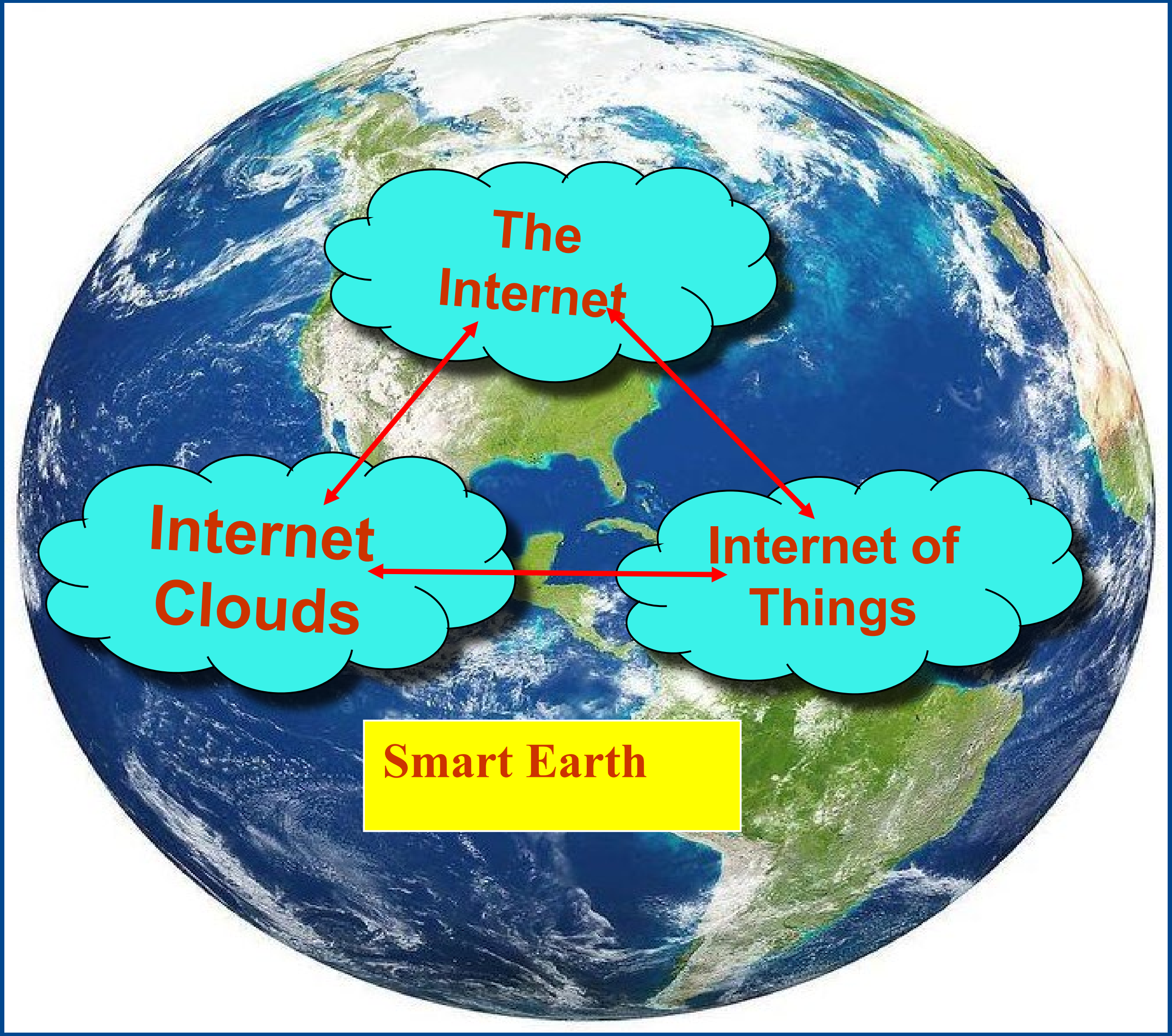
FIGURE 1.19

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

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

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

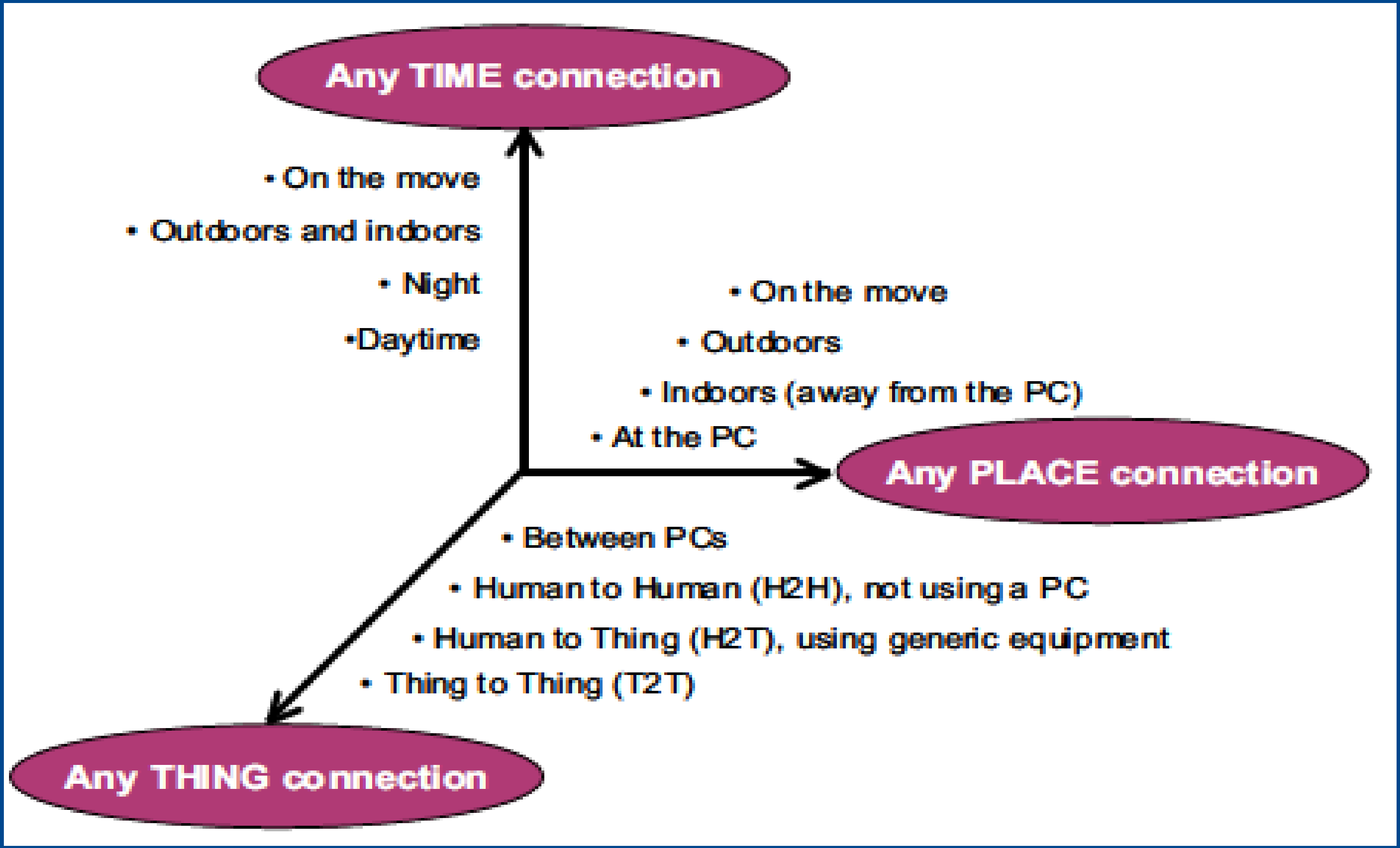
The Internet of Things (IoT)



Smart Earth:

An IBM Dream

Opportunities of IoT in 3 Dimensions



(courtesy of Wikipedia, 2010)

System Scalability vs. OS Multiplicity

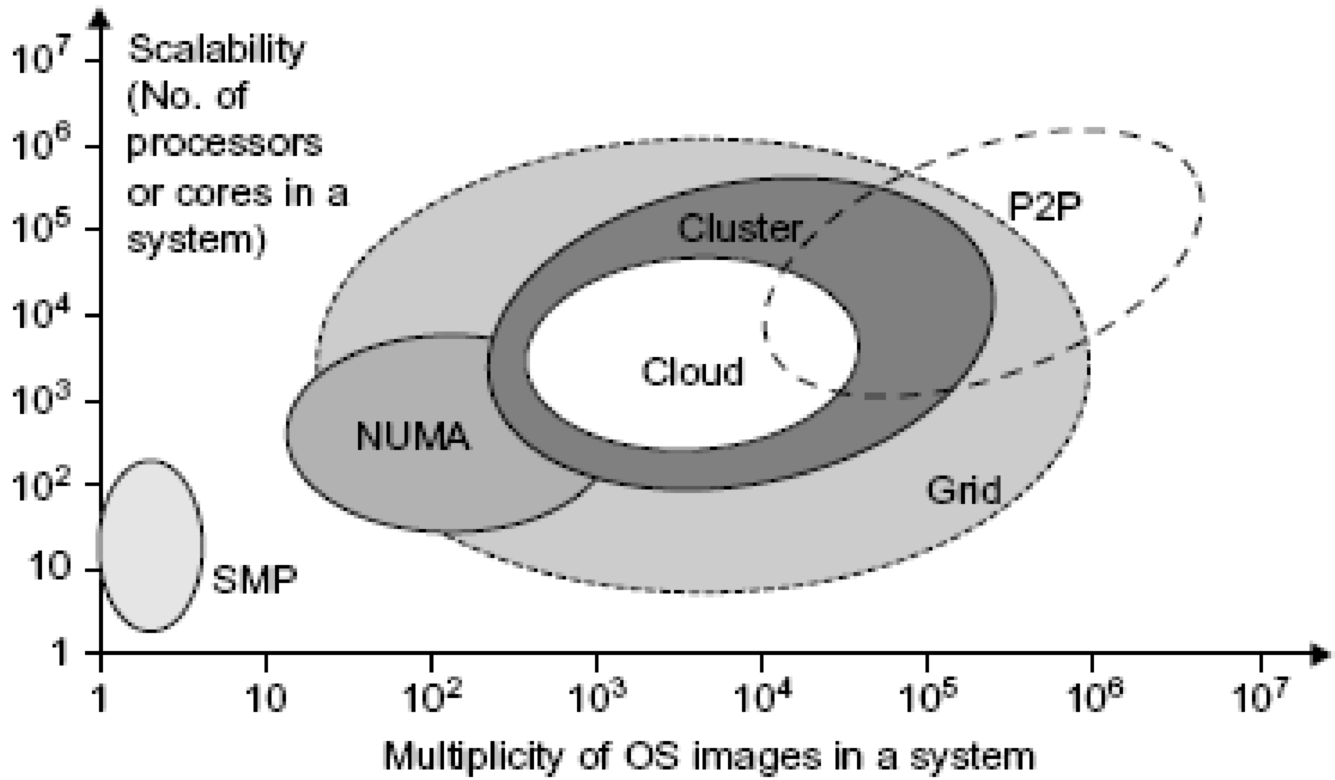


FIGURE 1.23

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

System Availability vs. Configuration Size :

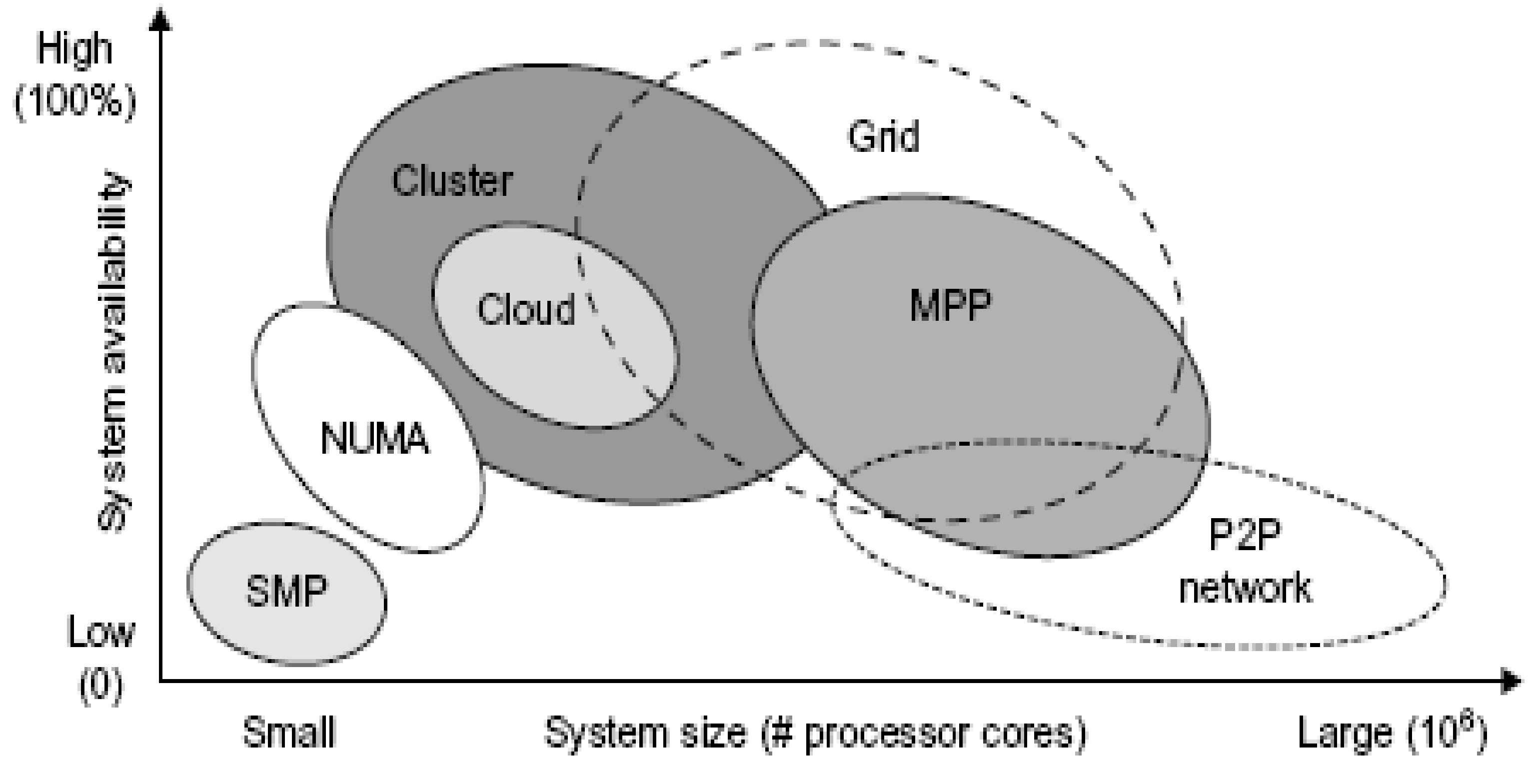


FIGURE 1.24

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

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

Transparent Cloud Computing Environment

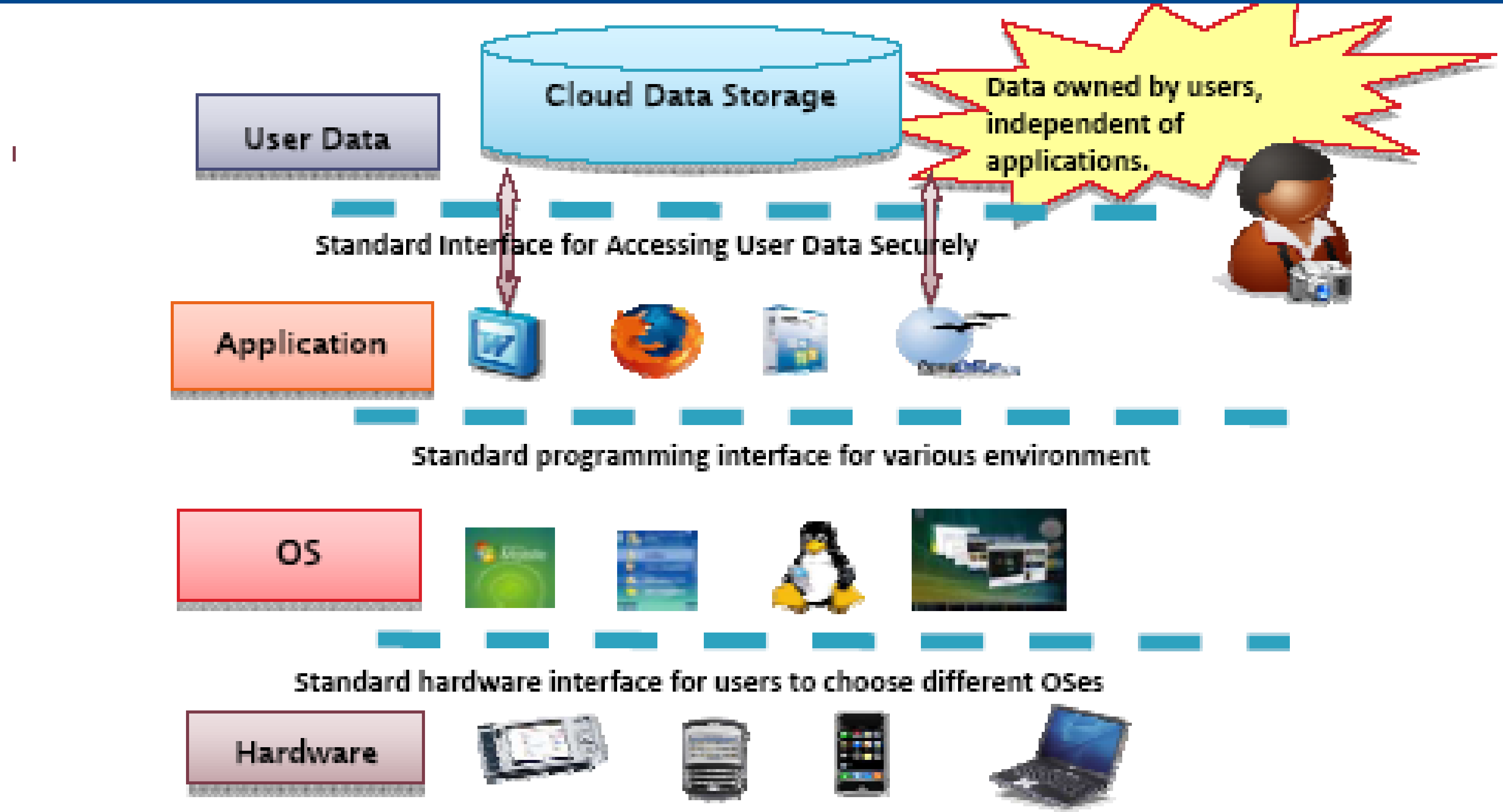


Figure 3 Transparent computing that separates the user data, application, OS, and hardware in time and space – an ideal model for future Cloud platform construction

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.

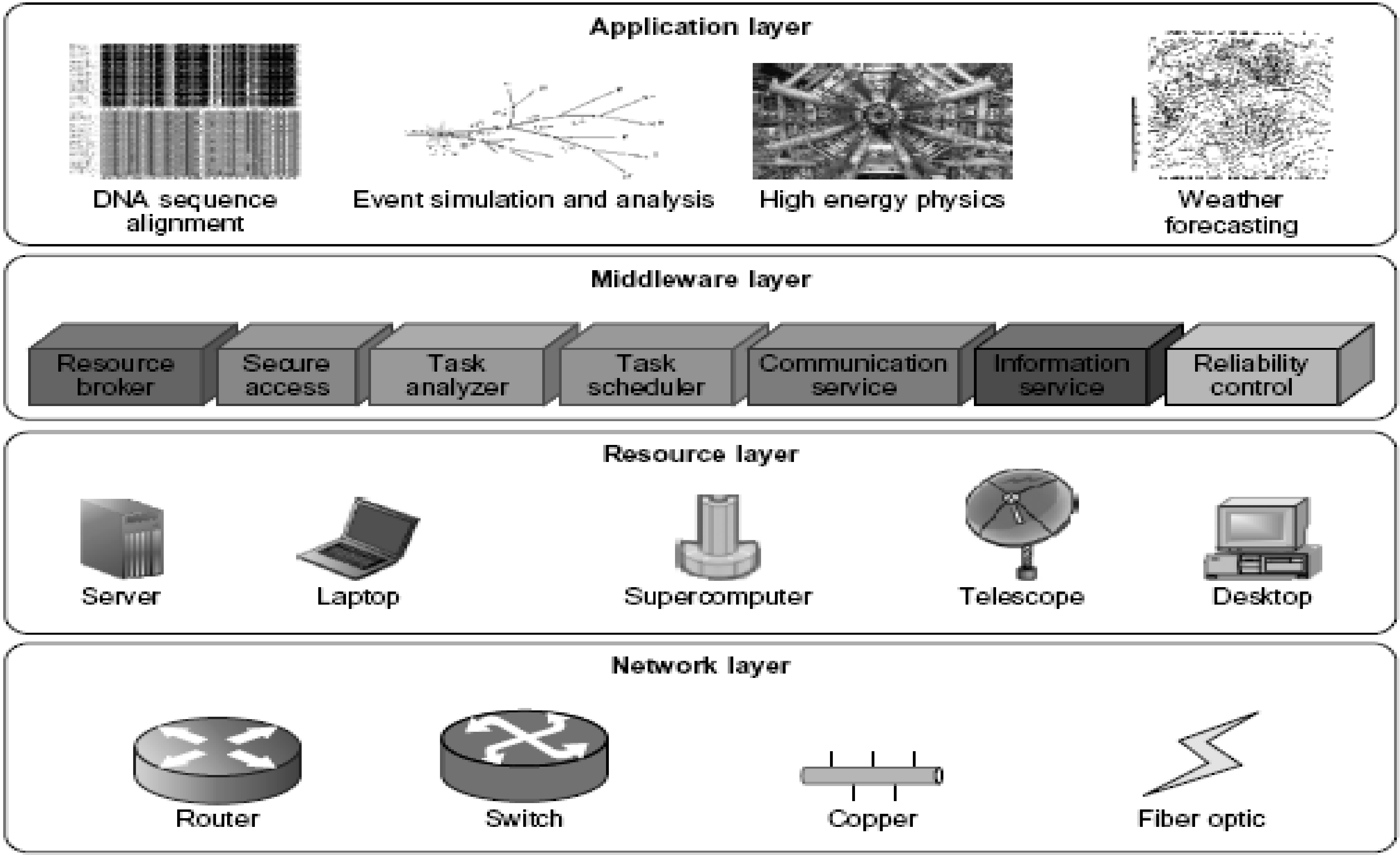


FIGURE 1.26

Four operational layers of distributed computing systems.

(Courtesy of Zomaya, Rivandi and Lee of the University of Sydney [33])

Energy Efficiency :

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

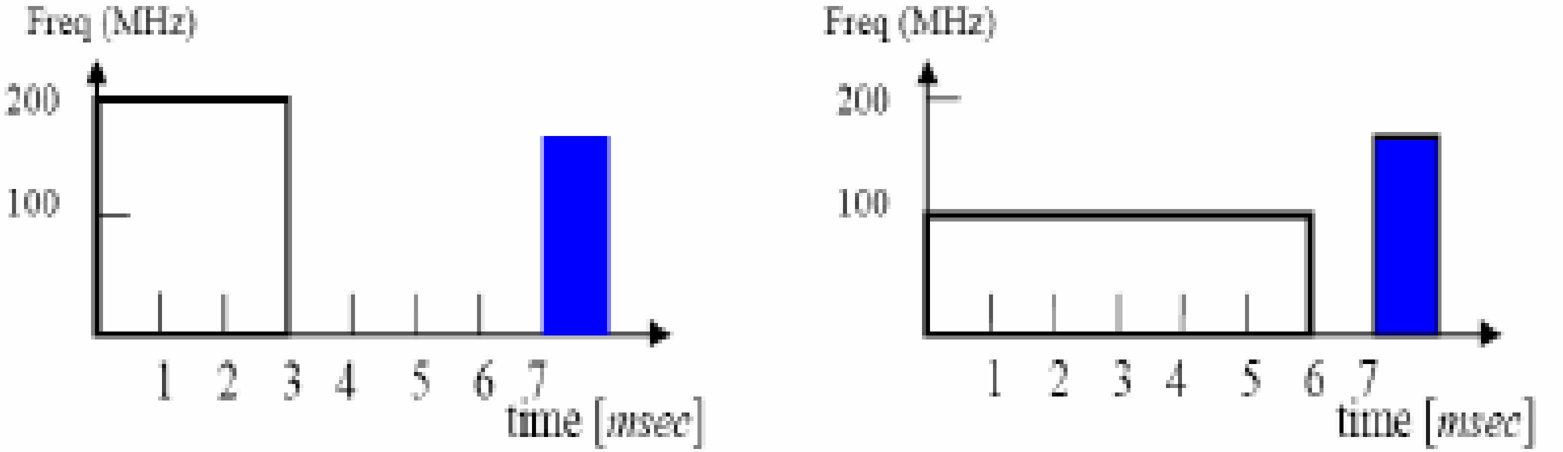


Figure 1.30 DVFS technique (right) original task (left) voltage-frequency scaled task (Courtesy of R.Ge, et al, "Performance Constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters", *Proc. of ACM Supercomputing Conf.*, 2005 [18].)

System Attacks and Network Threads

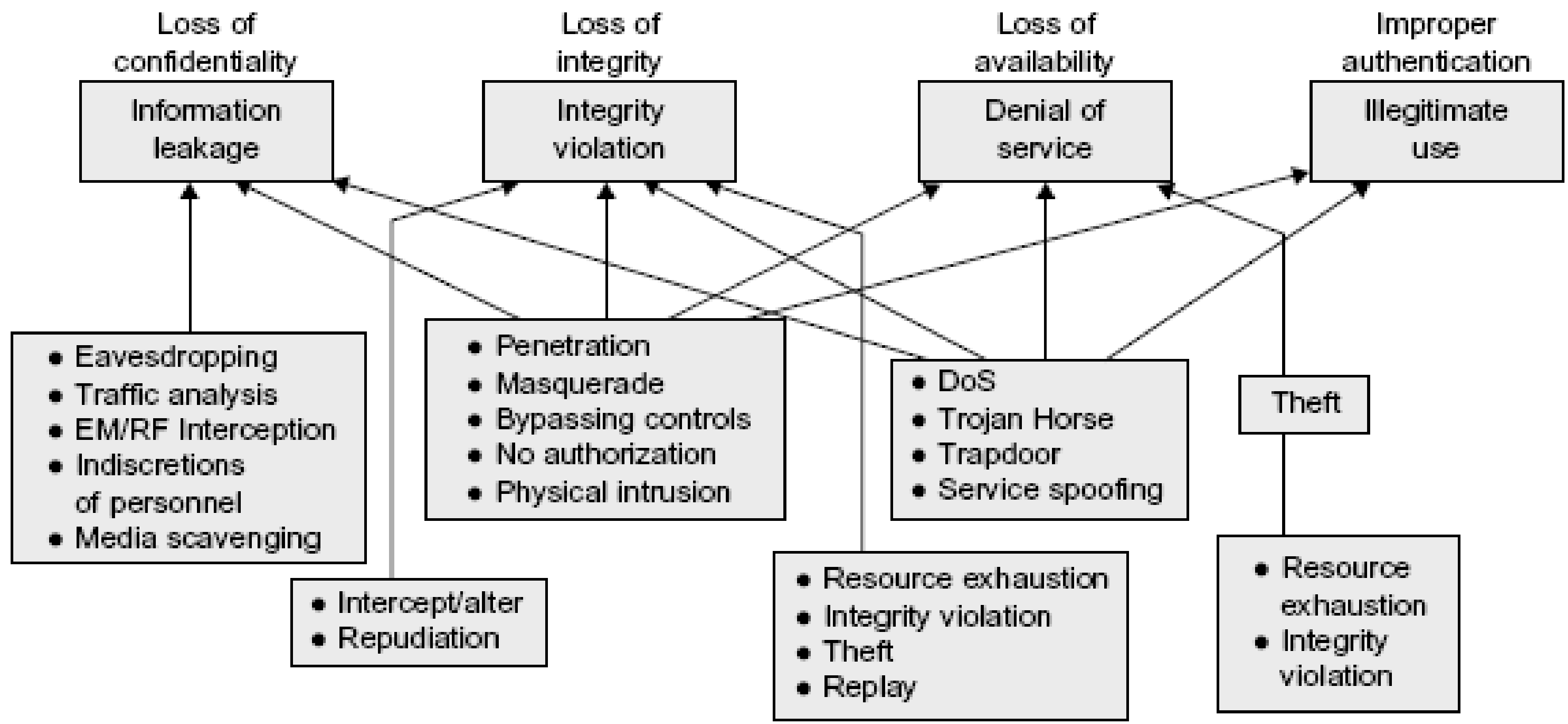


FIGURE 1.25

Various system attacks and network threats to the cyberspace.

Four Reference Books:

1. K. Hwang, G. Fox, and J. Dongarra, *Distributed and Cloud Computing: from Parallel Processing to the Internet of Things* *Morgan Kauffmann Publishers*, 2011

2. R. Buyya, J. Broberg, and A. Goscinski (eds), Cloud Computing: Principles and Paradigms, ISBN-13: 978-0470887998, Wiley Press, USA, February 2011.

3. T. Chou, *Introduction to Cloud Computing: Business and Technology,* Lecture Notes at Stanford University and at Tsinghua University, Active Book Press, 2010.

4. T. Hey, Tansley and Tolle (Editors), *The Fourth Paradigm : Data-Intensive Scientific Discovery,* Microsoft Research, 2009.