NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE
(NAAC Accredited)
(Approved by AICTE, Affiliated to APJ Abdul Kalam Technological University, Kerala)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE MATERIALS

CS468 CLOUD COMPUTING

VISION OF THE INSTITUTION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

MISSION OF THE INSTITUTION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

ABOUT DEPARTMENT

- ♦ Established in: 2002
- ♦ Course offered : B.Tech in Computer Science and Engineering

  M.Tech in Computer Science and Engineering

  M.Tech in Cyber Security
- ♦ Approved by AICTE New Delhi and Accredited by NAAC
- ♦ Affiliated to the University of A P J Abdul Kalam Technological University.

DEPARTMENT VISION

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering Professionals to facilitate continuous technological advancement.

DEPARTMENT MISSION

1. To Impart Quality Education by creative Teaching Learning Process
2. To Promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
3. To Inculcate Entrepreneurship Skills among Students.
4. To cultivate Moral and Ethical Values in their Profession.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.

PEO2: Graduates will be able to Analyse, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.

PEO3: Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.

PEO4: Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Teamworkand leadership qualities.

PROGRAM OUTCOMES (POS)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1: Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

PSO2: Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance optimization.

PSO3: Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

COURSE OUTCOMES

CO1	Identify the significance of implementing virtualization techniques
CO2	Interpret the various cloud computing models and services
CO3	Compare the various public cloud platforms and software environments
CO4	Apply appropriate cloud programming methods to solve big data problems
CO5	Describe the need of security mechanisms in cloud
CO6	Illustrate the use of various cloud services available online

MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	**3**	**3**	**2**	**2**	**3**							**2**
CO2	**3**	**2**	**2**	**3**	**3**							**2**
CO3	**3**	**2**	**2**		**3**							**2**
CO4	**3**	**2**	**3**	**3**	**3**							**2**
CO5	**3**	**3**				**3**		**2**				**2**
CO6	**3**	**3**	**3**	**3**	**3**	**3**		**2**	**3**	**3**	**3**	**2**

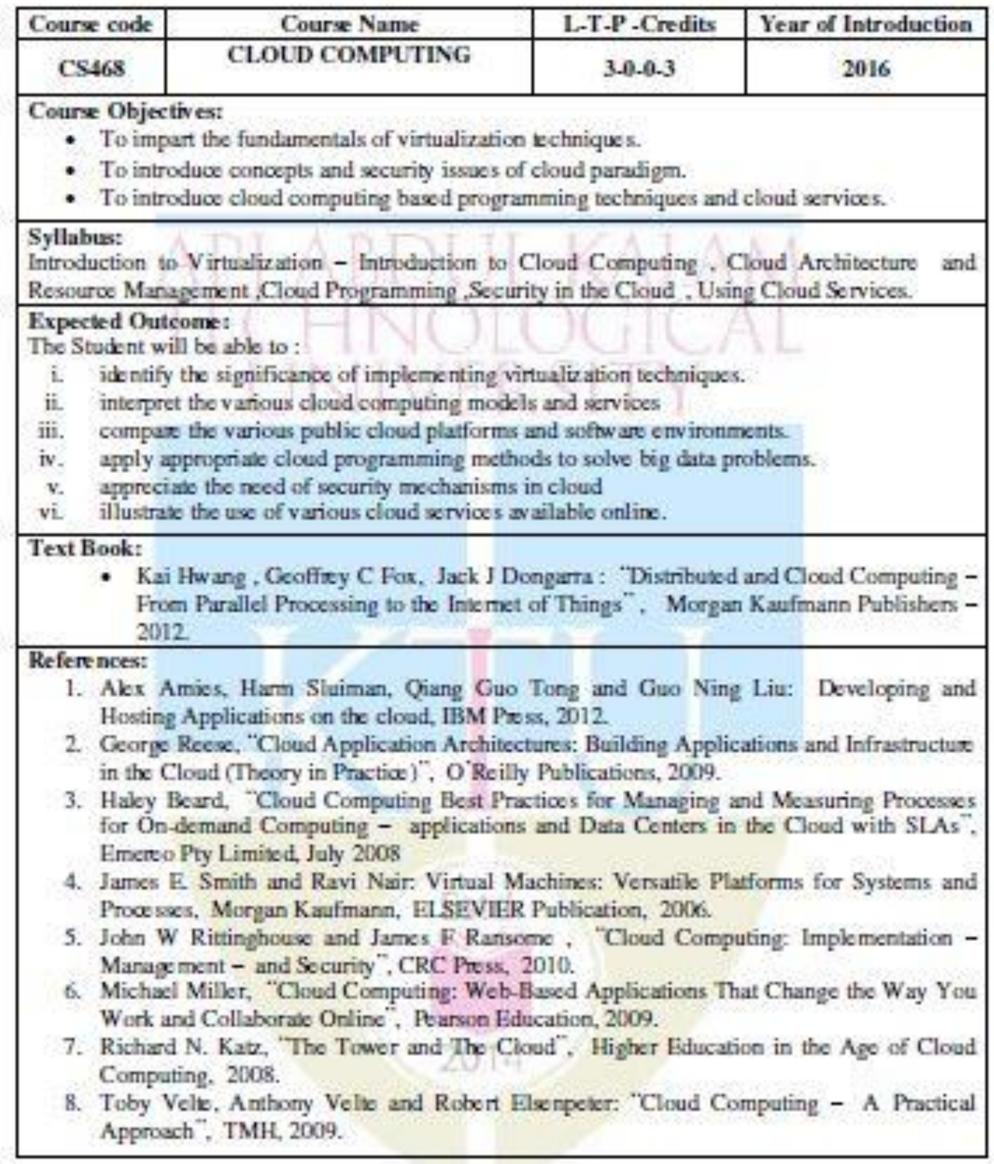
Note: H-Highly correlated=3, M-Medium correlated=2, L-Less correlated=1

MAPPING OF COURSE OUTCOMES WITH PROGRAM SPECIFIC OUTCOMES

	PSO 1	PSO 2	PSO 3
CO1	2	3	
CO2	3	3	
CO3	3	3	
CO4	3	3	
CO5	3	3	
CO6	3	3	3

Note: H-Highly correlated=3, M-Medium correlated=2, L-Less correlated=1

SYLLABUS



Course code	Course Name	L-T-P -Credits	Year of Introduction
CS468	CLOUD COMPUTING	3-0-0-3	2016

Course Objectives:

- To impart the fundamentals of virtualization techniques.
- To introduce concepts and security issues of cloud paradigm.
- To introduce cloud computing based programming techniques and cloud services.

Syllabus:
Introduction to Virtualization – Introduction to Cloud Computing , Cloud Architecture and Resource Management ,Cloud Programming ,Security in the Cloud , Using Cloud Services.

Expected Outcome:
The Student will be able to :

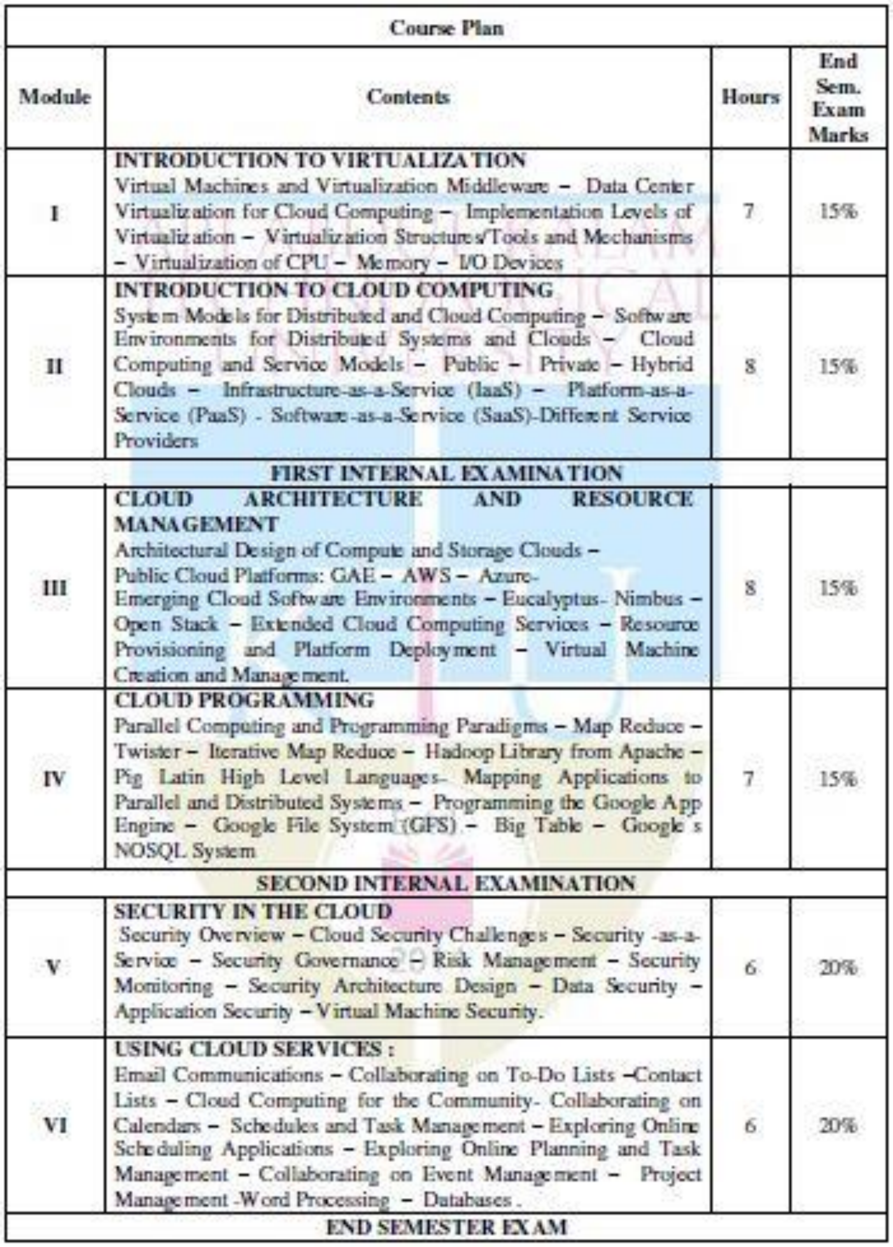
i. identify the significance of implementing virtualization techniques.
ii. interpret the various cloud computing models and services
iii. compare the various public cloud platforms and software environments.
iv. apply appropriate cloud programming methods to solve big data problems.
v. appreciate the need of security mechanisms in cloud
vi. illustrate the use of various cloud services available online.

Text Book:

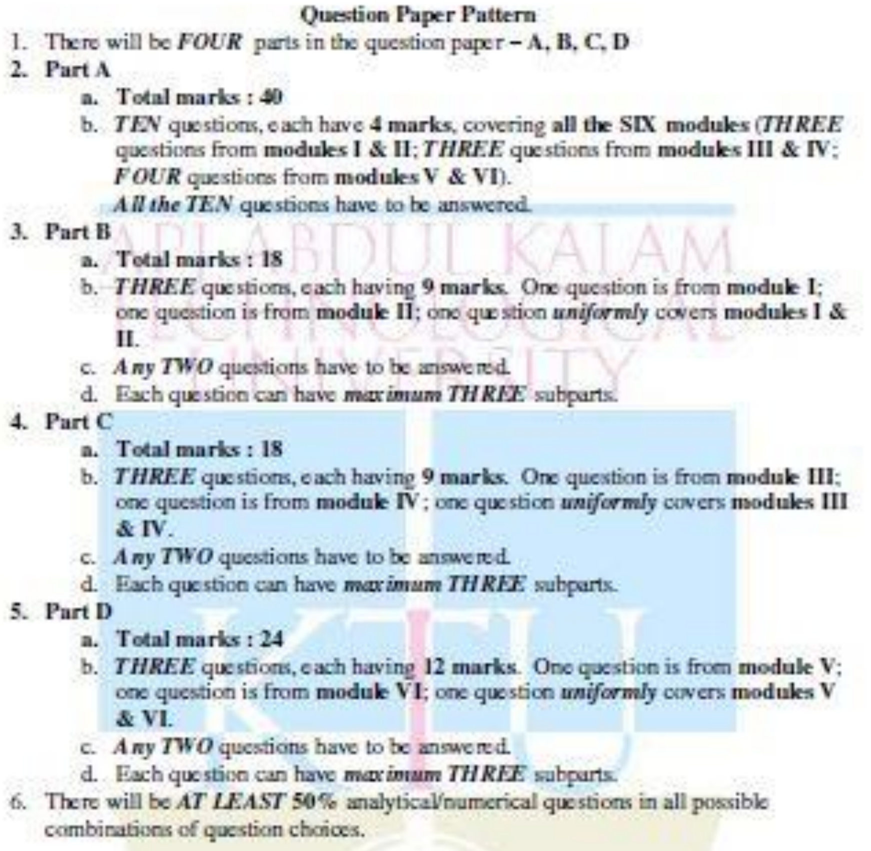
- Kai Hwang , Geoffrey C Fox, Jack J Dongarra : “Distributed and Cloud Computing – From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers – 2012.

References:

1. Alex Amies, Harm Sluiman, Qiang Guo Tong and Guo Ning Liu: Developing and Hosting Applications on the cloud, IBM Press, 2012.
2. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud (Theory in Practice)”, O'Reilly Publications, 2009.
3. Haley Beard, “Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing – applications and Data Centers in the Cloud with SLAs”, Emereo Pty Limited, July 2008
4. James E. Smith and Ravi Nair: Virtual Machines: Versatile Platforms for Systems and Processes, Morgan Kaufmann, ELSEVIER Publication, 2006.
5. John W Rittinghouse and James F Ransome , “Cloud Computing: Implementation – Management – and Security”, CRC Press, 2010.
6. Michael Miller, “Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online”, Pearson Education, 2009.
7. Richard N. Katz, “The Tower and The Cloud”, Higher Education in the Age of Cloud Computing, 2008.
8. Toby Velte, Anthony Velte and Robert Elsenpeter: “Cloud Computing – A Practical Approach”, TMH, 2009.



Course Plan			
Module	Contents	Hours	End Sem. Exam Marks
I	**INTRODUCTION TO VIRTUALIZATION** Virtual Machines and Virtualization Middleware – Data Center Virtualization for Cloud Computing – Implementation Levels of Virtualization – Virtualization Structures/Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices	7	15%
II	**INTRODUCTION TO CLOUD COMPUTING** System Models for Distributed and Cloud Computing – Software Environments for Distributed Systems and Clouds – Cloud Computing and Service Models – Public – Private – Hybrid Clouds – Infrastructure-as-a-Service (IaaS) – Platform-as-a-Service (PaaS) - Software-as-a-Service (SaaS)-Different Service Providers	8	15%
FIRST INTERNAL EXAMINATION			
III	**CLOUD ARCHITECTURE AND RESOURCE MANAGEMENT** Architectural Design of Compute and Storage Clouds – Public Cloud Platforms: GAE – AWS – Azure- Emerging Cloud Software Environments – Eucalyptus- Nimbus – Open Stack – Extended Cloud Computing Services – Resource Provisioning and Platform Deployment – Virtual Machine Creation and Management.	8	15%
IV	**CLOUD PROGRAMMING** Parallel Computing and Programming Paradigms – Map Reduce – Twister – Iterative Map Reduce – Hadoop Library from Apache – Pig Latin High Level Languages- Mapping Applications to Parallel and Distributed Systems – Programming the Google App Engine – Google File System (GFS) – Big Table – Google s NOSQL System	7	15%
SECOND INTERNAL EXAMINATION			
V	**SECURITY IN THE CLOUD** Security Overview – Cloud Security Challenges – Security -as-a-Service – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security.	6	20%
VI	**USING CLOUD SERVICES :** Email Communications – Collaborating on To-Do Lists –Contact Lists – Cloud Computing for the Community- Collaborating on Calendars – Schedules and Task Management – Exploring Online Scheduling Applications – Exploring Online Planning and Task Management – Collaborating on Event Management – Project Management -Word Processing – Databases .	6	20%
END SEMESTER EXAM			



Question Paper Pattern

1. There will be ***FOUR*** parts in the question paper – **A, B, C, D**
2. **Part A**
   a. **Total marks : 40**
   b. ***TEN*** questions, each have **4 marks**, covering **all the SIX modules** (***THREE*** questions from **modules I & II**; ***THREE*** questions from **modules III & IV**; ***FOUR*** questions from **modules V & VI**).
   All the TEN questions have to be answered.
3. **Part B**
   a. **Total marks : 18**
   b. ***THREE*** questions, each having **9 marks**. One question is from **module I**; one question is from **module II**; one question ***uniformly*** covers **modules I & II.**
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
4. **Part C**
   a. **Total marks : 18**
   b. ***THREE*** questions, each having **9 marks**. One question is from **module III**; one question is from **module IV**; one question ***uniformly*** covers **modules III & IV**.
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
5. **Part D**
   a. **Total marks : 24**
   b. ***THREE*** questions, each having **12 marks**. One question is from **module V**; one question is from **module VI**; one question ***uniformly*** covers **modules V & VI.**
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
6. There will be ***AT LEAST* 50%** analytical/numerical questions in all possible combinations of question choices.

QUESTION BANK

MODULE I				
Q:NO:	**QUESTIONS**	**CO**	**KL**	**PAGE NO:**
1	Explain the VM architectures in detail.	CO1	K2	16
2	Explain the levels of virtualization implementation with neat diagram.	CO1	K5	22
3	Briefly describe the different types of VM configurations.	CO1	K2	17
4	Point out VM primitive operations.	CO1	K4	18
5	Explain binary translation with full virtualization	CO1	K2	34
6	Write note on XEN architecture with a neat diagram	CO1	K6	32
7	Differentiate between CPU virtualization and memory virtualization	CO1	K4	39, 41
8	Explain about virtual machines.	CO1	K2	
9	Write note on CPU virtualization.	CO1	K6	39
10	Write notes on: (a) I/O Virtualization and (b) Virtualization in multi-core processors.	CO1	K6	43
11	Explain the concept of data center virtualization for cloud computing.	CO1	K2	19

12	Explain OS virtualization with its advantages and disadvantages.	CO1	K5	29
MODULE II				
1	Explain public, private and hybrid clouds.	CO2	K5	62
2	Explain cloud ecosystems with neat diagram.	CO2	K5	65
3	Explain the concept of clusters of cooperative computers.	CO2	K2	47
4	Write short note on grid computing infrastructures.	CO2	K6	50
5	Describe parallel and distributed programming models.	CO2	K2	46
6	List out the cloud computing models.	CO2	K4	61
7	List out cloud design objectives.	CO2	K4	64
8	Briefly describe Peer-to-Peer Network families.	CO2	K2	53
9	Differentiate between PaaS and SaaS.	CO2	K4	38
10	Explain layered architecture for web services and grids.	CO2	K2	51
11	Write note on Infrastructure-as-a-Service(IaaS).	CO2	K6	38

12	Explain the concept of Service Oriented Architecture in detail.	CO2	K5	59
MODULE III				
1	List out the functional modules of GAE.	CO3	K4	86
2	Write note on sector/sphere.	CO3	K6	97
3	Explain Virtualization support and disaster recovery.	CO3	K2	74
4	Write note on GAE with neat diagram.	CO3	K6	83
5	Explain market oriented cloud architecture.	CO3	K2	71
6	Explain the public cloud platforms in detail.	CO3	K5	82
7	Explain briefly about Manjrasoft Aneka Cloud and Appliances.	CO3	K5	104
8	Write note on OpenStack.	CO3	K6	100
9	Describe about Microsoft Windows Azure. Briefly describe the architectural design challenges.	CO3	K4	90
10	Write short note on Amazon Web Service	CO3	K6	88
11	Explain Nimbus with a neat diagram.	CO3	K5	94

MODULE IV				
1	Point out the issue in parallel computing.	CO4	K4	116
2	Explain the concept of programming GAE.	CO4	K2	135
3	Explain about the building blocks of Big Table	CO4	K5	143
4	Explain the architecture of GFS.	CO4	K5	139
5	Explain Pig-Latin and its operators	CO4	K2	131
6	Briefly explain application classification for parallel and distributed systems.	CO4	K2	133
7	Write note on Hadoop library from Apache	CO4	K6	126
8	Point out the steps in mutation of Google file system.	CO4	K4	141
9	Define parallel computing and programming paradigm with its motivation.	CO4	K1	116
10	Write note on GFS.	CO4	K6	138
11	Explain map reduce with an example.	CO4	K2	117

12	Write note on iterative map reduce and twister.	CO4	K6	124
MODULE V				
1	Explain the benefits of cloud security monitoring.	CO5	K5	154
2	Explain the concept of virtual machine security	CO5	K2	163
3	Explain the dimensions of cloud security.	CO5	K2	147
4	Explain risk management in cloud.	CO5	K5	151
5	Point out the cloud security challenges	CO5	K4	157
6	Point out any five categories in security as a service.	CO5	K4	149
7	Write note on CSA with a neat diagram.	CO5	K6	155
8	Write note on security as a service in cloud.	CO5	K6	148
9	Explain briefly about authorization with diagram.	CO5	K5	162
10	Describe the key objectives of security governance.	CO5	K4	150
11	Write note on security overview in cloud.	CO5	K6	146

12	Write note on authentication.	CO5	K6	159
MODULE VI				
1	Explain about centralizing email communications.	CO6	K2	167
2	Differentiate between collaborating on to-do list and contact list	CO6	K4	168,169
3	Describe the method of exploring online planning and task management.	CO6	K4	194
4	Point out any five planning and task applications.	CO6	K4	194, 195
5	Write note on collaborating on schedules.	CO6	K6	172
6	Write notes on (i) Prsedo (ii) Windows live events (iii) Schedule book	CO6	K6	190, 191, 192
7	Write note on event management applications.	CO6	K6	204
8	Write on exploring online calendar applications	CO6	K6	179
9	Explain briefly about online scheduling applications.	CO6	K5	189

10	Explain the concept of Google calendar and yahoo calendar.	CO6	K2	180
11	Briefly describe collaborating on group projects and events.	CO6	K4	175
12	Explain the concept of cloud computing for community.	CO6	K5	170

APPENDIX 1

CONTENT BEYOND THE SYLLABUS

S:NO;	TOPIC	PAGE NO:
1	Cloud Security Challenges	222
2	Recent Trends in Cloud Computing	225

MODULE 1

INTRODUCTION TO VIRTUALIZATION

➢ VIRTUAL MACHINES AND VIRTUALIZATION MIDDLEWARE

- A conventional computer has a single OS image. This offers a rigid architecture that tightly couples application software to a specific hardware platform.
- Some software running well on one machine may not be executable on another platform with a different instruction set under a fixed OS.
- Virtual machines (VMs) offer novel solutions to underutilized resources, application inflexibility, software manageability, and security concerns in existing physical machines.

❖ Virtual Machines

- In Figure 1.1, the host machine is equipped with the physical hardware, as shown at the bottom of the figure. An example is an x-86 architecture desktop running its installed Windows OS, as shown in part (a) of the figure.
- 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).
- Figure 1.1(b) shows a native VM installed with the use of a VMM called a hypervisor in privileged mode. For example, the hardware has x-86 architecture running the Windows system. The guest OS could be a Linux system and the hypervisor is the XEN system developed at Cambridge

University. This hypervisor approach is also called bare-metal VM, because the hypervisor handles the bare hardware (CPU, memory, and I/O) directly.

- Another architecture is the host VM shown in Figure 1.1(c). Here the VMM runs in non-privileged mode. The host OS need not be modified.
- The VM can also be implemented with a dual mode, as shown in Figure 1.1(d). Part of the VMM runs at the user level and another part runs at the supervisor level. In this case, the host OS may have to be modified to some extent. Multiple VMs can be ported to a given hardware system to support the virtualization process.
- The VM approach offers hardware independence of the OS and applications. The user application running on its dedicated OS could be bundled together as a virtual appliance that can be ported to any hardware platform. The VM could run on an OS different from that of the host computer.

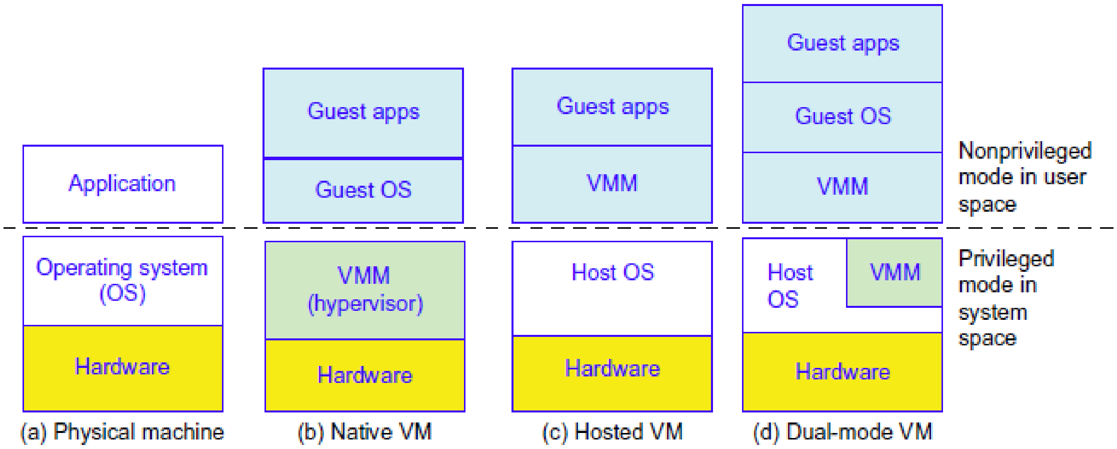


Figure 1.1 VM Configurations

❖ VM Primitive Operations

- The VMM provides the VM abstraction to the guest OS. With full virtualization, the VMM exports a VM abstraction identical to the physical machine so that a standard OS such as Windows 2000 or Linux can run just as it would on the physical hardware.
- Low-level VMM operations are indicated by Mendel Rosenblum and illustrated in Figure 1.2

  a) First, the VMs can be multiplexed between hardware machines, as shown in Figure 1.2(a).

  b) Second, a VM can be suspended and stored in stable storage, as shown in Figure 1.2(b).

  c) Third, a suspended VM can be resumed or provisioned to a new hardware platform, as shown in Figure 1.2(c).

  d) Finally, a VM can be migrated from one hardware platform to another, as shown in Figure 1.2(d).

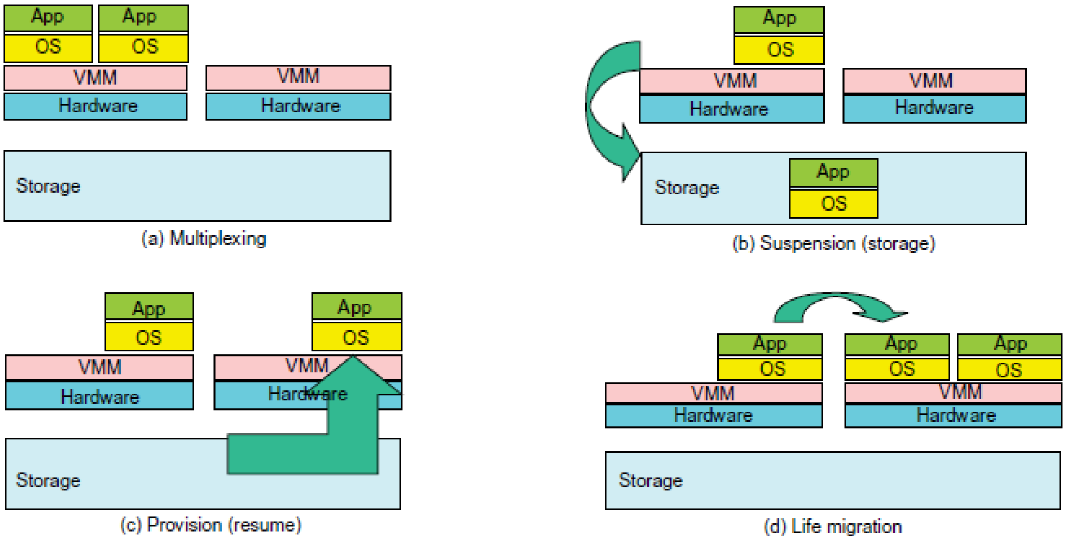


Figure 1.2 VM Primitive operations

- These VM operations enable a VM to be provisioned to any available hardware platform. They also enable flexibility in porting distributed application executions.
- Furthermore, the VM approach will significantly enhance the utilization of server resources. Multiple server functions can be consolidated on the same hardware platform to achieve higher system efficiency.
- This will eliminate server sprawl via deployment of systems as VMs, which move transparency to the shared hardware.

➢ DATA CENTER VIRTUALIZATION FOR CLOUD COMPUTING

- Cloud architecture is built with commodity hardware and network devices. Almost all cloud platforms choose the popularx86 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.
- In other words, storage and energy efficiency are more important than shear speed performance.

- Figure 1.3 shows the server growth and cost breakdown of data centers over the past 15 years. Worldwide, about 43 million servers are in use as of 2010. The cost of utilities exceeds the cost of hardware after three years.

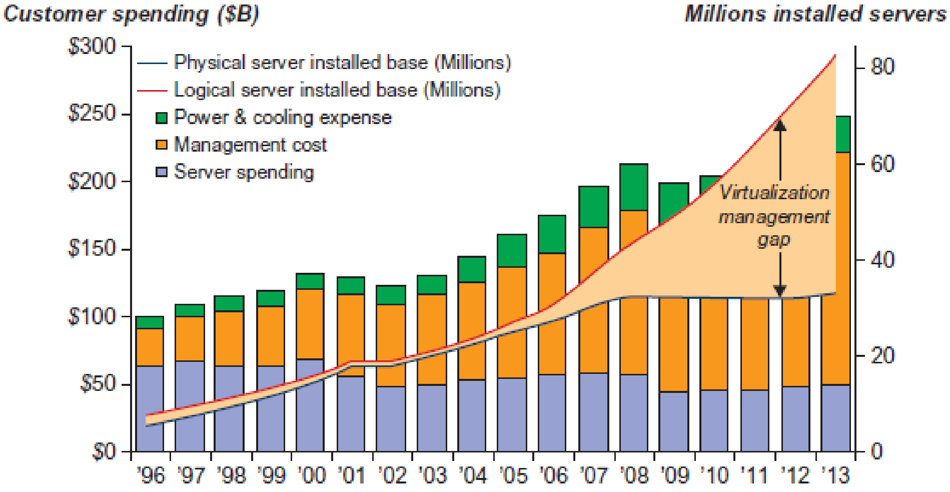


Figure 1.3 Growth and cost breakdown of data centers over the years.

❖ Data Center Growth and Cost Breakdown

- A large data center may be built with thousands of servers. Smaller data centers are typically built with hundreds of servers. The cost to build and maintain data center servers has increased over the years.
- According to a 2009 IDC report, typically only 30 percent of data center costs goes toward purchasing IT equipment (such as servers and disks), 33 percent is attributed to the chiller, 18 percent to the uninterruptible power supply (UPS), 9 percent to computer room air conditioning (CRAC), and

the remaining 7 percent to power distribution, lighting, and transformer costs.

- Thus, about 60 percent of the cost to run a data center is allocated to management and maintenance.
- The server purchase cost did not increase much with time. The cost of electricity and cooling did increase from 5 percent to 14 percent in 15 years.

❖ Low-Cost Design Philosophy

- High-end switches or routers may be too cost-prohibitive for building data centers. Thus, using high-bandwidth networks may not fit the economics of cloud computing.
- Given a fixed budget, commodity switches and networks are more desirable in data centers. Similarly, using commodity x86 servers is more desired over expensive mainframes.
- The software layer handles network traffic balancing, fault tolerance, and expandability. Currently, nearly all cloud computing data centers use Ethernet as their fundamental network technology.

❖ Convergence of Technologies

- Essentially, cloud computing is enabled by the convergence of technologies in four areas: (1) hardware virtualization and multi-core chips, (2) utility and grid computing, (3) SOA, Web 2.0, and WS mashups, and (4) atonomic 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, and mashups of platforms are pushing the cloud another step forward.
- Finally, achievements in autonomic computing and automated data center operations contribute to the rise of cloud computing.

➢ IMPLEMENTATION LEVELS OF VIRTUALIZATION

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.
- The idea of VMs can be dated back to the 1960s. The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.
- Hardware resources (CPU, memory, I/O devices, etc.) or software resources (operating system and software libraries) can be virtualized in various functional layers.
- This virtualization technology has been revitalized as the demand for distributed and cloud computing increased sharply in recent years.
- The idea is to separate the hardware from the software to yield better system efficiency.

- For example, computer users gained access to much enlarged memory space when the concept of virtual memory was introduced. Similarly, virtualization techniques can be applied to enhance the use of compute engines, networks, and storage.

❖ Levels of Virtualization Implementation

- A traditional computer runs with a host operating system specially tailored for its hardware architecture, as shown in Figure 1.4(a).
- After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS.
- This is often done by adding additional software, called a virtualization layer as shown in Figure 1.4(b). This virtualization layer is known as hypervisor or virtual machine monitor (VMM) .
- The VMs are shown in the upper boxes, where applications run with their own guest OS over the virtualized CPU, memory, and I/O resources.
- The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs, exclusively.
- This can be implemented at various operational levels and the virtualization software creates the abstraction of VMs by interposing a virtualization layer at various levels of a computer system.

- Common virtualization layers include the instruction set architecture (ISA) level, hardware level, operating system level, library support level, and application level.

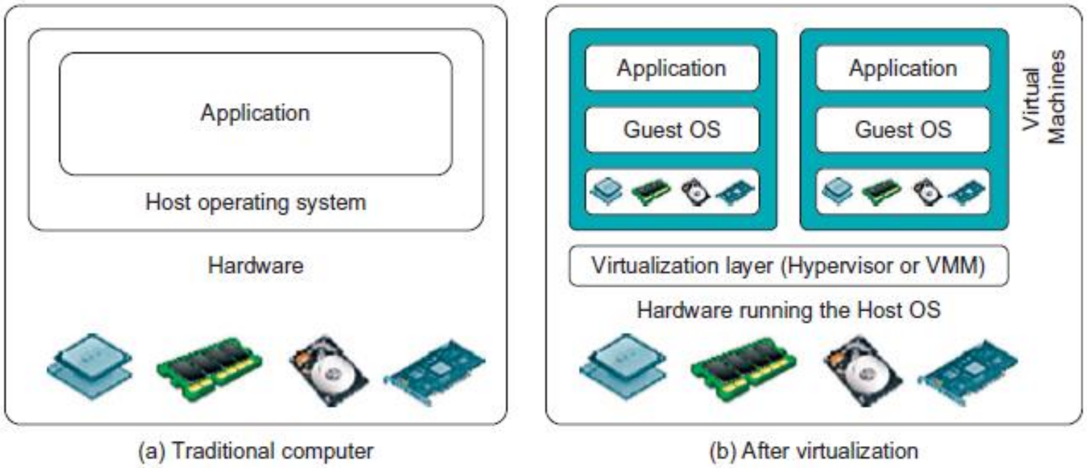


Figure 1.4 The architecture of a computer system before and after virtualization

(a) Instruction Set Architecture Level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine.
- With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.
- Instruction set emulation leads to virtual ISAs created on any hardware machine. The basic emulation method is through code interpretation.
- An interpreter program interprets the source instructions to target instructions one by one. One source instruction may require tens or

hundreds of native target instructions to perform its function. Obviously, this process is relatively slow.

- For better performance, dynamic binary translation is desired. This approach translates basic blocks of dynamic source instructions to target instructions.
- The basic blocks can also be extended to program traces or super blocks to increase translation efficiency. Instruction set emulation requires binary translation and optimization.
- A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

(b) Hardware Abstraction Level

- Hardware-level virtualization is performed right on top of the bare hardware.
- On the one hand, this approach generates a virtual hardware environment for a VM. On the other hand, the process manages the underlying hardware through virtualization.
- The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices.
- The intention is to upgrade the hardware utilization rate by multiple users concurrently.

(c) Operating System Level

- This refers to an abstraction layer between traditional OS and user applications. OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers.

- The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users.
- It is also used, to a lesser extent, in consolidating server hardware by moving services on separate hosts into containers or VMs on one server.

(d) Library Support Level

- Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS. Since most systems provide well-documented APIs, such an interface becomes another candidate for virtualization.
- Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks.
- The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts.

(e) User-Application Level

- Virtualization at the application level virtualizes an application as a VM. On a traditional OS, an application often runs as a process.

- Therefore, application-level virtualization is also known as process-level virtualization. The most popular approach is to deploy high level language (HLL) VMs.
- In this scenario, the virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.
- Any program written in the HLL and compiled for this VM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are two good examples of this class of VM.
- Other forms of application-level virtualization are known as application isolation, application sandboxing, or application streaming.
- The process involves wrapping the application in a layer that is isolated from the host OS and other applications. The result is an application that is much easier to distribute and remove from user workstations.

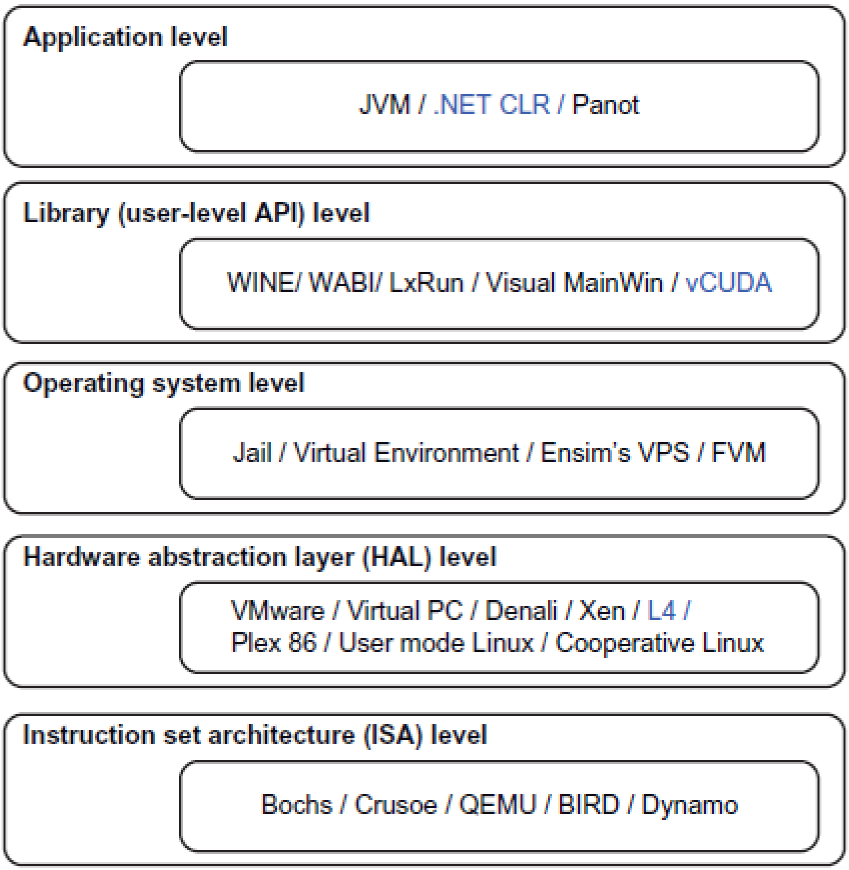

Application level
JVM / .NET CLR / Panot
Library (user-level API) level
WINE/ WABI/ LxRun / Visual MainWin / vCUDA
Operating system level
Jail / Virtual Environment / Ensim's VPS / FVM
Hardware abstraction layer (HAL) level
VMware / Virtual PC / Denali / Xen / L4 /
Plex 86 / User mode Linux / Cooperative Linux
Instruction set architecture (ISA) level
Bochs / Crusoe / QEMU / BIRD / Dynamo

Figure 1.5 Levels of Virtualization Implementation

❖ 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.
- Currently, new VMs originate either as fresh boots or as replicates of a template VM, unaware of the current application state. Therefore, to better support cloud computing, a large amount of research and development should be done.

Why OS-Level Virtualization?

- Operating system 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.
- Although VEs can be customized for different people, they share the same operating system kernel. Therefore, OS-level virtualization is also called single-OS image virtualization.

Advantages of OS Extensions

- Compared to hardware-level virtualization, the benefits of OS extensions are twofold:

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

  (2) for an OS-level VM, it is possible for a VM and its host environment to synchronize state changes when necessary.
- These benefits can be achieved via two mechanisms of OS-level virtualization:

  (1) All OS-level VMs on the same physical machine share a single operating system kernel; and

  (2) the virtualization layer can be designed in a way that allows processes in VMs to access as many resources of the host machine as possible, but never to modify them.

Disadvantages of OS Extensions

- 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.
- That is, although different OS-level VMs may have different operating system distributions, they must pertain to the same operating system family.

- For example, a Windows distribution such as Windows XP cannot run on a Linux-based container.
- However, users of cloud computing have various preferences. Some prefer Windows and others prefer Linux or other operating systems. Therefore, there is a challenge for OS-level virtualization in such cases.

➢ 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, paravirtualization, and host-based virtualization.
- The hypervisor is also known as the VMM (Virtual Machine Monitor). They both perform the same virtualization operations.

❖ Hypervisor and Xen Architecture

- The hypervisor supports hardware-level virtualization 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.
- A micro-kernel hypervisor includes only the basic and unchanging functions (such as physical memory management and processor scheduling). The device drivers and other changeable components are outside the hypervisor.
- A monolithic hypervisor implements all the aforementioned functions, including those of the device drivers. Therefore, the size of the hypervisor code of a micro-kernel hypervisor is smaller than that of a monolithic hypervisor.
- Essentially, a hypervisor must be able to convert physical devices into virtual resources dedicated for the deployed VM to use.

❖ The Xen Architecture

- Xen is an open source hypervisor program developed by Cambridge University. Xen is a microkernel 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 1.6. Xen does not include any device drivers natively .
- 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 and Oracle VM.
- 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).

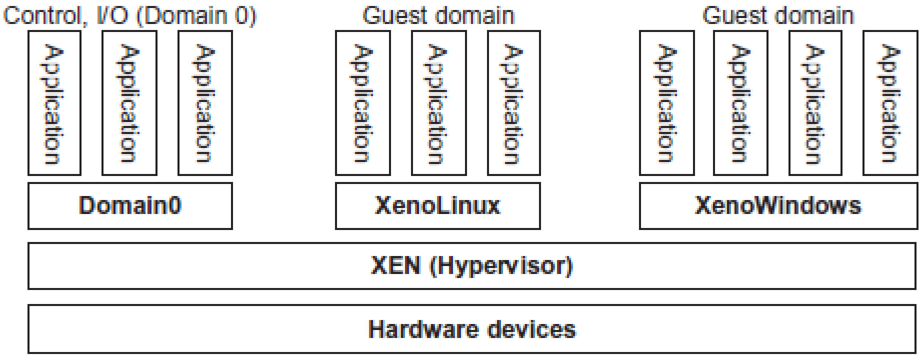


Figure 1.6 Xen Architecture

❖ **Binary Translation with Full Virtualization**

- Depending on implementation technologies, hardware virtualization can be classified into two categories: full virtualization and host-based virtualization.
- Full virtualization does not need to modify the host OS. It relies on binary translation to trap and to virtualize the execution of certain sensitive, nonvirtualizable instructions.
- The guest OSes and their applications consist of noncritical and critical instructions.
- In a host-based system, both a host OS and a guest OS are used. A virtualization layer is built between the host OS and guest OS.

Full Virtualization

- With full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software. Both the hypervisor and VMM approaches are considered full virtualization.
- Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do. Therefore, running noncritical instructions on hardware not only can promote efficiency, but also can ensure system security.

Binary Translation of Guest OS Requests Using a VMM

- As shown in figure 1.7, VMware puts the VMM at Ring 0 and the guest OS at Ring 1. The VMM scans the instruction stream and identifies the privileged, control- and behavior-sensitive instructions.
- When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions.
- The method used in this emulation is called binary translation. Therefore, full virtualization combines binary translation and direct execution.
- The guest OS is completely decoupled from the underlying hardware. Consequently, the guest OS is unaware that it is being virtualized.
- The performance of full virtualization may not be ideal, because it involves binary translation which is rather time-consuming. In

particular, the full virtualization of I/O-intensive applications is a really a big challenge.

.

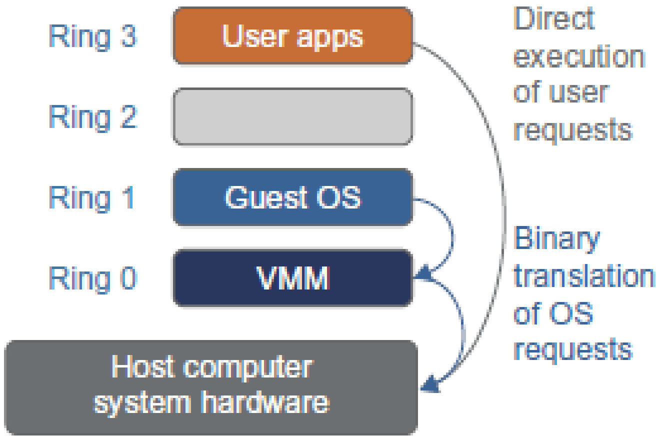

Ring 3
User apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host computer system hardware
Direct execution of user requests
Binary translation of OS requests

Figure 1.7: Binary Translation of Guest OS Requests Using a VMM

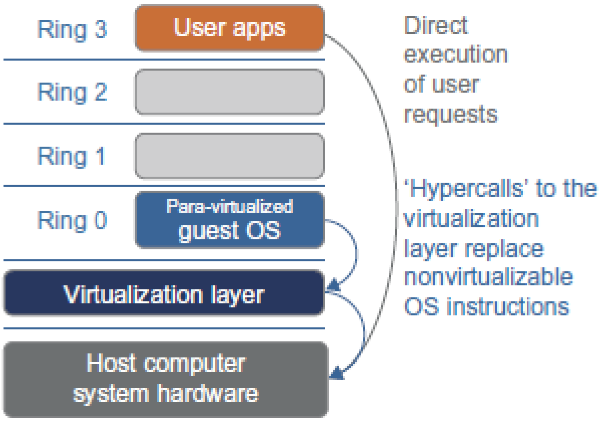
Host-Based Virtualization

- An alternative VM architecture is to install a 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.
- This host based architecture has some distinct advantages, as enumerated next. First, the user can install this VM architecture without modifying the host OS. The virtualizing software can rely on the host OS to provide device drivers and other low-level services.
- This will simplify the VM design and ease its deployment.

- Second, the host-based approach appeals to many host machine configurations. Compared to the hypervisor/VMM architecture, the performance of the host-based architecture may also be low.

❖ Para-Virtualization with Compiler Support

- Para-virtualization needs to modify the guest operating systems. A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications.
- Performance degradation is a critical issue of a virtualized system. No one wants to use a VM if it is much slower than using a physical machine.
- The virtualization layer can be inserted at different positions in a machine software stack. However, para-virtualization attempts to reduce the virtualization overhead, and thus improve performance by modifying only the guest OS kernel.
- Figure 1.8 illustrates the concept of a para-virtualized VM architecture. The guest operating systems are para-virtualized. They are assisted by an intelligent compiler to replace the nonvirtualizable OS instructions by hypercalls as illustrated in Figure 1.9.


Ring 3
User apps
Ring 2
Ring 1
Ring 0
Para-virtualized guest OS
Virtualization layer
Host computer system hardware
Direct execution of user requests
'Hypercalls' to the virtualization layer replace nonvirtualizable OS instructions

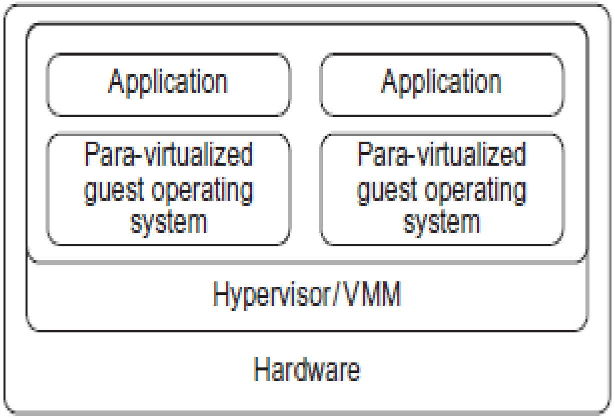

Application
Application
Para-virtualized guest operating system
Para-virtualized guest operating system
Hypervisor/VMM
Hardware

Figure 1.8: Para Virtualized Architecture

Figure 1.9: Para-virtualized guest OS assisted by an intelligent compiler

Para-Virtualization Architecture

- When the x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS.
- According to the x86 ring definition, the virtualization layer should also be installed at Ring 0. Different instructions at Ring 0 may cause some problems.
- In Figure 1.9, we show that para-virtualization replaces nonvirtualizable instructions with hypercalls that communicate directly with the hypervisor or VMM.
- However, when the guest OS kernel is modified for virtualization, it can no longer run on the hardware directly.
- Although para-virtualization reduces the overhead, it has incurred other problems. First, its compatibility and portability may be in doubt, because it must support the unmodified OS as well.

- Second, the cost of maintaining para-virtualized OSes is high, because they may require deep OS kernel modifications. Finally, the performance advantage of para-virtualization varies greatly due to workload variations.

➢ VIRTUALIZATION OF CPU, MEMORY, AND I/O DEVICES

- To support virtualization, processors such as the x86 employ a special running mode and instructions, known as hardware-assisted virtualization.
- In this way, the VMM and guest OS run in different modes and all sensitive instructions of the guest OS and its applications are trapped in the VMM. To save processor states, mode switching is completed by hardware.
- For the x86 architecture, Intel and AMD have proprietary technologies for hardware-assisted virtualization.

❖ Hardware Support for Virtualization

- Modern operating systems and processors permit multiple processes to run simultaneously. If there is no protection mechanism in a processor, all instructions from different processes will access the hardware directly and cause a system crash.
- Therefore, all processors have at least two modes, user mode and supervisor mode, to ensure controlled access of critical hardware.
- Instructions running in supervisor mode are called privileged instructions. Other instructions are unprivileged instructions.

- In a virtualized environment, it is more difficult to make OSes and applications run correctly because there are more layers in the machine stack.

❖ 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.
- Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency. Other critical instructions should be handled carefully for correctness and stability.
- The critical instructions are divided into three categories: privileged instructions, controlsensitive instructions, and behavior-sensitive instructions.
- Privileged instructions execute in a privileged 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.
- 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.
- However, not all CPU architectures are virtualizable. RISC CPU architectures can be naturally virtualized because all control- and behavior-sensitive instructions are privileged instructions.
- On the contrary, x86 CPU architectures are not primarily designed to support virtualization. This is because about 10 sensitive instructions, such as SGDT and SMSW, are not privileged instructions. When these instructions execute in virtualization, they cannot be trapped in the VMM.

❖ Memory Virtualization

- Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems.
- In a traditional execution environment, the operating system 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.

- 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.
- Furthermore, MMU virtualization should be supported, which is transparent to the guest OS.
- The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs. But the guest OS cannot directly access the actual machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory. Figure 1.10 shows the two-level memory mapping procedure.

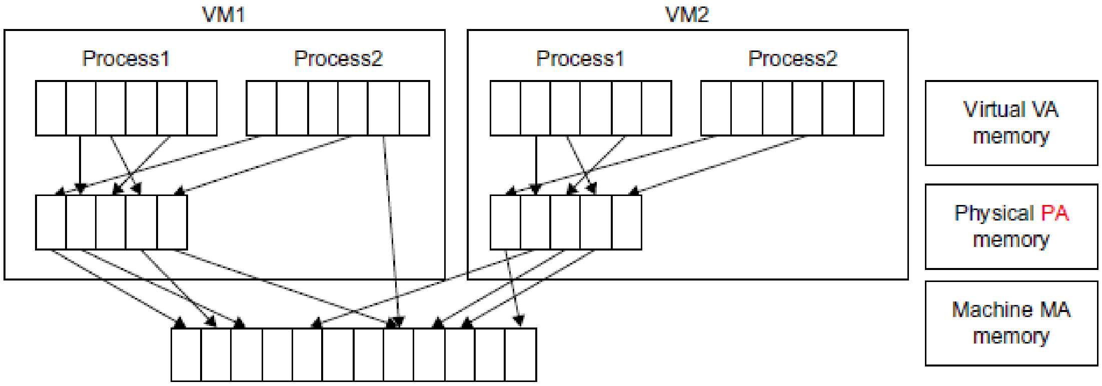


Figure 1.10 Two level memory mapping procedure

- Since each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the shadow page table.

- Nested page tables add another layer of indirection to virtual memory. The MMU already handles virtual-to-physical translations as defined by the OS.
- Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor. Since modern operating systems maintain a set of page tables for every process, the shadow page tables will get flooded.
- Consequently, the performance overhead and cost of memory will be very high. VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation.
- Processors use TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access.
- When the guest OS changes the virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup.

❖ I/O Virtualization

- I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.
- At the time of this writing, there are three 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. Generally, this approach emulates well-known, real-world devices.

- All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software.
- This software is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices. The full device emulation approach is shown in Figure 1.11.

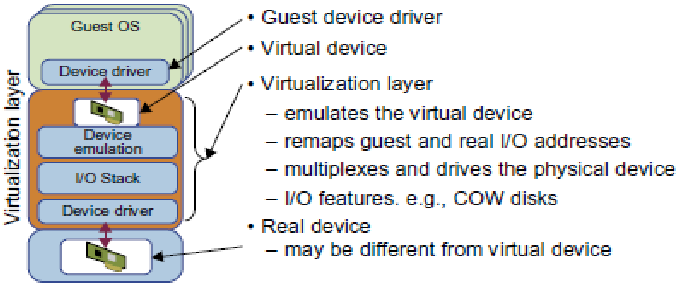

Guest OS
Device driver
Virtualization layer
Device emulation
I/O Stack
Device driver
• Guest device driver
• Virtual device
• Virtualization layer
– emulates the virtual device
– remaps guest and real I/O addresses
– multiplexes and drives the physical device
– I/O features. e.g., COW disks
• Real device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

Figure 1.11 Full device emulation

- A single hardware device can be shared by multiple VMs that run concurrently. However, software emulation runs much slower than the hardware it emulates [10,15].
- The para-virtualization method of I/O virtualization is typically used in Xen. It is also known as the split driver model consisting of a frontend driver and a backend driver.
- The frontend driver is running in Domain U and the backend driver is running in Domain 0. They interact with each other via a block of shared memory.
- 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 and multiplexing the I/O data of different VMs.

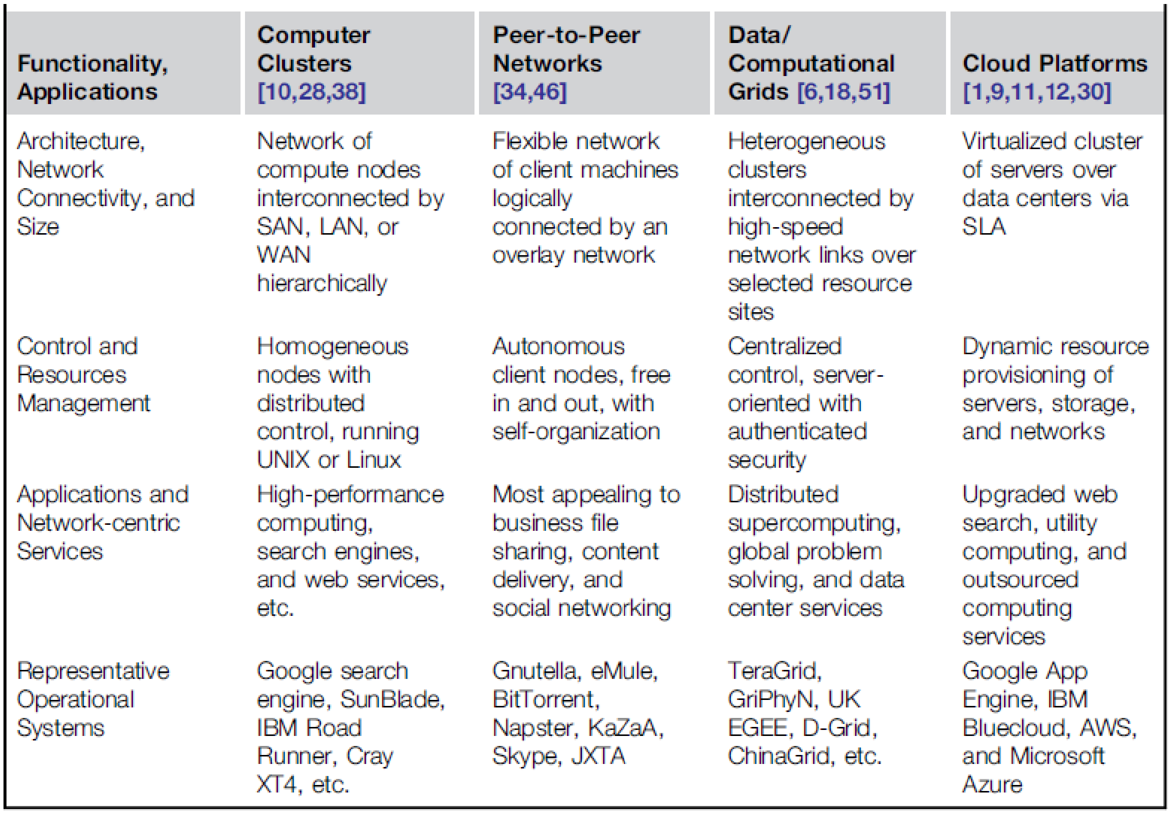
- Although para-I/O-virtualization achieves better device performance than full device emulation, it comes with a higher CPU overhead.
- Direct I/O virtualization lets the VM access devices directly. It can achieve close-to-native performance without high CPU costs.
- However, current direct I/O virtualization implementations focus on networking for mainframes. There are a lot of challenges for commodity hardware devices.

MODULE 2

INTRODUCTION TO CLOUD COMPUTING

➢ System Models For Distributed and Cloud Computing

- Distributed and cloud computing systems are built over a large number of autonomous computer nodes.
- These node machines are interconnected by SANs, LANs, or WANs in a hierarchical manner. With today’s networking technology, a few 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. In this sense, one can build a massive system with millions of computers connected to edge networks.
- Massive systems are considered highly scalable, and can reach web-scale connectivity, either physically or logically.
- In Table 2.1, massive systems are classified into four groups: clusters, P2P networks, computing grids, and Internet clouds over huge data centers.
- In terms of node number, these four system classes may involve hundreds, thousands, or even millions of computers as participating nodes.
- These machines work collectively, cooperatively, or collaboratively at various levels. The table entries characterize these four system classes in various technical and application aspects.



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

Table 2.1 Classification of Parallel and Distributed Computing Systems

❖ Clusters of Cooperative Computers

- A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.
- In the past, clustered computer systems have demonstrated impressive results in handling heavy workloads with large data sets.

Cluster Architecture

- Figure 2.1 shows the architecture of a typical server cluster built around a low-latency, high bandwidth interconnection network.

- This network can be as simple as a SAN (e.g., Myrinet) or a LAN (e.g., Ethernet). To build a larger cluster with more nodes, the interconnection network can be built with multiple levels of Gigabit Ethernet, Myrinet, or InfiniBand switches.
- Through hierarchical construction using a SAN, LAN, or WAN, one can build scalable clusters with an increasing number of nodes. 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.
- Most clusters have loosely coupled node computers. All resources of a server node are managed by their own OS. Thus, most clusters have multiple system images as a result of having many autonomous nodes under different OS control.

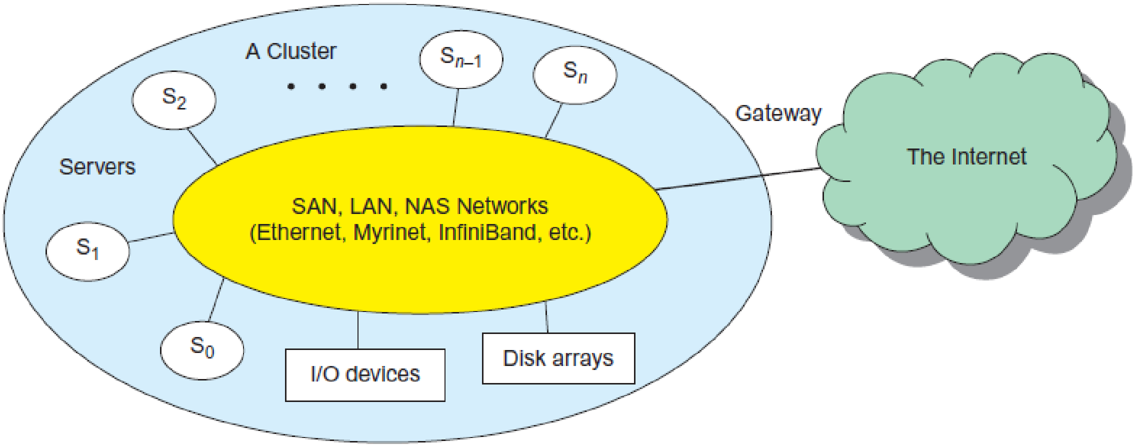


Figure 2.1 A cluster of servers interconnected by a high-bandwidth SAN or LAN with shared I/O devices and disk arrays

Single-System Image

- Greg Pfister has indicated that an ideal cluster should merge multiple system images into a single-system image (SSI).
- Cluster designers desire a cluster operating system or some middleware to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.
- 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 cluster with multiple system images is nothing but a collection of independent computers.

Hardware, Software, and Middleware Support

- Special cluster middleware supports are needed to create SSI or high availability (HA). Both sequential and parallel applications can run on the cluster, and special parallel environments are needed to facilitate use of the cluster resources.
- For example, distributed memory has multiple images. Users may want all distributed memory to be shared by all servers by forming distributed shared memory (DSM).
- Many SSI features are expensive or difficult to achieve at various cluster operational levels.
- Instead of achieving SSI, many clusters are loosely coupled machines. Using virtualization, one can build many virtual clusters dynamically, upon user demand.

❖ Grid Computing Infrastructures

Computational Grids

- Like an electric utility power grid, 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.
- Figure 2.2 shows an example computational grid built over multiple resource sites owned by different organizations.

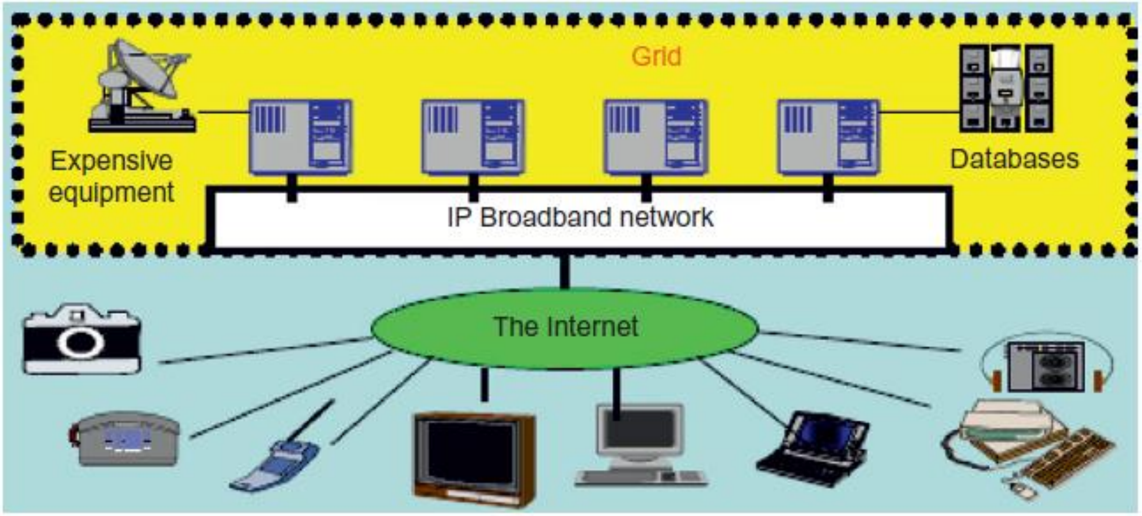


Figure 2.2 Computational Grid

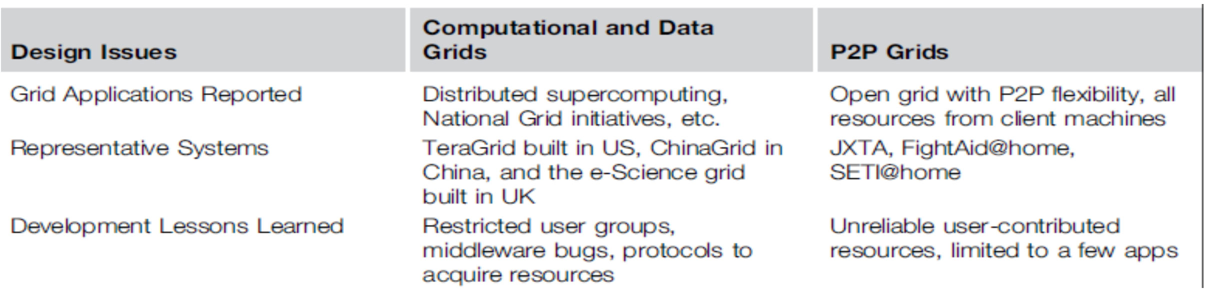
- The resource sites offer complementary computing resources, including workstations, large servers, a mesh of processors, and Linux clusters to satisfy a chain of computational needs.
- The grid is built across various IP broadband networks including LANs and WANs already used by enterprises or organizations over the Internet. The grid is presented to users as an integrated resource pool as shown in the upper half of the figure.
- Special instruments may be involved such as using the radio telescope in SETI@Home search of life in the galaxy and the austrophysics@Swineburne for pulsars.
- At the server end, the grid is a network. At the client end, we see wired or wireless terminal devices. The grid integrates the computing, communication, contents, and transactions as rented services.

- Enterprises and consumers form the user base, which then defines the usage trends and service characteristics.

Grid Families

- Grid technology demands new distributed computing models, software/middleware support, network protocols, and hardware infrastructures.
- National grid projects are followed by industrial grid platform development by IBM, Microsoft, Sun, HP, Dell, Cisco, EMC, Platform Computing, and others.
- New grid service providers (GSPs) and new grid applications have emerged rapidly, similar to the growth of Internet and web services in the past two decades.
- In Table 2.2, grid systems are classified in essentially two categories: computational or data grids and P2P grids.

Table 2.2 Grid Categories



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 Network Families

- The P2P architecture offers a distributed model of networked systems. First, a P2P network is client-oriented instead of server-oriented.

P2P Systems

- In a P2P system, every node acts as both a client and a server, providing part of the system resources.
- Peer machines are simply client computers connected to the Internet. All client machines act autonomously to join or leave the system freely.
- This implies that no master-slave relationship exists among the peers. No central coordination or central database is needed.
- In other words, no peer machine has a global view of the entire P2P system. The system is self-organizing with distributed control.
- Figure 2.3 shows the architecture of a P2P network at two abstraction levels.

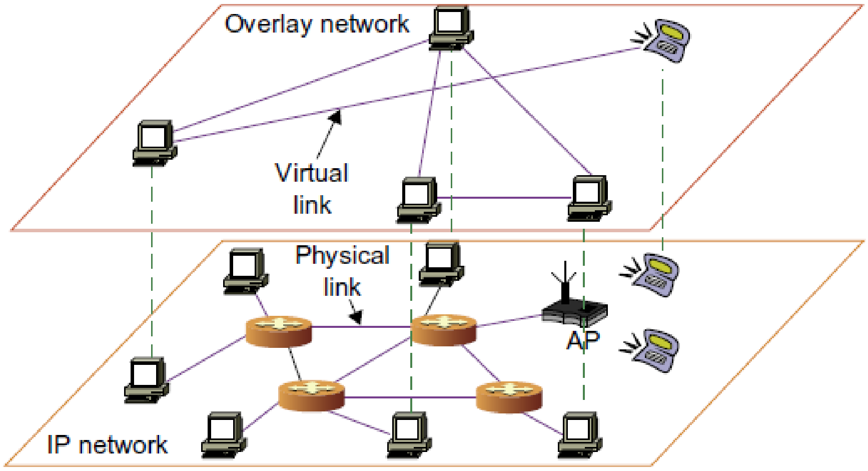


Figure 2.3 P2P network at two abstraction levels

- Initially, the peers are totally unrelated. Each peer machine joins or leaves the P2P network voluntarily.

- Only the participating peers form the physical network at any time. Unlike the cluster or grid, a P2P network does not use a dedicated interconnection network.
- The physical network is simply an ad hoc network formed at various Internet domains randomly using the TCP/IP and NAI protocols.
- Thus, the physical network varies in size and topology dynamically due to the free membership in the P2P network.

Overlay Networks

- Data items or files are distributed in the participating peers. Based on communication or file-sharing needs, the peer IDs form an overlay network at the logical level. This overlay is a virtual network.
- There are two types of overlay networks: unstructured and structured.
- An unstructured overlay network is characterized by a random graph. There is no fixed route to send messages or files among the nodes.
- Often, flooding is applied to send a query to all nodes in an unstructured overlay, thus resulting in heavy network traffic and nondeterministic search results.
- Structured overlay networks follow certain connectivity topology and rules for inserting and removing nodes (peer IDs) from the overlay graph. Routing mechanisms are developed to take advantage of the structured overlays.

P2P Application Families

- Based on application, P2P networks are classified into four groups.

- The first family is for distributed file sharing of digital contents (music, videos, etc.) on the P2P network. This includes many popular P2P networks such as Gnutella, Napster, and BitTorrent, among others.
- Collaboration P2P networks include MSN or Skype chatting, instant messaging, and collaborative design, among others.
- The third family is for distributed P2P computing in specific applications. For example, SETI@home provides 25 Tflops of distributed computing power, collectively, over 3 million Internet host machines.
- Other P2P platforms, such as JXTA, .NET, and FightingAID@home, support naming, discovery, communication, security, and resource aggregation in some P2P applications.

P2P Computing Challenges

- P2P computing faces three types of heterogeneity problems in hardware, software, and network requirements.
- There are too many hardware models and architectures to select from; incompatibility exists between software and the OS; and different network connections and protocols make it too complex to apply in real applications.
- We need system scalability as the workload increases. System scaling is directly related to performance and bandwidth. P2P networks do have these properties.
- Data location is also important to affect collective performance. Data locality, network proximity, and interoperability are three design objectives in distributed P2P applications.

❖ Cloud Computing over the Internet

- In the future, working with large data sets will typically mean sending the computations (programs) to the data, rather than copying the data to the workstations.
- This reflects the trend in IT of moving computing and data from desktops to large data centers, where there is on-demand provision of software, hardware, and data as a service.
- This data explosion has promoted the idea of cloud computing.

Internet Clouds

- 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. Some computer users think of the cloud as a centralized resource pool.

- Others consider the cloud to be a server cluster which practices distributed computing over all the servers used.

The Cloud Landscape

- Traditionally, a distributed computing system tends to be owned and operated by an autonomous administrative domain (e.g., a research laboratory or company) for on-premises computing needs.
- However, these traditional systems have encountered several performance bottlenecks: constant system maintenance, poor utilization, and increasing costs associated with hardware/software upgrades.

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

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

(iii) **Software as a Service (SaaS)** This refers to browser-initiated application software over thousands of paid cloud customers. The SaaS model applies to business processes, industry applications, consumer relationship management (CRM), enterprise resources planning (ERP), human resources (HR), and collaborative applications. On the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are rather low, compared with conventional hosting of user applications.

- The following list highlights eight reasons to adapt the cloud for upgraded Internet applications and web services:

  (a) Desired location in areas with protected space and higher energy efficiency

  (b) Sharing of peak-load capacity among a large pool of users, improving overall utilization

  (c) Separation of infrastructure maintenance duties from domain-specific application development

  (d) Significant reduction in cloud computing cost, compared with traditional computing paradigms

  (e) Cloud computing programming and application development

  (f) Service and data discovery and content/service distribution

  (g) Privacy, security, copyright, and reliability issues

  (h) Service agreements, business models, and pricing policies

➢ SOFTWARE ENVIRONMENTS FOR DISTRIBUTED SYSTEMS AND CLOUDS

❖ Service-Oriented Architecture (SOA)

- In grids/web services, Java, and CORBA, an entity is, respectively, a service, a Java object, and a CORBA distributed object in a variety of languages.
- These architectures build on the traditional seven Open Systems Interconnection (OSI) layers that provide the base networking abstractions.
- On top of this we have a base software environment, which would be .NET or Apache Axis for web services, the Java Virtual Machine for Java, and a broker network for CORBA.
- On top of this base environment one would build a higher level environment reflecting the special features of the distributed computing environment.
- This starts with entity interfaces and inter-entity communication, which rebuild the top four OSI layers but at the entity and not the bit level.
- As shown in Figure 2.4, service-oriented architecture (SOA) has evolved over the years. SOA applies to building grids, clouds, grids of clouds, clouds of grids, clouds of clouds (also known as interclouds), and systems of systems in general.
- A large number of sensors provide data-collection services, denoted in the figure as SS (sensor service).
- A sensor can be a ZigBee device, a Bluetooth device, a WiFi access point, a personal computer, a GPA, or a wireless phone, among other things.
- Raw data is collected by sensor services. All the SS devices interact with large or small computers, many forms of grids, databases, the compute cloud, the storage cloud, the filter cloud, the discovery cloud, and so on.

- Filter services (fs in the figure) are used to eliminate unwanted raw data, in order to respond to specific requests from the web, the grid, or web services.
- A collection of filter services forms a filter cloud.

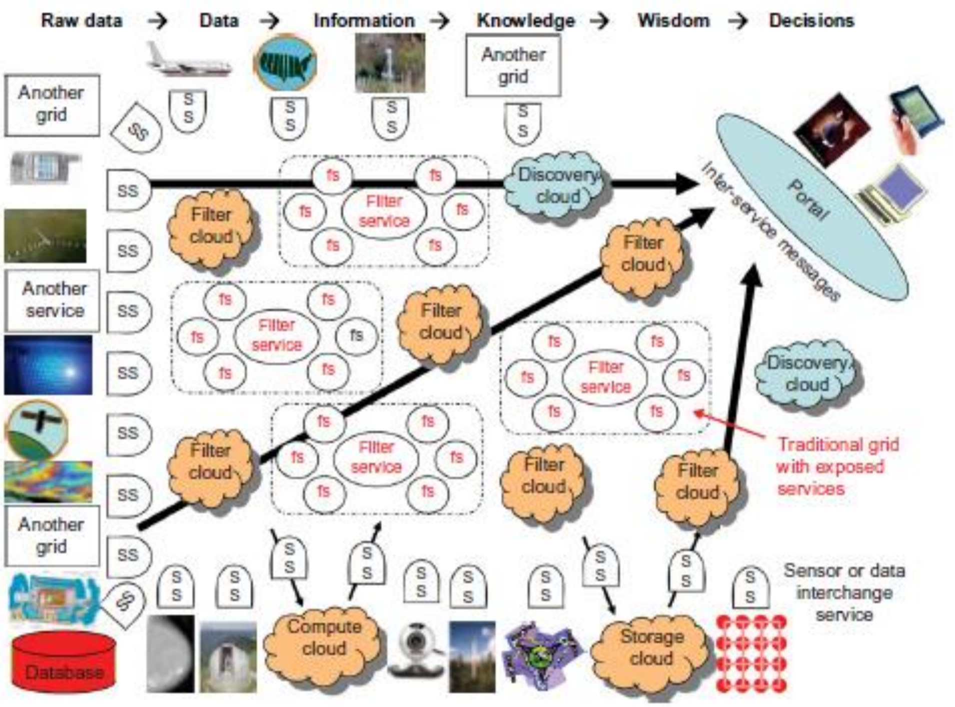


Figure 2.4 Evolution of SOA

➢ CLOUD COMPUTING AND SERVICE MODELS

- 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 Clouds

- A 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. The callout box in top of Figure 2.5 shows the architecture of a typical public cloud. 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.

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.

Hybrid Clouds

- A hybrid cloud is built with both public and private clouds, as shown at the lower-left corner of Figure 2.5.
- Private clouds can also support a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud.
- For example, the Research Compute Cloud (RC2) is a private cloud, built by IBM, that interconnects the computing and IT resources at eight IBM Research Centers scattered throughout the United States, 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.

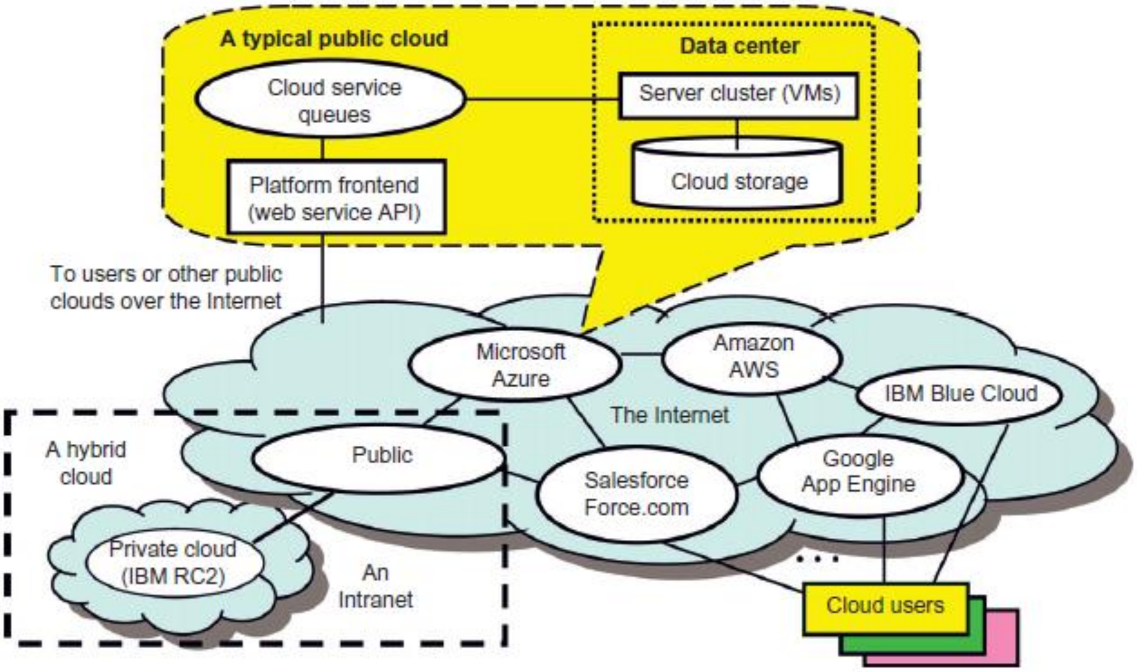


Figure 2.5 Public, private, and hybrid clouds

Cloud Design Objectives

- Despite the controversy surrounding the replacement of desktop or deskside computing by centralized computing and storage services at data centers or big IT companies, the cloud computing community has reached some consensus on what has to be done to make cloud computing universally acceptable.

- The following list highlights six design objectives for cloud computing:

a) **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.

b) **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.

c) **Scalability in performance** The cloud platforms and software and infrastructure services must be able to scale in performance as the number of users increases.

d) **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.

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

f) **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 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.
- An ecosystem was suggested by Sotomayor, (Figure 2.6) for building private clouds.
- They suggested four levels of ecosystem development in a private cloud.

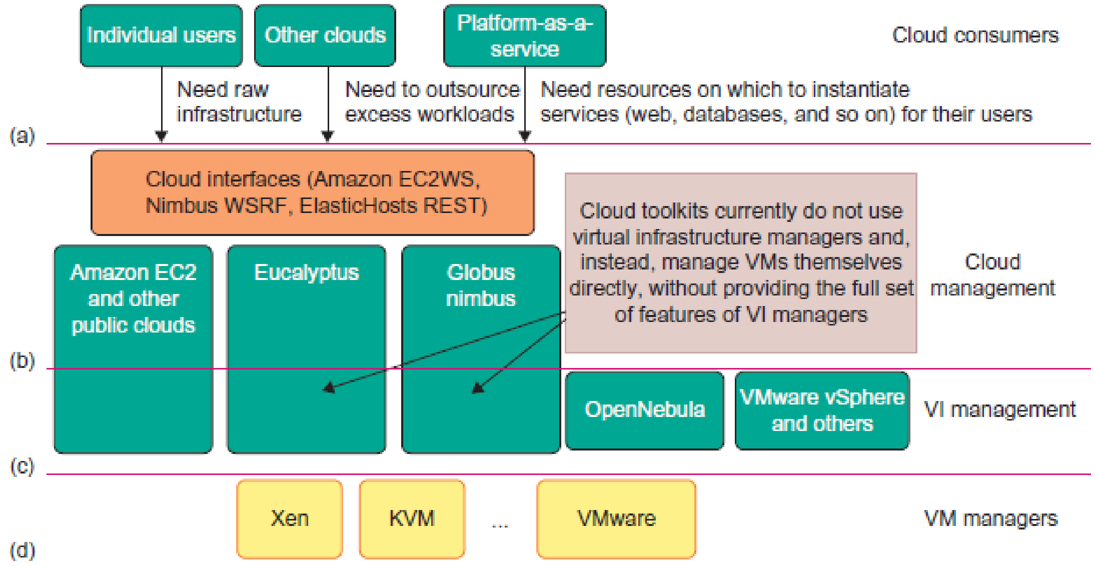

Individual users
Other clouds
Platform-as-a-service
Cloud consumers
Need raw infrastructure
Need to outsource excess workloads
Need resources on which to instantiate services (web, databases, and so on) for their users
(a)
Cloud interfaces (Amazon EC2WS, Nimbus WSRF, ElasticHosts REST)
Cloud toolkits currently do not use virtual infrastructure managers and, instead, manage VMs themselves directly, without providing the full set of features of VI managers
Amazon EC2 and other public clouds
Eucalyptus
Globus nimbus
Cloud management
(b)
OpenNebula
VMware vSphere and others
VI management
(c)
Xen
KVM
...
VMware
VM managers
(d)

Figure 2.6 Cloud Ecosystem

- 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.
- 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.

MODULE 3

CLOUD ARCHITECTURE AND RESOURCE MANAGEMENT

➢ ARCHITECTURAL DESIGN OF COMPUTE AND STORAGE CLOUDS

❖ A Generic Cloud Architecture Design

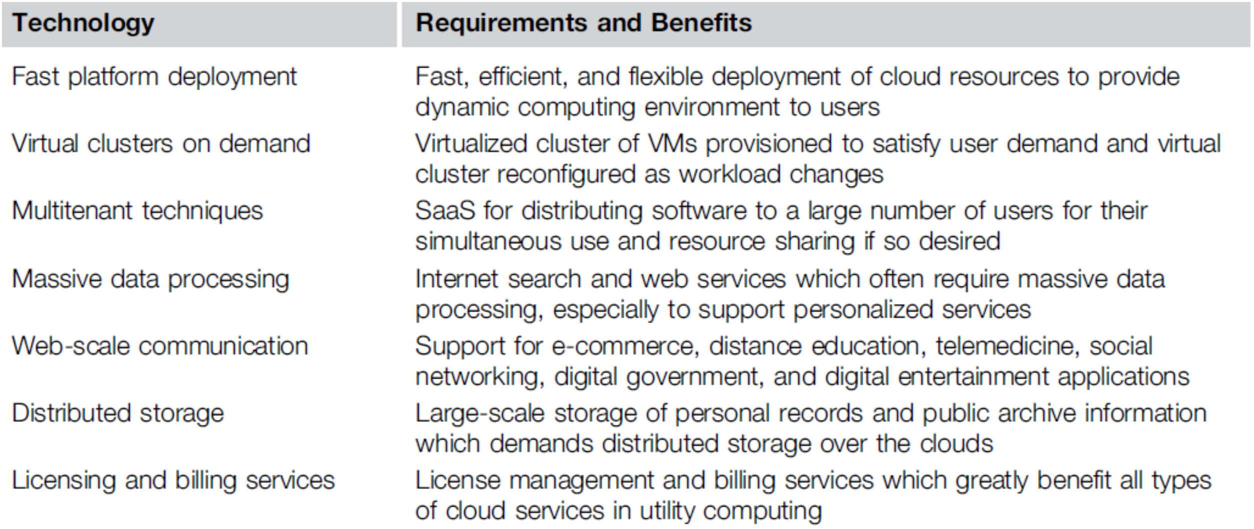
Cloud Platform Design Goals

- Scalability, virtualization, efficiency, and reliability are four major design goals of a cloud computing platform.
- Clouds support Web 2.0 applications. Cloud management receives the user request, finds the correct resources, and then calls the provisioning services which invoke the resources in the cloud.
- The cloud management software needs to support both physical and virtual machines.
- Security in shared resources and shared access of data centers also pose another design challenge.
- The platform needs to establish a very large-scale HPC infrastructure. The hardware and software systems are combined to make it easy and efficient to operate.
- System scalability can benefit from cluster architecture. If one service takes a lot of processing power, storage capacity, or network traffic, it is simple to add more servers and bandwidth.

- System reliability can benefit from this architecture. Data can be put into multiple locations.

Enabling Technologies for Clouds

- The key driving forces behind cloud computing are the ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software.
- Cloud users are able to demand more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers can increase system utilization via multiplexing, virtualization, and dynamic resource provisioning.
- Clouds are enabled by the progress in hardware, software, and networking technologies summarized in Table 3.1.



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

Table 3.1 Cloud-Enabling Technologies in Hardware, Software, and Networking

- These technologies play instrumental roles in making cloud computing a reality.
- Most of these technologies are mature today to meet increasing demand. In the hardware area, the rapid progress in multicore CPUs, memory chips, and disk arrays has made it possible to build faster data centers with huge amounts of storage space.
- Resource virtualization enables rapid cloud deployment and disaster recovery. Service-oriented architecture (SOA) also plays a vital role.

❖ Layered Cloud Architectural Development

- The architecture of a cloud is developed at three layers: infrastructure, platform, and application, as demonstrated in Figure 3.1.
- These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud.
- The services to public, private, and hybrid clouds are conveyed to users through networking support over the Internet and intranets involved.
- It is clear that the infrastructure layer is deployed first to support IaaS services. This infrastructure layer serves as the foundation for building the platform layer of the cloud for supporting PaaS services.
- In turn, the platform layer is a foundation for implementing the application layer for SaaS applications.

- Different types of cloud services demand application of these resources separately.

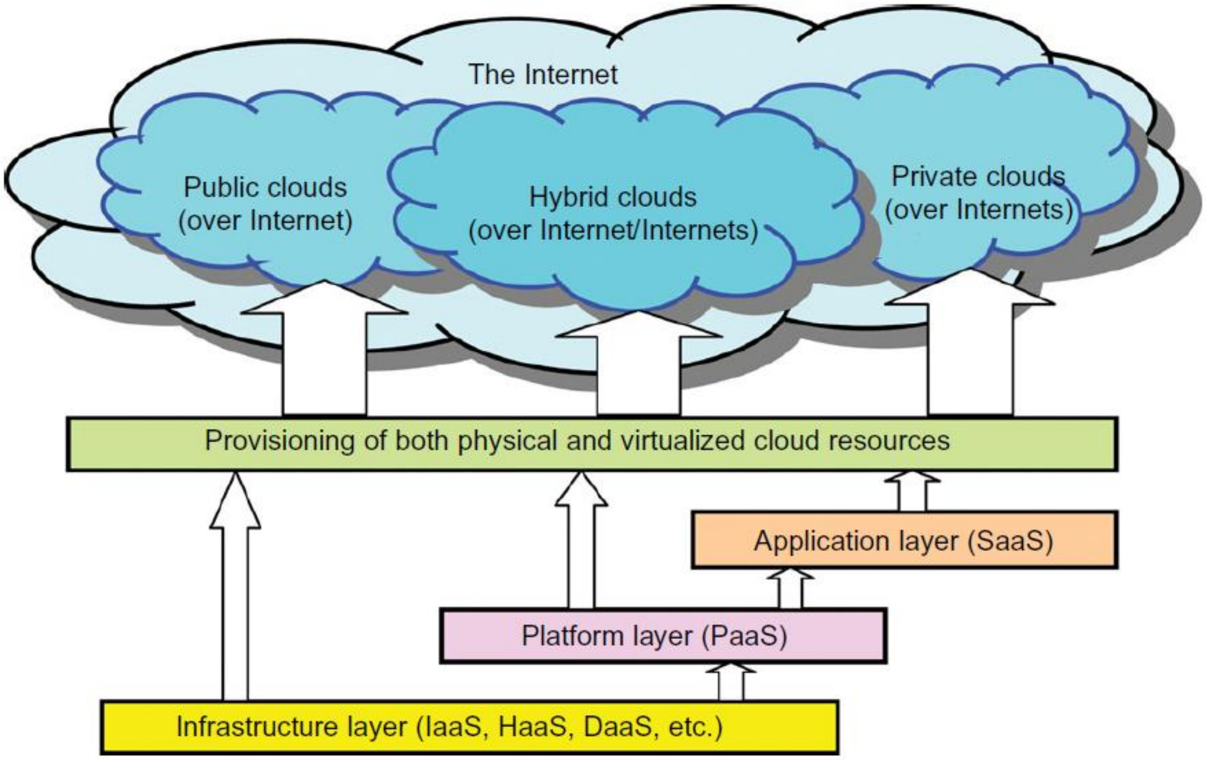

The Internet
Public clouds
(over Internet)
Hybrid clouds
(over Internet/Internets)
Private clouds
(over Internets)
Provisioning of both physical and virtualized cloud resources
Application layer (SaaS)
Platform layer (PaaS)
Infrastructure layer (IaaS, HaaS, DaaS, etc.)

Figure 4.1 Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

- The infrastructure layer is built with virtualized compute, storage, and network resources. The abstraction of these hardware resources is meant to provide the flexibility demanded by users.
- Internally, virtualization realizes automated provisioning of resources and optimizes the infrastructure management process.

- The platform layer is for general-purpose and repeated usage of the collection of software resources.
- This layer provides users with an environment to develop their applications, to test operation flows, and to monitor execution results and performance.
- The platform should be able to assure users that they have scalability, dependability, and security protection.
- In a way, the virtualized cloud platform serves as a “system middleware” between the infrastructure and application layers of the cloud.

Market-Oriented Cloud Architecture

- As consumers rely on cloud providers to meet more of their computing needs, they will require a specific level of QoS to be maintained by their providers, in order to meet their objectives and sustain their operations.
- Cloud providers consider and meet the different QoS parameters of each individual consumer as negotiated in specific SLAs.
- To achieve this, the providers cannot deploy traditional system-centric resource management architecture.
- Instead, market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand.
- The designer needs to provide feedback on economic incentives for both consumers and providers.

- The purpose is to promote QoS-based resource allocation mechanisms. In addition, clients can benefit from the potential cost reduction of providers, which could lead to a more competitive market, and thus lower prices.
- Figure 3.2 shows the high-level architecture for supporting market-oriented resource allocation in a cloud computing environment.

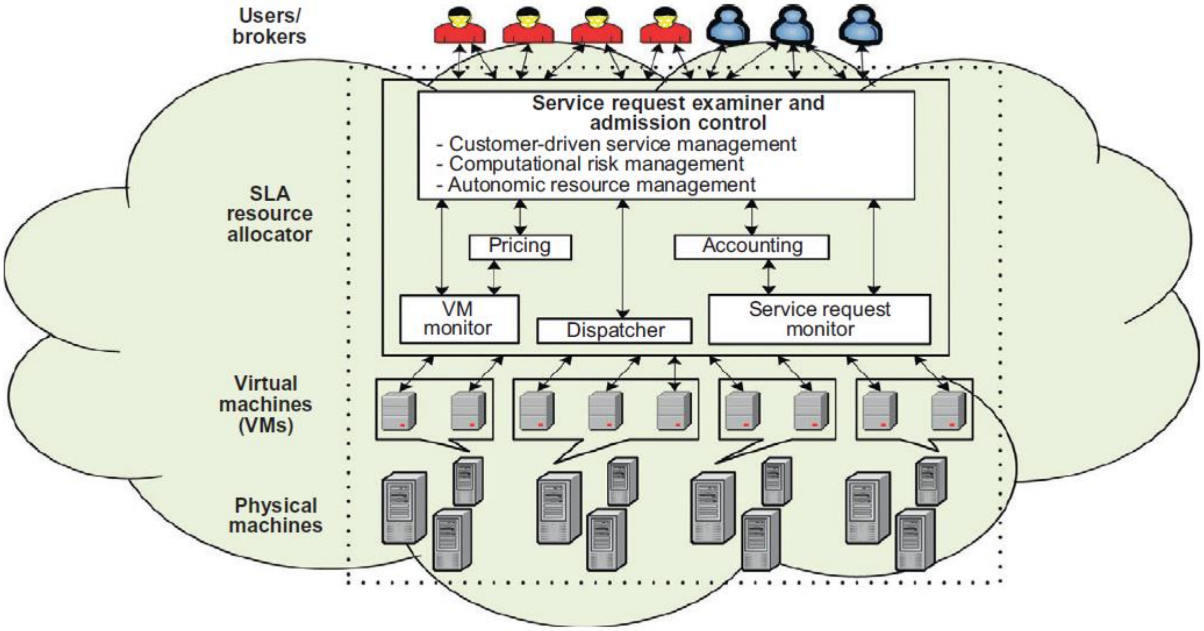

Users/ brokers
SLA resource allocator
Service request examiner and admission control
- Customer-driven service management
- Computational risk management
- Autonomic resource management
Pricing
Accounting
VM monitor
Dispatcher
Service request monitor
Virtual machines (VMs)
Physical machines

Figure 3.2 Market-Oriented Cloud Architecture

- This cloud is basically built with the following entities:
  a) Users or brokers acting on user's behalf submit service requests from anywhere in the world to the data center and cloud to be processed.
  b) The SLA resource allocator acts as the interface between the data center/cloud service provider and external users/brokers. It requires the interaction of the following mechanisms to support SLA-oriented resource management.

- When a service request is first submitted the service request examiner interprets the submitted request for QoS requirements before determining whether to accept or reject the request.
- The request examiner ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources.
- It also needs the latest status information regarding resource availability (from the VM Monitor mechanism) and workload processing (from the Service Request Monitor mechanism) in order to make resource allocation decisions effectively.
- Then it assigns requests to VMs and determines resource entitlements for allocated VMs.

c) The Pricing mechanism decides how service requests are charged. Pricing serves as a basis for managing the supply and demand of computing resources within the data center and facilitates in prioritizing resource allocations effectively. The Accounting mechanism maintains the actual usage of resources by requests so that the final cost can be computed and charged to users.

d) The VM Monitor mechanism keeps track of the availability of VMs and their resource entitlements. The Dispatcher mechanism starts the execution of accepted service requests on allocated VMs. The Service Request Monitor mechanism keeps track of the execution progress of service requests. Multiple VMs can be started and stopped on demand on a single physical machine to meet accepted service requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests.

- **Virtualization Support and Disaster Recovery**

- One very distinguishing feature of cloud computing infrastructure is the use of system virtualization and the modification to provisioning tools.
- Virtualization of servers on a shared cluster can consolidate web services.
- As the VMs are the containers of cloud services, the provisioning tools will first find the corresponding physical machines and deploy the VMs to those nodes before scheduling the service to run on the virtual nodes.
- In addition, in cloud computing, virtualization also means the resources and fundamental infrastructure are virtualized.
- The user will not care about the computing resources that are used for providing the services. Cloud users do not need to know and have no way to discover physical resources that are involved while processing a service request.
- Also, application developers do not care about some infrastructure issues such as scalability and fault tolerance (i.e., they are virtualized). Application developers focus on service logic.

Hardware Virtualization

- In many cloud computing systems, virtualization software is used to virtualize the hardware.

- System virtualization software is a special kind of software which simulates the execution of hardware and runs even unmodified operating systems.
- Cloud computing systems use virtualization software as the running environment for legacy software such as old operating systems and unusual applications.
- Virtualization software is also used as the platform for developing new cloud applications that enable developers to use any operating systems and programming environments they like.
- The development environment and deployment environment can now be the same, which eliminates some runtime problems.
- Some cloud computing providers have used virtualization technology to provide this service for developers.
- System virtualization software is considered the hardware analog mechanism to run an unmodified operating system, usually on bare hardware directly, on top of software.

Virtualization Support in Public Clouds

- Three public clouds in the context of virtualization support: AWS, Microsoft Azure, and GAE. AWS provides extreme flexibility (VMs) for users to execute their own applications.
- GAE provides limited application-level virtualization for users to build applications only based on the services that are created by Google.
- Microsoft provides programming-level virtualization (.NET virtualization) for users to build their applications.

- The VMware tools apply to workstations, servers, and virtual infrastructure. The Microsoft tools are used on PCs and some special servers.
- Virtualization leads to HA, disaster recovery, dynamic load leveling, and rich provisioning support. Both cloud computing and utility computing leverage the benefits of virtualization to provide a scalable and autonomous computing environment.

Storage Virtualization for Green Data Centers

- IT power consumption in the United States has more than doubled to 3 percent of the total energy consumed in the country. The large number of data centers in the country has contributed to this energy crisis to a great extent.
- More than half of the companies in the Fortune 500 are actively implementing new corporate energy policies. Recent surveys from both IDC and Gartner confirm the fact that virtualization had a great impact on cost reduction from reduced power consumption in physical computing systems.
- This alarming situation has made the IT industry become more energy-aware. With little evolution of alternate energy resources, there is an imminent need to conserve power in all computers
- . Virtualization and server consolidation have already proven handy in this aspect. Green data centers and benefits of storage virtualization are considered to further strengthen the synergy of green computing.

Virtualization for IaaS

- VM technology has increased in ubiquity. This has enabled users to create customized environments atop physical infrastructure for cloud computing.

- Use of VMs in clouds has the following distinct benefits:

  (1) System administrators consolidate workloads of underutilized servers in fewer servers;

  (2) VMs have the ability to run legacy code without interfering with other APIs;

  (3) VMs can be used to improve security through creation of sandboxes for running applications with questionable reliability; And

  (4) virtualized cloud platforms can apply performance isolation, letting providers offer some guarantees and better QoS to customer applications.

VM Cloning for Disaster Recovery

- VM technology requires an advanced disaster recovery scheme. One scheme is to recover one physical machine by another physical machine.
- The second scheme is to recover one VM by another VM. Traditional disaster recovery from one physical machine to another is rather slow, complex, and expensive.
- Total recovery time is attributed to the hardware configuration, installing and configuring the OS, installing the backup agents, and the long time to restart the physical machine.
- To recover a VM platform, the installation and configuration times for the OS and backup agents are eliminated. Therefore, we end up with a much shorter disaster recovery time, about 40 percent of that to recover the physical machines. Virtualization aids in fast disaster recovery by VM encapsulation.

- **Architectural Design Challenges**

Challenge 1—Service Availability and Data Lock-in Problem

- The management of a cloud service by a single company is often the source of single points of failure.
- To achieve HA, one can consider using multiple cloud providers. Even if a company has multiple data centers located in different geographic regions, it may have common software infrastructure and accounting systems.
- Therefore, using multiple cloud providers may provide more protection from failures.
- Another availability obstacle is distributed denial of service (DDoS) attacks. Criminals threaten to cut off the incomes of SaaS providers by making their services unavailable.
- Some utility computing services offer SaaS providers the opportunity to defend against DDoS attacks by using quick scale-ups.
- Software stacks have improved interoperability among different cloud platforms, but the APIs itself are still proprietary.
- Thus, customers cannot easily extract their data and programs from one site to run on another.
- The obvious solution is to standardize the APIs so that a SaaS developer can deploy services and data across multiple cloud providers. This will rescue the loss of all data due to the failure of a single company.

Challenge 2—Data Privacy and Security Concerns

- Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks.
- Many obstacles can be overcome immediately with well-understood technologies such as encrypted storage, virtual LANs, and network middleboxes (e.g., firewalls, packet filters).
- Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms. In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits.
- Another type of attack is the man-in-the-middle attack for VM migrations.
- In general, passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

Challenge 3—Unpredictable Performance and Bottlenecks

- Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic. Internet applications continue to become more data-intensive.
- If we assume applications to be "pulled apart" across the boundaries of clouds, this may complicate data placement and transport.
- Cloud users and providers have to think about the implications of placement and traffic at every level of the system, if they want to minimize costs.

- This kind of reasoning can be seen in Amazon’s development of its new CloudFront service.
- Therefore, data transfer bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed.

Challenge 4—Distributed Storage and Widespread Software Bugs

- The database is always growing in cloud applications. The opportunity is to create a storage system that will not only meet this growth, but also combine it with the cloud advantage of scaling arbitrarily up and down on demand.
- This demands the design of efficient distributed SANs. Data centers must meet programmers’ expectations in terms of scalability, data durability, and HA.
- Data consistence checking in SAN-connected data centers is a major challenge in cloud computing.
- Large-scale distributed bugs cannot be reproduced, so the debugging must occur at a scale in the production data centers. No data center will provide such a convenience.
- One solution may be a reliance on using VMs in cloud computing. The level of virtualization may make it possible to capture valuable information in ways that are impossible without using VMs.
- Debugging over simulators is another approach to attacking the problem, if the simulator is well designed.

Challenge 5—Cloud Scalability, Interoperability, and Standardization

- The pay-as-you-go model applies to storage and network bandwidth; both are counted in terms of the number of bytes used.
- Computation is different depending on virtualization level. GAE automatically scales in response to load increases and decreases; users are charged by the cycles used.
- AWS charges by the hour for the number of VM instances used, even if the machine is idle. The opportunity here is to scale quickly up and down in response to load variation, in order to save money, but without violating SLAs.
- Open Virtualization Format (OVF) describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of VMs. It also defines a format for distributing software to be deployed in VMs. This VM format does not rely on the use of a specific host platform, virtualization platform, or guest operating system.
- The approach is to address virtual platform-agnostic packaging with certification and integrity of packaged software.
- OVF also defines a transport mechanism for VM templates, and can apply to different virtualization platforms with different levels of virtualization.

Challenge 6—Software Licensing and Reputation Sharing

- Many cloud computing providers originally relied on open source software because the licensing model for commercial software is not ideal for utility computing.

- The primary opportunity is either for open source to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing.
- One can consider using both pay-for-use and bulk-use licensing schemes to widen the business coverage.
- Another legal issue concerns the transfer of legal liability. Cloud providers want legal liability to remain with the customer, and vice versa. This problem must be solved at the SLA level.

➢ 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.
- Figure 3.3 introduces five levels of cloud players. 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.
- Cloud services rely on new advances in machine virtualization, SOA, grid infrastructure management, and power efficiency.

- 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.

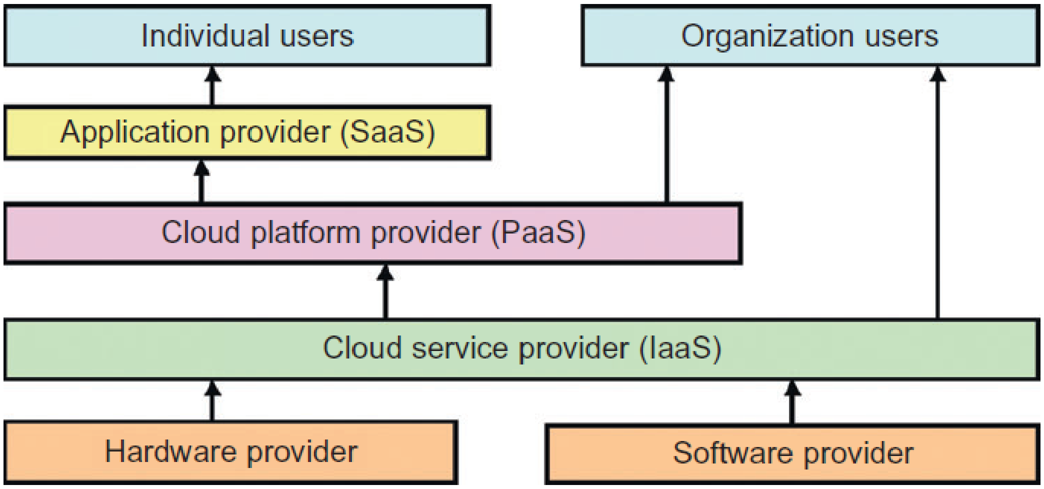

Individual users
Organization users
Application provider (SaaS)
Cloud platform provider (PaaS)
Cloud service provider (IaaS)
Hardware provider
Software provider

Figure 3.3 Five Levels of Cloud Players

❖ 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.
- Google has hundreds of data centers and has installed more than 460,000 servers worldwide.

.

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.
- In 2008, Google announced the GAE web application platform which is becoming a common platform for many small cloud service providers. This platform specializes in supporting scalable (elastic) web applications.
- GAE enables users to run their applications on a large number of data centers associated with Google's search engine operations.

GAE Architecture

- Figure 3.4 shows the major building blocks of the Google cloud platform which has been used to deliver the cloud services highlighted earlier.
- 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.

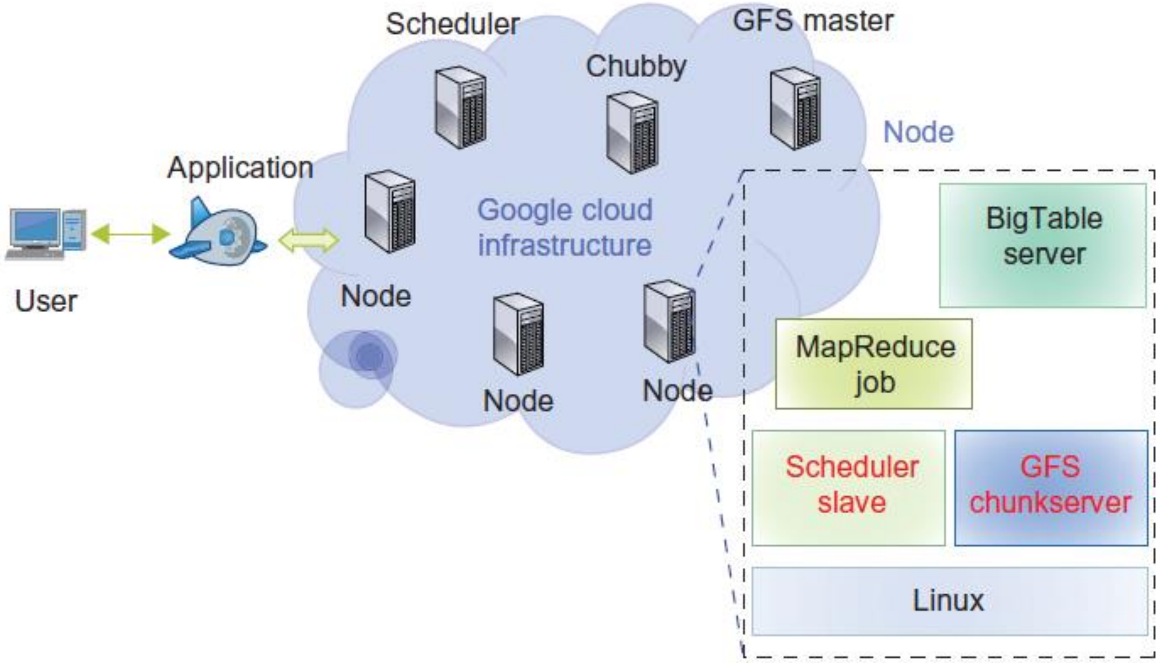


Figure 3.4 Building blocks of Google Cloud Platform

- 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 is one of the larger cloud application providers, although its fundamental service program is private and outside people cannot use the Google infrastructure to build their own service.
- The building blocks of Google's cloud computing application include the Google File System for storing large amounts of data, the MapReduce programming framework for application developers, Chubby for distributed application lock services, and BigTable as a storage service for accessing structural or semi-structural data.

Functional Modules of GAE

(a) The data store offers object-oriented, distributed, structured data storage services based on BigTable techniques. The data store secures data management operations.

(b) The application runtime environment offers a platform for scalable web programming and execution. It supports two development languages: Python and Java.

(c) 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.

(d) The administration console is used for easy management of user application development cycles, instead of for physical resource management.

(e) The GAE web service infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources by 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. Inside each data center, there might be thousands of server nodes to form different clusters.
- Each cluster can run multipurpose servers. GAE supports many web applications. One is a storage service to store application-specific data in the Google infrastructure.
- The data can be persistently stored in the backend storage server while still providing the facility for queries, sorting, and even transactions similar to traditional database systems.
- GAE also provides Google-specific services, such as the Gmail account service (which is the login service, that is, applications can use the Gmail account directly).

  This can eliminate the tedious work of building customized user management components in web applications. Thus, 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 3.5 shows the AWS architecture. 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.
- ELB automatically distributes incoming application traffic across multiple Amazon EC2 instances and allows user to avoid non operating nodes and to equalize load on functioning images.
- Both auto scaling and ELB are enabled by CloudWatch which monitors running instances.
- CloudWatch is a web service that provides monitoring for AWS cloud resources, starting with Amazon EC2.

- It provides customers with visibility into resource utilization, operational performance, and overall demand patterns, including metrics such as CPU utilization, disk reads and writes, and network traffic.
- Amazon (like Azure) offers a Relational Database Service (RDS) with a messaging interface. The Elastic MapReduce capability is equivalent to Hadoop running on the basic EC2 offering.
- AWS Import/Export allows one to ship large volumes of data to and from EC2 by shipping physical disks; it is well known that this is often the highest bandwidth connection between geographically distant systems.

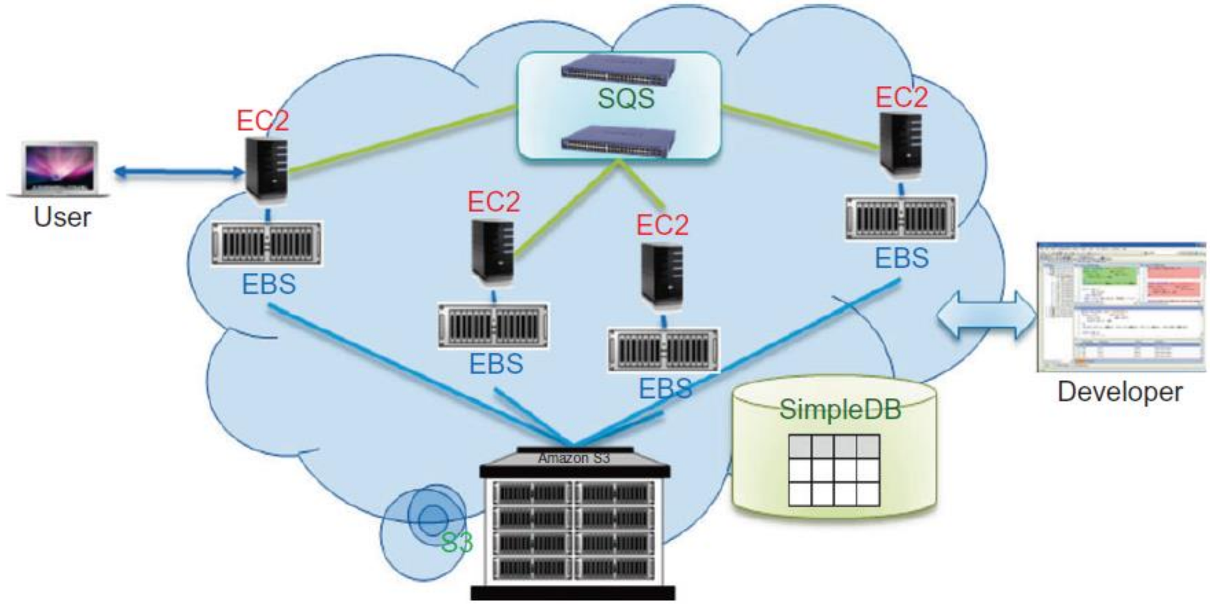


Figure 3.5 Amazon Cloud Architecture

- ELB automatically distributes incoming application traffic across multiple Amazon EC2 instances and allows user to avoid non operating nodes and to equalize load on functioning images.

- Both autoscaling and ELB are enabled by CloudWatch which monitors running instances. CloudWatch is a web service that provides monitoring for AWS cloud resources, starting with Amazon EC2.
- It provides customers with visibility into resource utilization, operational performance, and overall demand patterns, including metrics such as CPU utilization, disk reads and writes, and network traffic.
- Amazon (like Azure) offers a Relational Database Service (RDS) with a messaging interface.
- The Elastic MapReduce capability is equivalent to Hadoop running on the basic EC2 offering.
- AWS Import/Export allows one to ship large volumes of data to and from EC2 by shipping physical disks; it is well known that this is often the highest bandwidth connection between geographically distant systems.
- FPS provides developers of commercial systems on AWS with a convenient way to charge Amazon's customers that use such services built on AWS.
- Customers can pay using the same login credentials, shipping address, and payment information they already have on file with Amazon.
- The FWS allows merchants to access Amazon's fulfillment capabilities through a simple web service interface.

❖ 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.

- Figure 3.6 shows the overall architecture of Microsoft's cloud platform. 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.

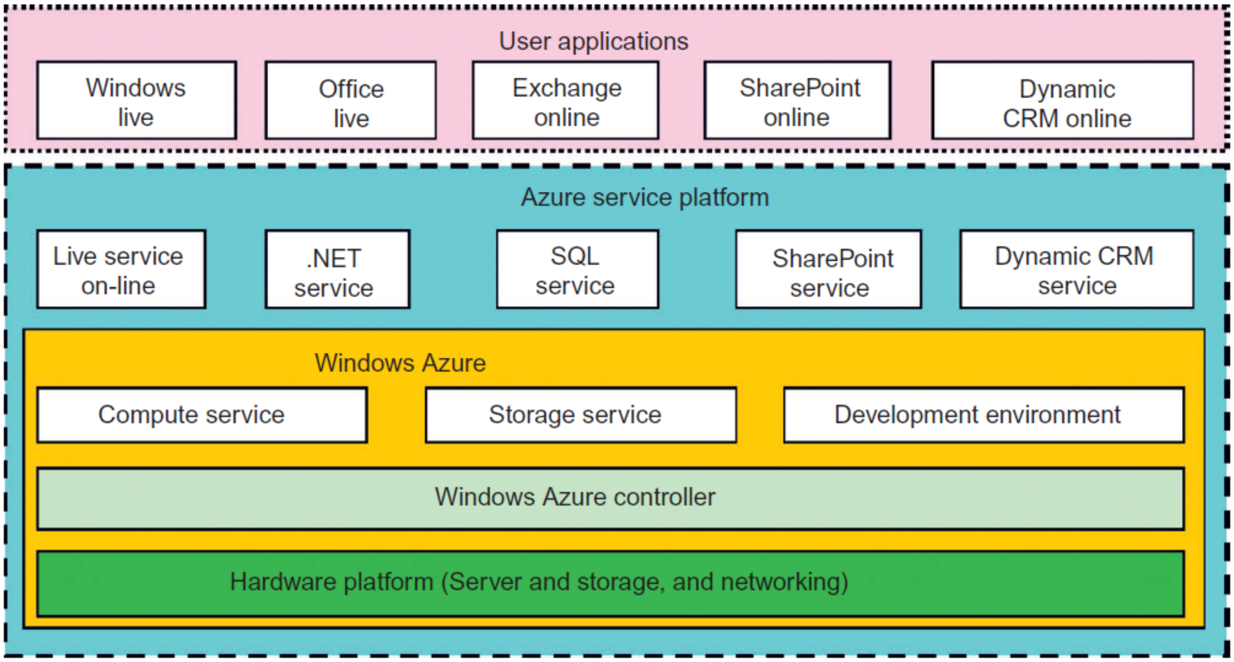


Figure 3.6 Microsoft Windows Azure platform for cloud computing.

- On top of the infrastructure are the various services for building different cloud applications. Cloud-level services provided by the Azure platform are introduced below.

  (a) **Live service** Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently.

(b) **.NET service** This package supports application development on local hosts and execution on cloud machines.

(c) **SQL Azure** This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud.

(d) **SharePoint service** This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services.

(e) **Dynamic CRM service** This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions.

- All these cloud services in Azure can interact with traditional Microsoft software applications, such as Windows Live, Office Live, Exchange online, SharePoint online, and dynamic CRM online.
- The Azure platform applies the standard web communication protocols SOAP and REST.
- The Azure service applications allow users to integrate the cloud application with other platforms or third-party clouds.

➢ EMERGING CLOUD SOFTWARE ENVIRONMENTS

❖ Open Source Eucalyptus and Nimbus

- Eucalyptus is a product from Eucalyptus Systems (www.eucalyptus.com) that was developed out of

- a research project at the University of California, Santa Barbara. Eucalyptus was initially aimed at bringing the cloud computing paradigm to academic supercomputers and clusters.
- Eucalyptus provides an AWS-compliant EC2-based web service interface for interacting with the cloud service.
- Additionally, Eucalyptus provides services, such as the AWS-compliant Walrus, and a user interface for managing users and images.

Eucalyptus Architecture

- The Eucalyptus system is an open software environment
- Figure 3.7 shows the architecture based on the need to manage VM images. The system supports cloud programmers in VM image management as follows.
- Essentially, the system has been extended to support the development of both the computer cloud and storage cloud.

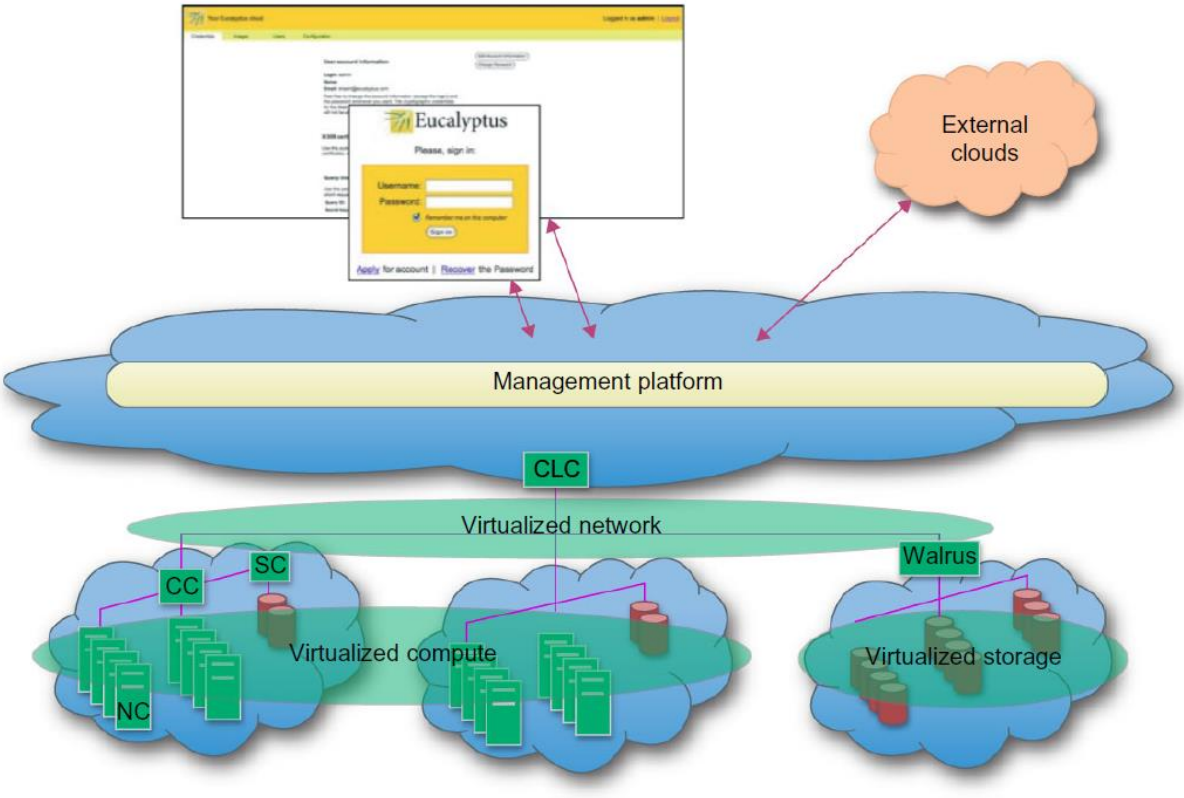


Figure 3.7 The Eucalyptus architecture for VM image management.

❖ Nimbus

- Nimbus is a set of open source tools that together provide an IaaS cloud computing solution.
- Figure 3.8 shows the architecture of Nimbus, which allows a client to lease remote resources by deploying VMs on those resources and configuring them to represent the environment desired by the user.
- To this end, Nimbus provides a special web interface known as Nimbus Web. Its aim is to provide administrative and user functions in a friendly interface.

- Nimbus Web is centered around a Python Django web application that is intended to be deployable completely separate from the Nimbus service.

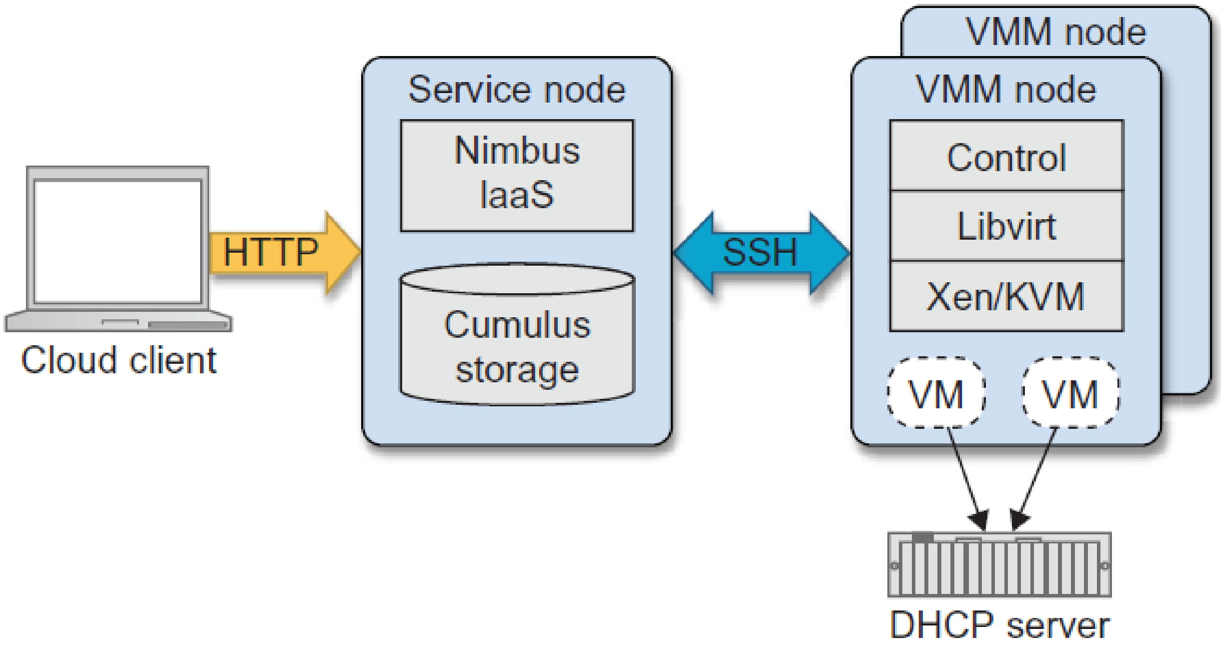


Figure 3.8 Nimbus Architecture

❖ OpenNebula, Sector/Sphere, and OpenStack

- OpenNebula is an open source toolkit which allows users to transform existing infrastructure into an IaaS cloud with cloud-like interfaces.
- Figure 3.9 shows the OpenNebula architecture and its main components.
- The architecture of OpenNebula has been designed to be flexible and modular to allow integration with different storage and network infrastructure configurations, and hypervisor technologies.

- Here, the core is a centralized component that manages the VM full life cycle, including setting up networks dynamically for groups of VMs and managing their storage requirements, such as VM disk image deployment or on-the-fly software environment creation.
- Another important component is the capacity manager or scheduler. It governs the functionality provided by the core.
- The default capacity scheduler is a requirement/rank matchmaker. However, it is also possible to develop more complex scheduling policies, through a lease model and advance reservations.
- The last main components are the access drivers. They provide an abstraction of the underlying infrastructure to expose the basic functionality of the monitoring, storage, and virtualization services available in the cluster.
- Therefore, OpenNebula is not tied to any specific environment and can provide a uniform management layer regardless of the virtualization platform.
- Additionally, OpenNebula offers management interfaces to integrate the core’s functionality within other data-center management tools, such as accounting or monitoring frameworks.

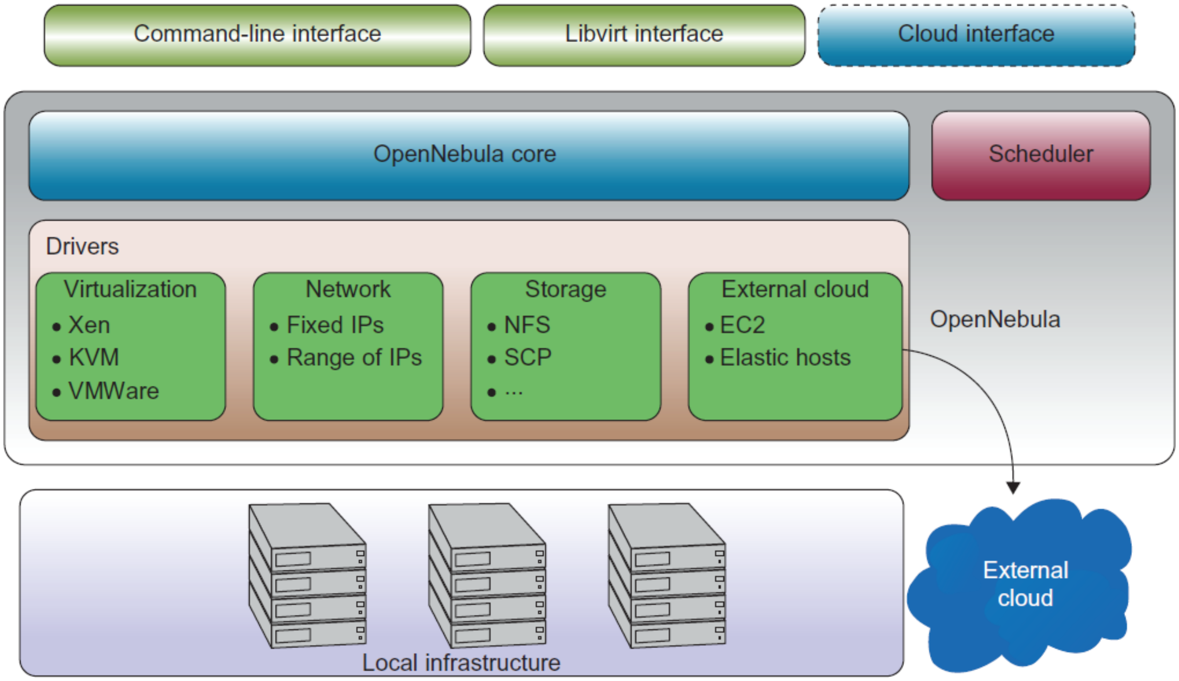


Figure 3.9 OpenNebula architecture and its main components.

- OpenNebula is able to adapt to organizations with changing resource needs, including addition or failure of physical resources.
- Some essential features to support changing environments are live migration and VM snapshots. Furthermore, when the local resources are insufficient, OpenNebula can support a hybrid cloud model by using cloud drivers to interface with external clouds.

❖ **Sector/Sphere**

- Sector/Sphere is a software platform that supports very large distributed data storage and simplified distributed data processing over large clusters of commodity computers, either within a data center or across multiple data centers.

- The system consists of the Sector distributed file system and the Sphere parallel data processing framework.
- Sector is a distributed file system (DFS) that can be deployed over a wide area and allows users to manage large data sets from any location with a highspeed network connection.
- The fault tolerance is implemented by replicating data in the file system and managing the replicas.
- Since Sector is aware of the network topology when it places replicas, it also provides better reliability, availability, and access throughout.
- The communication is performed using User Datagram Protocol (UDP) for message passing and user-defined type (UDT) for data transfer.
- On the other hand, Sphere is a parallel data processing engine designed to work with data managed by Sector.
- This coupling allows the system to make accurate decisions about job scheduling and data location. Sphere provides a programming framework that developers can use to process data stored in Sector. Thus, it allows UDFs to run on all input data segments in parallel.
- Such data segments are processed at their storage locations whenever possible (data locality). Failed data segments may be restarted on other nodes to achieve fault tolerance. In a Sphere application, both inputs and outputs are Sector files.

- Multiple Sphere processing segments can be combined to support more complicated applications, with inputs/outputs exchanged/shared via the Sector file system.
- The Sector/Sphere platform is supported by the architecture shown in Figure 3.10, which is composed of four components.

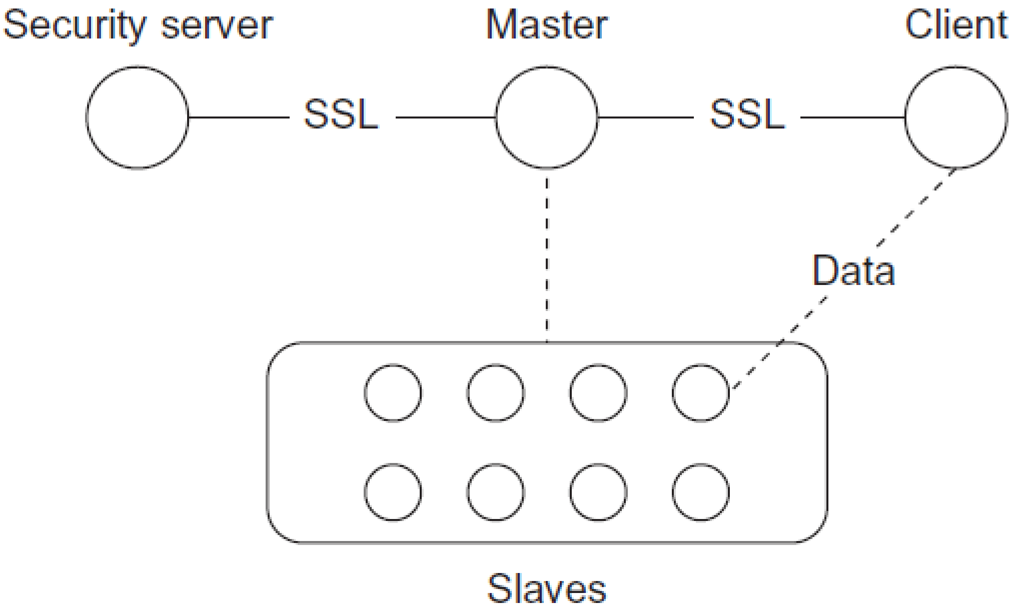


Figure 3.10 The Sector/Sphere system architecture.

- The first component is the security server, which is responsible for authenticating master servers, slave nodes, and users. We also have the master servers that can be considered the infrastructure core.
- The master server maintains file system metadata, schedules jobs, and responds to users' requests. Sector supports multiple active masters that can join and leave at runtime and can manage the requests.

- Another component is the slave nodes, where data is stored and processed. The slave nodes can be located within a single data center or across multiple
- data centers with high-speed network connections.
- The last component is the client component. This provides tools and programming APIs for accessing and processing Sector data.
- Finally, it is worthy to mention that as part of this platform, a new component has been developed. It is called Space and it consists of a framework to support column-based distributed data tables.
- Therefore, tables are stored by columns and are segmented onto multiple slave nodes. Tables are independent and no relationships between them are supported.

❖ **OpenStack**

- OpenStack was been introduced by Rackspace and NASA in July 2010. The project is building an open source community spanning technologists, developers, researchers, and industry to share resources and technologies with the goal of creating a massively scalable and secure cloud infrastructure.
- In the tradition of other open source projects, the software is open source and limited to just open source APIs such as Amazon.
- Currently, OpenStack focuses on the development of two aspects of cloud computing to address compute and storage aspects with the OpenStack Compute and OpenStack Storage solutions.

- “OpenStack Compute is the internal fabric of the cloud creating and managing large groups of virtual private servers” and “OpenStack Object Storage is software for creating redundant, scalable object storage using clusters of commodity servers to store terabytes or even petabytes of data.”

OpenStack Compute

- As part of its computing support efforts, OpenStack is developing a cloud computing fabric controller, a component of an IaaS system, known as Nova.
- The architecture for Nova is built on the concepts of shared-nothing and messaging-based information exchange. Hence, most communication in Nova is facilitated by message queues.
- To prevent blocking components while waiting for a response from others, deferred objects are introduced. Such objects include callbacks that get triggered when a response is received.
- This is very similar to established concepts from parallel computing, such as “futures,” which have been used in the grid community by projects such as the CoG Kit.
- To achieve the shared-nothing paradigm, the overall system state is kept in a distributed data system. State updates are made consistent through atomic transactions.
- Nova it implemented in Python while utilizing a number of externally supported libraries and components. This includes boto, an Amazon API provided in Python, and Tornado, a fast HTTP server used to implement the S3 capabilities in OpenStack. Figure 3.11 shows the main architecture of Open Stack Compute.

- The cloud controller maintains the global state of the system, ensures authorization while interacting with the User Manager via Lightweight Directory Access Protocol (LDAP), interacts with the S3 service, and manages nodes, as well as storage workers through a queue.

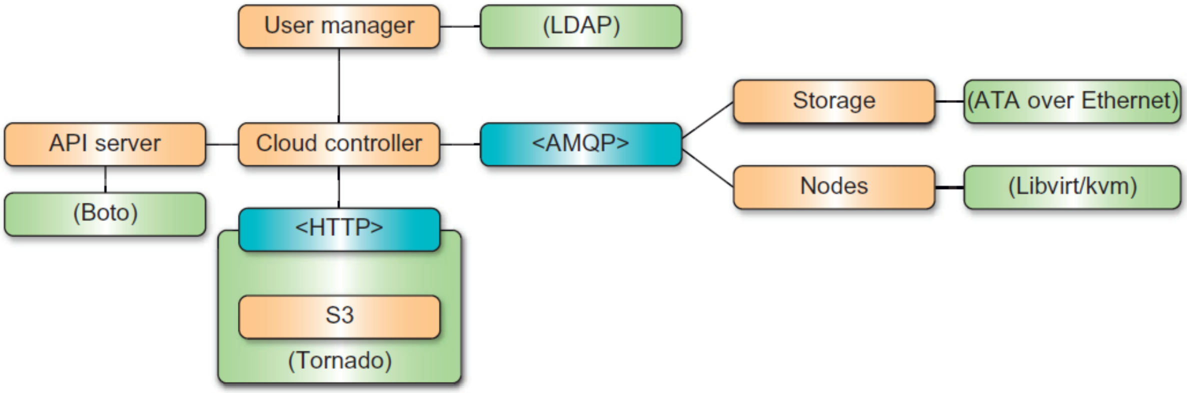


Figure 3.11 OpenStack Nova system architecture

- Additionally, Nova integrates networking components to manage private networks, public IP addressing, virtual private network (VPN) connectivity, and firewall rules. It includes the following types:
  - NetworkController manages address and virtual LAN (VLAN) allocations
  - RoutingNode governs the NAT (network address translation) conversion of public IPs to private IPs, and enforces firewall rules
  - AddressingNode runs Dynamic Host Configuration Protocol (DHCP) services for private networks
  - TunnelingNode provides VPN connectivity
- The network state (managed in the distributed object store) consists of the following:

- VLAN assignment to a project
- Private subnet assignment to a security group in a VLAN
- Private IP assignments to running instances
- Public IP allocations to a project
- Public IP associations to a private IP/running instance

OpenStack Storage

- The OpenStack storage solution is built around a number of interacting components and concepts, including a proxy server, a ring, an object server, a container server, an account server, replication, updaters, and auditors.
- The role of the proxy server is to enable lookups to the accounts, containers, or objects in OpenStack storage rings and route the requests.
- Thus, any object is streamed to or from an object server directly through the proxy server to or from the user.
- A ring represents a mapping between the names of entities stored on disk and their physical locations.
- Separate rings for accounts, containers, and objects exist. A ring includes the concept of using zones, devices, partitions, and replicas. Hence, it allows the system to deal with failures, and isolation of zones representing a drive, a server, a cabinet, a switch, or even a data center.
- Weights can be used to balance the distribution of partitions on drives across the cluster, allowing users to support heterogeneous storage resources.

- ❖ **Manjrasoft Aneka Cloud and Appliances**

- Aneka (www.manjrasoft.com/) is a cloud application platform developed by Manjrasoft, based in Melbourne, Australia.
- It is designed to support rapid development and deployment of parallel and distributed applications on private or public clouds.
- It provides a rich set of APIs for transparently exploiting distributed resources and expressing the business logic of applications by using preferred programming abstractions.
- System administrators can leverage a collection of tools to monitor and control the deployed infrastructure.
- It can be deployed on a public cloud such as Amazon EC2 accessible through the Internet to its subscribers, or a private cloud constituted by a set of nodes with restricted access as shown in Figure 3.12.
- Aneka acts as a workload distribution and management platform for accelerating applications in both Linux and Microsoft .NET framework environments.
- Some of the key advantages of Aneka over other workload distribution solutions include:
    a) Support of multiple programming and application environments
    b) Simultaneous support of multiple runtime environments
    c) Rapid deployment tools and framework
    d) Ability to harness multiple virtual and/or physical machines for accelerating application provisioning based on users' Quality of Service/service-level agreement (QoS/SLA) requirements

e) Built on top of the Microsoft .NET framework, with support for Linux environments through Mono

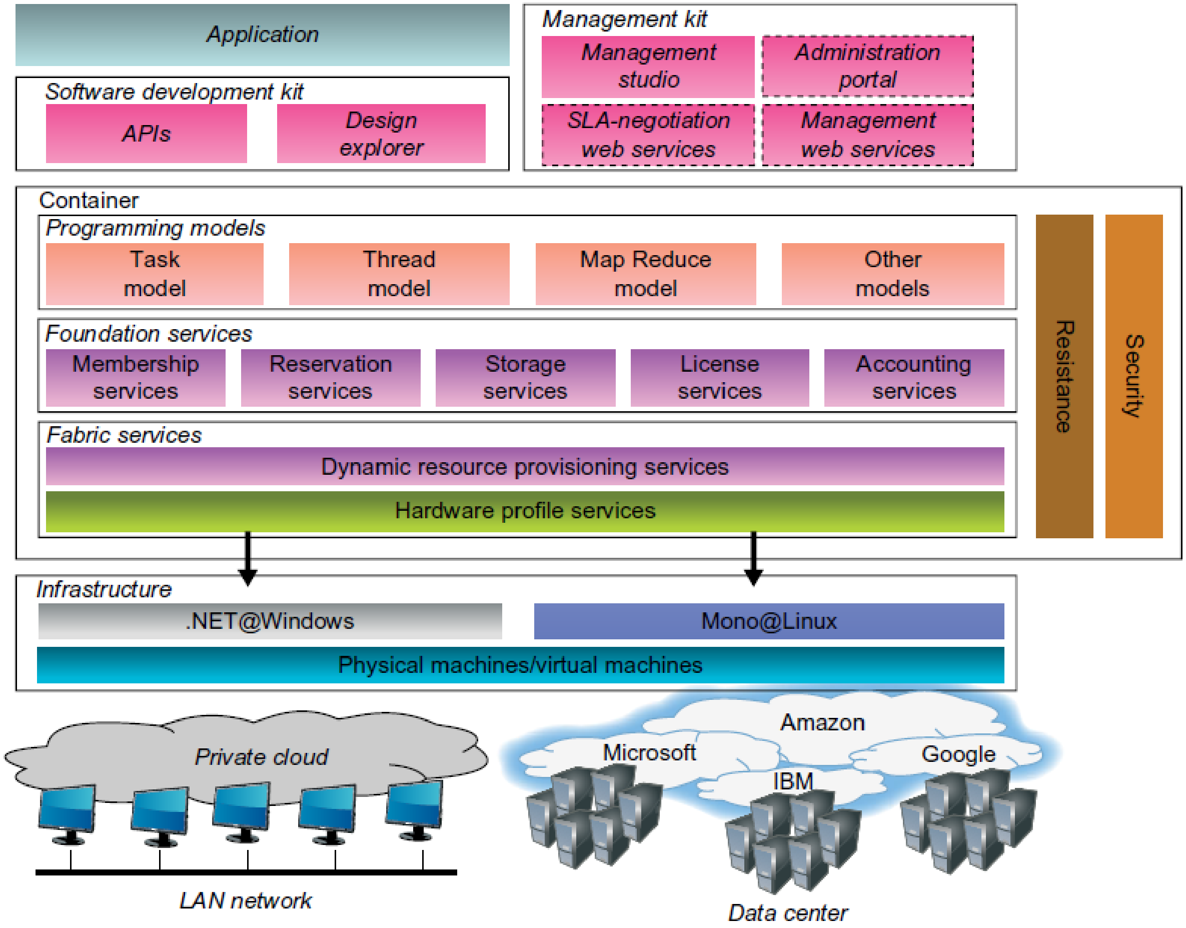


Figure 3.12 Architecture and components of Aneka.

- Aneka offers three types of capabilities which are essential for building, accelerating, and managing clouds and their applications:

1. **Build** Aneka includes a new SDK which combines APIs and tools to enable users to rapidly develop applications. Aneka also allows users to build different runtime environments such as enterprise/private cloud by harnessing compute resources in network or enterprise data centers,   Amazon EC2, and hybrid clouds by combining enterprise private clouds managed by Aneka with resources from Amazon EC2 or other enterprise clouds built and managed using XenServer
2. **Accelerate** Aneka supports rapid development and deployment of applications in multiple runtime environments running different operating systems such as Windows or Linux/UNIX. Aneka uses physical machines as much as possible to achieve maximum utilization in local environments.
3. **Manage** Management tools and capabilities supported by Aneka include a GUI and APIs to set up, monitor, manage, and maintain remote and global Aneka compute clouds. Aneka also has an accounting mechanism and manages priorities and scalability based on SLA/QoS which enables dynamic provisioning.

- Here are three important programming models supported by Aneka for both cloud and traditional parallel applications:
  1. Thread programming model, best solution to adopt for leveraging the computing capabilities of multicore nodes in a cloud of computers
  2. Task programming model, which allows for quickly prototyping and implementing an independent bag of task applications
  3. MapReduce programming model

Aneka Architecture

- Aneka as a cloud application platform features a homogeneous distributed runtime environment for applications.
- This environment is built by aggregating together physical and virtual nodes hosting the Aneka container.
- The container is a lightweight layer that interfaces with the hosting environment and manages the services deployed on a node.
- The interaction with the hosting platform is mediated through the Platform Abstraction Layer (PAL), which hides in its implementation all the heterogeneity of the different operating systems.
- The PAL, together with the container, represents the hosting environment of services which implement the core capabilities of the middleware and make it a dynamically composable and extensible system.
- The available services can be aggregated into three major categories:
  - **Fabric Services:** Fabric services implement the fundamental operations of the infrastructure of the cloud. These services include HA and failover for improved reliability, node membership and directory, resource provisioning, performance monitoring, and hardware profiling.
  - **Foundation Services:** Foundation services constitute the core functionalities of the Aneka middleware. They provide a basic set of capabilities that enhance application execution in the cloud. These services provide added value to the infrastructure and are of use to system administrators and developers. Within this category we can list storage management, resource reservation, reporting, accounting,

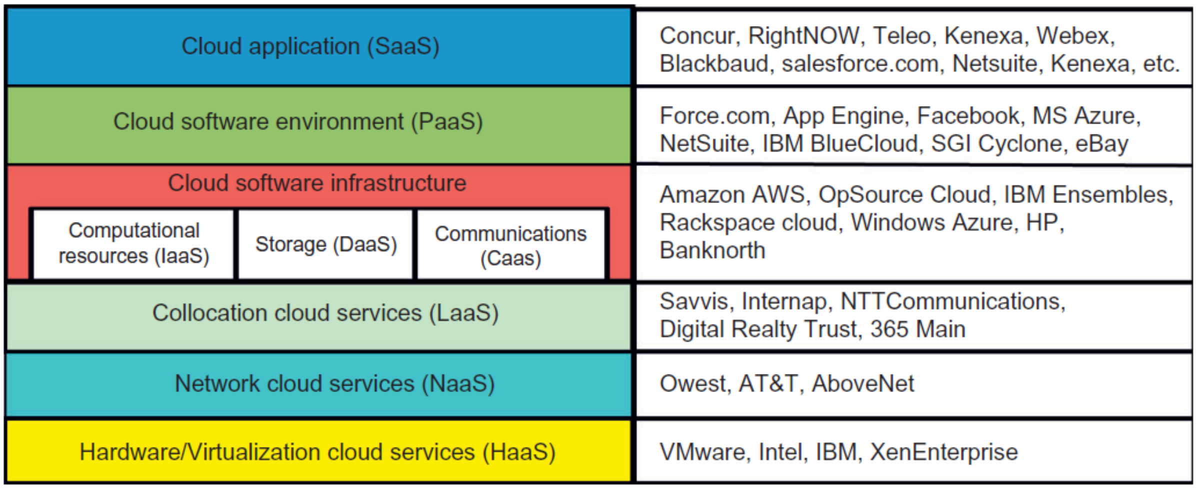
billing, services monitoring, and licensing. Services in this level operate across the range of supported application models.

- **Application Services:** Application services deal directly with the execution of applications and are in charge of providing the appropriate runtime environment for each application model. They leverage foundation and fabric services for several tasks of application execution such as elastic scalability, data transfer, and performance monitoring, accounting, and billing. At this level, Aneka expresses its true potential in supporting different application models and distributed programming patterns.

➢ EXTENDED CLOUD COMPUTING SERVICES

- Figure 3.13 shows six layers of cloud services, ranging from hardware, network, and collocation to infrastructure, platform, and software applications.
- We already introduced the top three service layers as SaaS, PaaS, and IaaS, respectively. The cloud platform provides PaaS, which sits on top of the IaaS infrastructure.
- The top layer offers SaaS. These must be implemented on the cloud platforms provided.
- Although the three basic models are dissimilar in usage, they are built one on top of another.
- The implication is that one cannot launch SaaS applications with a cloud platform.

- The cloud platform cannot be built if compute and storage infrastructures are not there.
- The bottom three layers are more related to physical requirements. The bottommost layer provides Hardware as a Service (HaaS).
- The next layer is for interconnecting all the hardware components, and is simply called Network as a Service (NaaS). Virtual LANs fall within the scope of NaaS.
- The next layer up offers Location as a Service (LaaS), which provides a collocation service to house, power, and secure all the physical hardware and network resources.
- The cloud infrastructure layer can be further subdivided as Data as a Service (DaaS) and Communication as a Service (CaaS) in addition to compute and storage in IaaS.



Cloud application (SaaS)	Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.
Cloud software environment (PaaS)	Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay
Cloud software infrastructure Computational resources (IaaS) Storage (DaaS) Communications (Caas)	Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth
Collocation cloud services (LaaS)	Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main
Network cloud services (NaaS)	Owest, AT&T, AboveNet
Hardware/Virtualization cloud services (HaaS)	VMware, Intel, IBM, XenEnterprise

Figure 3.13 Stack of six layers of cloud services and their providers.

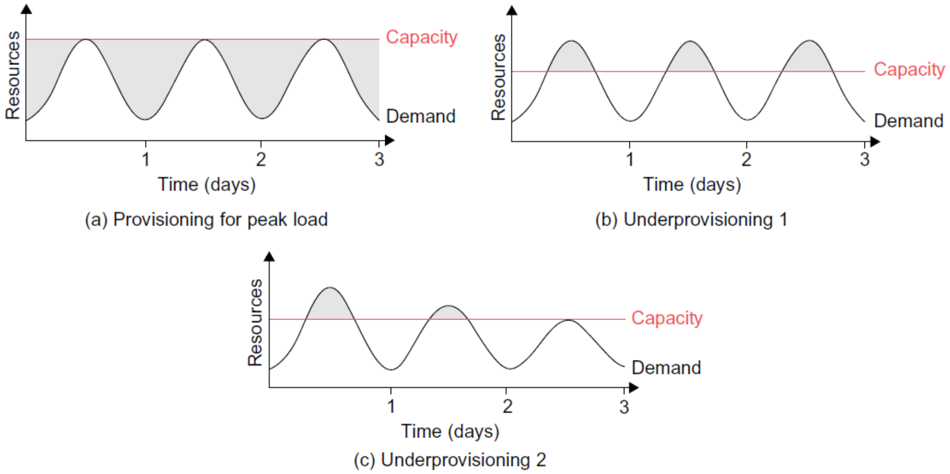
➢ RESOURCE PROVISIONING AND PLATFORM DEPLOYMENT

❖ Provisioning of Compute Resources (VMs)

- Providers supply cloud services by signing SLAs with end users. The SLAs must commit sufficient resources such as CPU, memory, and bandwidth that the user can use for a preset period.
- Under provisioning of resources will lead to broken SLAs and penalties.
- Overprovisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider.
- Deploying an autonomous system to efficiently provision resources to users is a challenging problem.

❖ Resource Provisioning Methods

- Figure 3.14 shows three cases of static cloud resource provisioning policies.
- In case (a), overprovisioning with the peak load causes heavy resource waste (shaded area).
- In case (b), underprovisioning (along the capacity line) of resources results in losses by both user and provider in that paid demand by the users (the shaded area above the capacity) is not served and wasted resources still exist for those demanded areas below the provisioned capacity.
- In case (c), the constant provisioning of resources with fixed capacity to a declining user demand could result in even worse resource waste



(a) Provisioning for peak load

(b) Underprovisioning 1

(c) Underprovisioning 2

❖ Demand-Driven Resource Provisioning

- This method adds or removes computing instances based on the current utilization level of the allocated resources.
- The demand-driven method automatically allocates two Xeon processors for the user application, when the user was using one Xeon processor more than 60 percent of the time for an extended period.

❖ Event-Driven Resource Provisioning

- This scheme adds or removes machine instances based on a specific time event.
- The scheme works better for seasonal or predicted events such as Christmastime in the West and the Lunar New Year in the East.
- During these events, the number of users grows before the event period and then decreases during the event period.

- **Popularity-Driven Resource Provisioning**
    - In this method, the Internet searches for popularity of certain applications and creates the instances by popularity demand.
    - The scheme anticipates increased traffic with popularity. Again, the scheme has a minimal loss of QoS, if the predicted popularity is correct.
    - Resources may be wasted if traffic does not occur as expected.

- **Dynamic Resource Deployment**
    - The cloud uses VMs as building blocks to create an execution environment across multiple resource sites.
    - The InterGrid-managed infrastructure was developed by a Melbourne University group.
    - Dynamic resource deployment can be implemented to achieve scalability in performance.
    - The Inter- Grid is a Java-implemented software system that lets users create execution cloud environments on top of all participating grid resources.

- **Provisioning of Storage Resources**
    - The data storage layer is built on top of the physical or virtual servers. As the cloud computing applications often provide service to users, it

is unavoidable that the data is stored in the clusters of the cloud provider.

- The service can be accessed anywhere in the world. One example is e-mail systems.
- A typical large e-mail system might have millions of users and each user can have thousands of e-mails and consume multiple gigabytes of disk space.
- A distributed file system is very important for storing large-scale data. However, other forms of data storage also exist. Some data does not need the namespace of a tree structure file system, and instead, databases are built with stored data files.

➢ VIRTUAL MACHINE CREATION AND MANAGEMENT

❖ Independent Service Management

- Independent services request facilities to execute many unrelated tasks. Commonly, the APIs provided are some web services that the developer can use conveniently.

❖ Running Third-Party Applications

- Cloud platforms have to provide support for building applications that are constructed by third-party application providers or programmers.
- As current web applications are often provided by using Web 2.0 forms (interactive applications with Ajax), the programming interfaces are

different from the traditional programming interfaces such as functions in runtime libraries.

- The APIs are often in the form of services.
- Web service application engines are often used by programmers for building applications. The web browsers are the user interface for end users.

❖ Virtual Machine Manager

- The VM manager is the link between the gateway and resources. The gateway doesn't share physical resources directly, but relies on virtualization technology for abstracting them.
- Hence, the actual resources it uses are VMs. The manager manages VMs deployed on a set of physical resources.
- The VM manager implementation is generic so that it can connect with different VIEs. Typically, VIEs can create and stop VMs on a physical cluster.

❖ Virtual Machine Templates

- A VM template is analogous to a computer's configuration and contains a description for a VM with the following static information:
  - The number of cores or processors to be assigned to the VM
  - The amount of memory the VM requires
  - The kernel used to boot the VM's operating system
  - The disk image containing the VM's file system
  - The price per hour of using a VM

- **Distributed VM Management**
  - Distributed VM manager makes requests for VMs and queries their status.
  - This manager requests VMs from the gateway on behalf of the user application.
  - The manager obtains the list of requested VMs from the gateway. This list contains a tuple of public IP/private IP addresses for each VM with Secure Shell (SSH) tunnels.
  - Users must specify which VM template they want to use and the number of VM instances needed, the deadline, the wall time, and the address for an alternative gateway.

MODULE 4

CLOUD PROGRAMMING

➢ PARALLEL AND DISTRIBUTED PROGRAMMING PARADIGMS

❖ Parallel Computing and Programming Paradigms

The system issues for running a typical parallel program in either a parallel or a distributed manner would include the following:

- **Partitioning** This is applicable to both computation and data as follows:
- **Computation partitioning** This splits a given job or a program into smaller tasks. Partitioning greatly depends on correctly identifying portions of the job or program that can be performed concurrently. In other words, upon identifying parallelism in the structure of the program, it can be divided into parts to be run on different workers. Different parts may process different data or a copy of the same data.
- **Data partitioning** This splits the input or intermediate data into smaller pieces. Similarly, upon identification of parallelism in the input data, it can also be divided into pieces to be processed on different workers. Data pieces may be processed by different parts of a program or a copy of the same program.
- **Mapping** This assigns the either smaller parts of a program or the smaller pieces of data to underlying resources. This process aims to appropriately assign such parts or pieces to be run simultaneously on different workers and is usually handled by resource allocators in the system.
- **Synchronization** Because different workers may perform different tasks, synchronization and coordination among workers is necessary so that race conditions are prevented and data dependency among different workers is properly managed. Multiple accesses to a shared resource by different workers may raise race conditions, whereas data dependency happens when a worker needs the processed data of other workers.
- **Communication** Because data dependency is one of the main reasons for communication among workers, communication is always triggered when the intermediate data is sent to workers.

- **Scheduling** For a job or program, when the number of computation parts (tasks) or data pieces is more than the number of available workers, a scheduler selects a sequence of tasks or data pieces to be assigned to the workers. It is worth noting that the resource allocator performs the actual mapping of the computation or data pieces to workers, while the scheduler only picks the next part from the queue of unassigned tasks based on a set of rules called the scheduling policy

Motivation for Programming Paradigms

1. simplicity of writing parallel programs
2. to improve productivity of programmers
3. to decrease programs’ time to market
4. to leverage underlying resources more efficiently
5. to increase system throughput
6. to support higher levels of abstraction

❖ MapReduce, Twister, and Iterative MapReduce

- MapReduce is a software framework which supports parallel and distributed computing on large data sets.
- This software framework abstracts the data flow of running a parallel program on a distributed computing system by providing users with two interfaces in the form of two functions: Map and Reduce. Users can override these two functions to interact with and manipulate the data flow of running their programs.
- Figure 4.1 illustrates the logical data flow from the Map to the Reduce function in MapReduce frameworks.

- In this framework, the “value” part of the data, (key, value), is the actual data, and the “key” part is only used by the MapReduce controller to control the data flow.

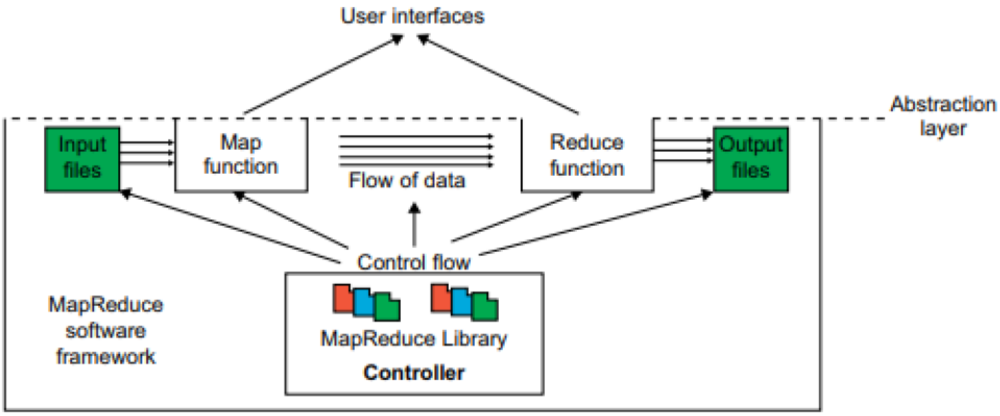


Figure 4.1: MapReduce Framework

-

The user overrides the Map and Reduce functions first and then invokes the provided MapReduce (Spec, & Results) function from the library to start the flow of data.

- The MapReduce function, MapReduce (Spec, & Results), takes an important parameter which is a specification object, the Spec.

- This object is first initialized inside the user's program, and then the user writes code to fill it with the names of input and output files, as well as other optional tuning parameters.

- This object is also filled with the name of the Map and Reduce functions to identify these userdefined functions to the MapReduce library.

- The overall structure of a user's program containing the Map, Reduce, and the Main functions is given below.

- The Map and Reduce are two major subroutines. They will be called to implement the desired function performed in the main program.

```
Map Function (... . )
{
    ... ...
}
Reduce Function (... . )
{
   ... ...
 }
Main Function (... . )
{
   Initialize Spec object ... ...
   MapReduce (Spec, & Results)
}
```

The input data to both the Map and the Reduce functions has a particular structure.

-

- This also pertains for the output data. The input data to the Map function is in the form of a (key, value) pair.
- For example, the key is the line offset within the input file and the value is the content of the line.
- The output data from the Map function is structured as (key, value) pairs called intermediate (key, value) pairs.
- In other words, the user-defined Map function processes each input (key, value) pair and produces a number of (zero, one, or more) intermediate (key, value) pairs.
- Here, the goal is to process all input (key, value) pairs to the Map function in parallel (Figure 4.2).
- In turn, the Reduce function receives the intermediate (key, value) pairs in the form of a group of intermediate values associated with one intermediate

key, (key, [set of values]).

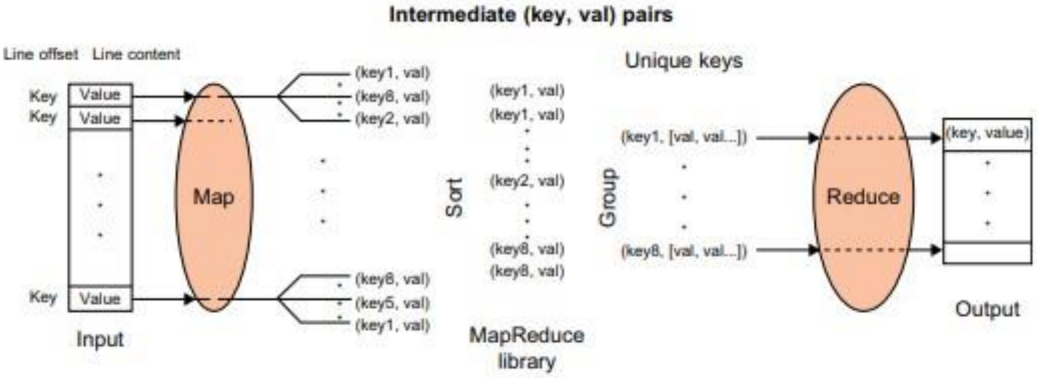


Figure 4.2: MapReduce logical data flow in 5 processing stages over successive (key, value) pairs.

In fact, the MapReduce framework forms these groups by first sorting the intermediate (key, value) pairs and then grouping values with the same key. It should be noted that the data is sorted to simplify the grouping process.

- The Reduce function processes each (key, [set of values]) group and produces a set of (key, value) pairs as output.

-

- To clarify the data flow in a sample MapReduce application, one of the wellknown MapReduce problems, namely word count, to count the number of occurrences of each word in a collection of documents is presented here.

- Figure 4.3 demonstrates the data flow of the word-count problem for a simple input file containing only two lines as follows: (1) “most people ignore most poetry” and (2) “most poetry ignores most people.”

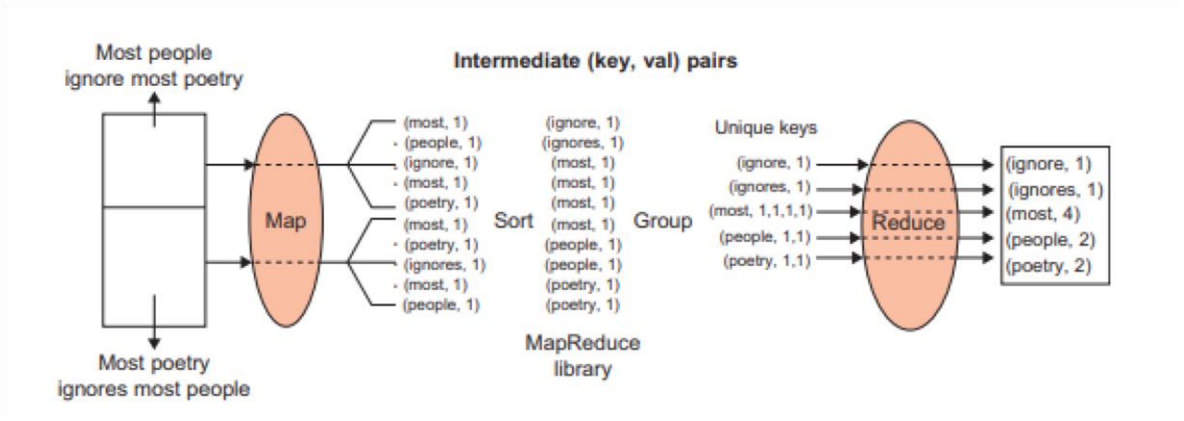


Figure 4.3: The data flow of a word-count problem using the MapReduce functions

- In this case, the Map function simultaneously produces a number of intermediate (key, value) pairs for each line of content so that each word is the intermediate key with 1 as its intermediate value; for example, (ignore, 1).

- Then the MapReduce library collects all the generated intermediate (key, value) pairs and sorts them to group the 1’s for identical words; for example, (people, [1,1]).

Groups are then sent to the Reduce function in parallel so that it can sum up the 1 values for each word and generate the actual number of occurrence for each word in the file; for example, (people, 2).

- The main responsibility of the MapReduce framework is to efficiently run a user’s program on a distributed computing system.

-

- Therefore, the MapReduce framework meticulously handles all partitioning, mapping, synchronization, communication, and scheduling details of such data flows:

1. **Data partitioning** The MapReduce library splits the input data (files), already stored in GFS, into M pieces that also correspond to the number of map tasks.

2. **Computation partitioning** This is implicitly handled (in the MapReduce framework) by obliging users to write their programs in the form of the Map and Reduce functions. Therefore, the MapReduce library only generates copies of a user program (e.g., by a fork system call) containing the Map and the Reduce functions, distributes them, and starts them up on a number of available computation engines.

3. **Determining the master and workers** The MapReduce architecture is based on a masterworker model. Therefore, one of the copies of the user program becomes the master and the rest become workers. The master picks idle workers, and assigns the map and reduce tasks to them. A map/reduce worker is typically a computation engine such as a cluster node to run map/ reduce tasks by executing Map/Reduce functions. Steps 4–7 describe the map workers.

4. **Reading the input data (data distribution)** Each map worker reads its corresponding portion of the input data, namely the input data split, and sends it to its Map function. Although a map worker may run more than one Map

function, which means it has been assigned more than one input data split, each worker is usually assigned one input split only.

5. **Map function** Each Map function receives the input data split as a set of

(key, value) pairs to process and produce the intermediated (key, value) pairs. **6. Combiner function** This is an optional local function within the map worker which applies to intermediate (key, value) pairs. The user can invoke the Combiner function inside the user program. The Combiner function runs the same code written by users for the Reduce function as its functionality is identical to it. The Combiner function merges the local data of each map worker before sending it over the network to effectively reduce its communication costs.

7. **Partitioning function** As mentioned in our discussion of the MapReduce data flow, the intermediate (key, value) pairs with identical keys are grouped together because all values inside each group should be processed by only one Reduce function to generate the final result. However, in real implementations, since there are M map and R reduce tasks, intermediate (key, value) pairs with the same key might be produced by different map tasks, although they should be grouped and processed together by one Reduce function only.

8. **Synchronization** MapReduce applies a simple synchronization policy to coordinate map workers with reduce workers, in which the communication between them starts when all map tasks finish.

9. **Communication** Reduce worker i, already notified of the location of region i of all map workers, uses a remote procedure call to read the data from the respective region of all map workers. Since all reduce workers read the data from all map workers, all-to-all communication among all map and reduce workers, which incurs network congestion, occurs in the network. This issue is one of the major bottlenecks in increasing the performance of such systems **10. Sorting and Grouping** When the process of reading the input data is finalized by a reduce worker, the data is initially buffered in the local disk of the reduce worker. Then the reduce worker groups intermediate (key, value) pairs by sorting the data based on their keys, followed by grouping all occurrences of identical keys.

11. Reduce function The reduce worker iterates over the grouped (key, value) pairs, and for each unique key, it sends the key and corresponding values to the Reduce function. Then this function processes its input data and stores the output results in predetermined files in the user’s program.

❖ Twister and Iterative MapReduce

- It is important to understand the performance of different runtimes and, in particular, to compare MPI and MapReduce.

- The two major sources of parallel overhead are load imbalance and communication (which is equivalent to synchronization overhead as communication synchronizes parallel units [threads or processes]).

- The communication overhead in MapReduce can be quite high, for two reasons:

  a) MapReduce reads and writes via files, whereas MPI transfers information directly between nodes over the network.

  b) MPI does not transfer all data from node to node, but just the amount needed to update information. We can call the MPI flow δ flow and the MapReduce flow full data flow.

The same phenomenon is seen in all "classic parallel" loosely synchronous applications which typically exhibit an iteration structure over compute phases followed by communication phases.

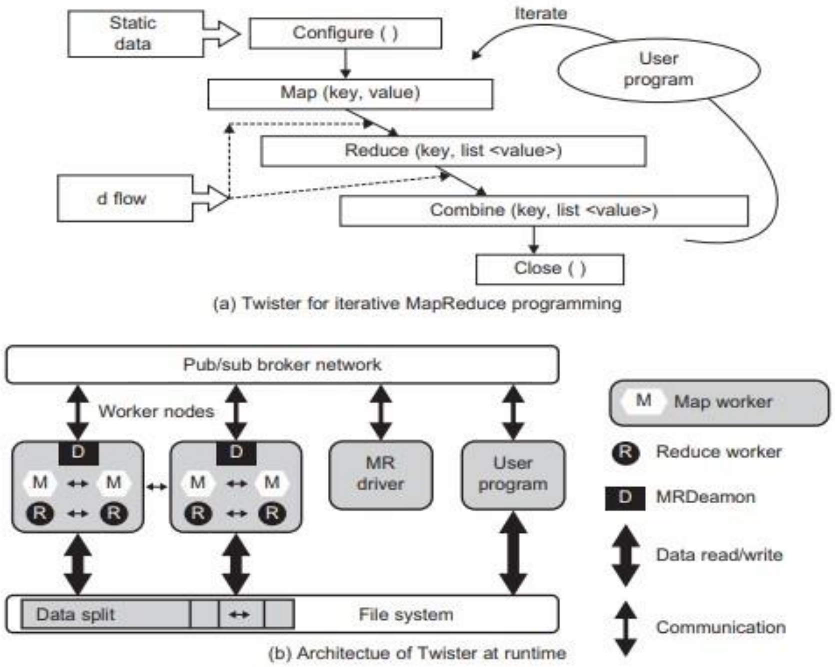
- We can address the performance issues with two important changes:

1. Stream information between steps without writing intermediate steps to disk.

2. Use long-running threads or processors to communicate the δ (between iterations) flow.

- These changes will lead to major performance increases at the cost of poorer fault tolerance and ease to support dynamic changes such as the number of available nodes.

- The Twister programming paradigm and its implementation architecture at run time are illustrated in Figure 4.4.(a, b).



(a) Twister for iterative MapReduce programming

(b) Architectue of Twister at runtime

Figure 4.4: Twister

-

Twister distinguishes the static data which is never reloaded from the dynamic δ flow that is communicated.

- The Map-Reduce pair is iteratively executed in long-running threads. We compare in Figure 4.5.
- The different thread and process structures of 4 parallel programming paradigms: namely Hadoop, Dryad, Twister (also called MapReduce++), and MPI.
- Dryad can use pipes and avoids costly disk writing according to the original papers.

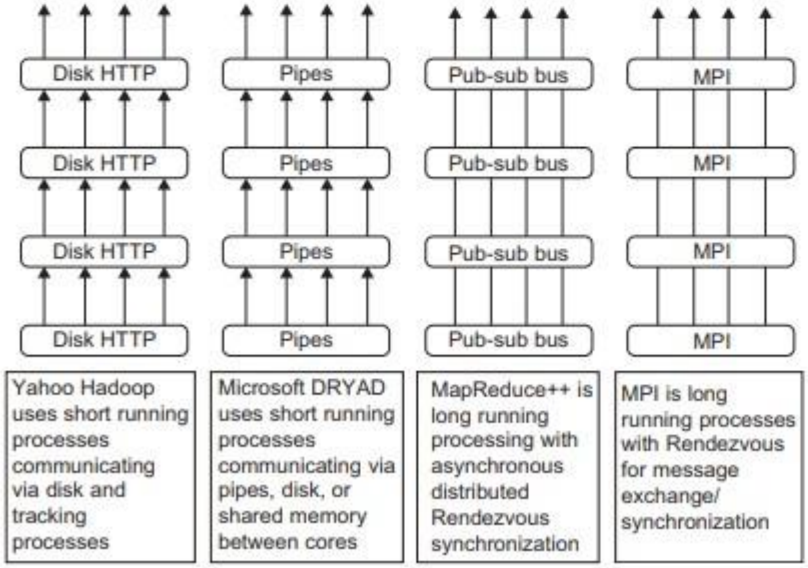

Disk HTTP
Disk HTTP
Disk HTTP
Disk HTTP
Pipes
Pipes
Pipes
Pipes
Pub-sub bus
Pub-sub bus
Pub-sub bus
Pub-sub bus
MPI
MPI
MPI
MPI
Yahoo Hadoop uses short running processes communicating via disk and tracking processes
Microsoft DRYAD uses short running processes communicating via pipes, disk, or shared memory between cores
MapReduce++ is long running processing with asynchronous distributed Rendezvous synchronization
MPI is long running processes with Rendezvous for message exchange/ synchronization

Figure 4.5: Thread and process structure of four parallel programming paradigms at runtimes.

❖ Hadoop Library from Apache

- Hadoop is an open source implementation of MapReduce coded and released in Java (rather than C) by Apache.

The Hadoop implementation of MapReduce uses the Hadoop Distributed File System (HDFS) as its underlying layer rather than GFS.

- The Hadoop core is divided into two fundamental layers: the MapReduce engine and HDFS.
- The MapReduce engine is the computation engine running on top of HDFS as its data storage manager.

HDFS: HDFS is a distributed file system inspired by GFS that organizes files and stores their data on a distributed computing system.

HDFS Architecture: HDFS has a master/slave architecture containing a single NameNode as the master and a number of DataNodes as workers

(slaves). To store a file in this architecture, HDFS splits the file into fixedsize blocks (e.g., 64 MB) and stores them on workers (DataNodes). The mapping of blocks to DataNodes is determined by the NameNode. The

NameNode (master) also manages the file system's metadata and namespace.

- **HDFS Features:** Distributed file systems have special requirements, such as performance, scalability, concurrency control, fault tolerance, and security requirements, to operate efficiently. However, because HDFS is not a general-purpose file system, as it only executes specific types of applications, it does not need all the requirements of a general distributed file system.
- **HDFS Fault Tolerance:** One of the main aspects of HDFS is its fault tolerance characteristic. Since Hadoop is designed to be deployed on lowcost hardware by default, a hardware failure in this system is considered to be common rather than an exception. Therefore, Hadoop considers the following issues to fulfill reliability requirements of the file system :

    1. **Block replication** To reliably store data in HDFS, file blocks are replicated in this system. In other words, HDFS stores a file as a set of blocks and each block is replicated and distributed across the whole cluster. The replication factor is set by the user and is three by default.

    2. **Replica placement** The placement of replicas is another factor to fulfill the desired fault tolerance in HDFS. Although storing replicas on different nodes (DataNodes) located in different racks across the whole cluster provides more reliability, it is sometimes ignored as the cost of communication between two nodes in different racks is relatively high in comparison with that of different nodes located in the same rack.

Therefore, sometimes HDFS compromises its reliability to achieve lower communication costs.

3. **Heartbeat and Blockreport messages** Heartbeats and Blockreports are periodic messages sent to the NameNode by each DataNode in a cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly, while each Blockreport contains a list of all blocks on a DataNode. The NameNode receives such messages because it is the sole decision maker of all replicas in the system.

- **HDFS High-Throughput Acces to Large Data Sets (Files):** Because HDFS is primarily designed for batch processing rather than interactive processing, data access throughput in HDFS is more important than latency. Also, because applications run on HDFS typically have large data sets, individual files are broken into large blocks (e.g., 64 MB) to allow HDFS to decrease the amount of metadata storage required per file. This provides two advantages: The list of blocks per file will shrink as the size of individual blocks increases, and by keeping large amounts of data sequentially within a block, HDFS provides fast streaming reads of data.

- **HDFS Operation:** The control flow of HDFS operations such as write and read can properly highlight roles of the NameNode and DataNodes in the managing operations. In this section, the control flow of the main operations of HDFS on files is further described to manifest the interaction between the user, the NameNode, and the DataNodes in such systems

  a) **Reading a file** To read a file in HDFS, a user sends an "open" request to the NameNode to get the location of file blocks. For each file block, the NameNode returns the address of a set of DataNodes containing replica information for the requested file. The number of addresses depends on the number of block replicas. Upon receiving such information, the user calls the read function to connect to the closest DataNode containing the first block of the file. After the first block is streamed from the respective DataNode to the user, the established connection is terminated and the same process is repeated for all blocks of the requested file until the whole file is streamed to the user. •

  b) **Writing to a file** To write a file in HDFS, a user sends a "create" request to the NameNode to create a new file in the file system namespace. If the file does not exist, the NameNode notifies the user and allows him to start writing data to the file by calling the write function. The first block of the file is written to an internal queue termed the data queue while a data streamer monitors its writing into a DataNode. Since each file block needs to be replicated by a predefined factor, the data streamer first sends a request to the NameNode to get a list of suitable DataNodes to store replicas of the first block.

Architecture of MapReduce in Hadoop

- The topmost layer of Hadoop is the MapReduce engine that manages the data flow and control flow of MapReduce jobs over distributed computing systems.

- Figure 4.6 shows the MapReduce engine architecture cooperating with HDFS. Similar to HDFS, the MapReduce engine also has a master/slave architecture consisting of a single JobTracker as the master and a number of TaskTrackers as the slaves (workers).

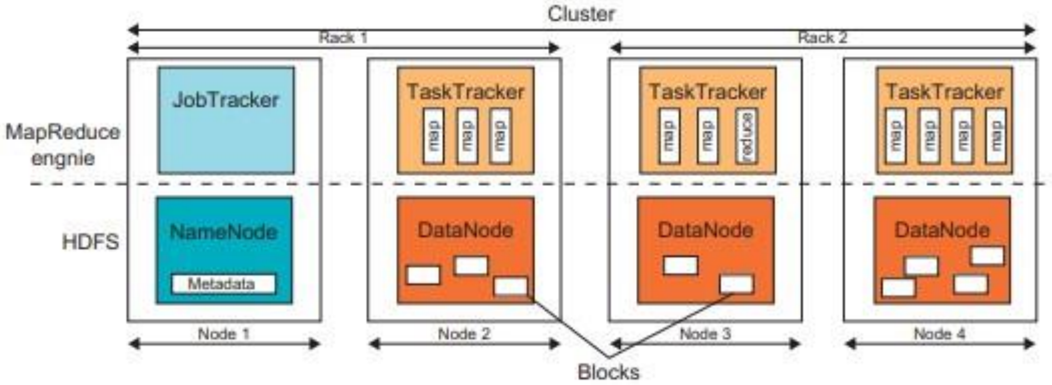


Figure 4.6: HDFS and MapReduce architecture in Hadoop

- The JobTracker manages the MapReduce job over a cluster and is responsible for monitoring jobs and assigning tasks to TaskTrackers.
- The TaskTracker manages the execution of the map and/or reduce tasks on a single computation node in the cluster.
- Each TaskTracker node has a number of simultaneous execution slots, each executing either a map or a reduce task. Slots are defined as the number of simultaneous threads supported by CPUs of the TaskTracker node.

Running a Job in Hadoop

- Three components contribute in running a job in this system: a user node, a JobTracker, and several TaskTrackers.
- The data flow starts by calling the runJob(conf) function inside a user program running on the user node, in which conf is an object containing some tuning parameters for the MapReduce framework and HDFS.
- The runJob(conf) function and conf are comparable to the MapReduce(Spec, &Results) function and Spec in the first implementation of MapReduce by Google.
- Figure 1.7 depicts the data flow of running a MapReduce job in Hadoop

a) **Job Submission** Each job is submitted from a user node to the JobTracker node that might be situated in a different node within the cluster through the following procedure:

- A user node asks for a new job ID from the JobTracker and computes input file splits.
- The user node copies some resources, such as the job's JAR file, configuration file, and computed input splits, to the JobTracker's file system.
- The user node submits the job to the JobTracker by calling the submitJob() function.

b) **Task assignment** The JobTracker creates one map task for each computed input split by the user node and assigns the map tasks to the execution slots of the TaskTrackers. The JobTracker considers the localization of the data when assigning the map tasks to the

TaskTrackers. The JobTracker also creates reduce tasks and assigns them to the TaskTrackers. The number of reduce tasks is predetermined by the user, and there is no locality consideration in assigning them.

c) **Task execution** The control flow to execute a task (either map or reduce) starts inside the TaskTracker by copying the job JAR file to its file system. Instructions inside the job JAR file are executed after launching a Java Virtual Machine (JVM) to run its map or reduce task.

d) **Task running check** A task running check is performed by receiving periodic heartbeat messages to the JobTracker from the TaskTrackers. Each heartbeat notifies the JobTracker that the sending TaskTracker is alive, and whether the sending TaskTracker is ready to run a new task.

❖ Pig Latin High-Level Languages

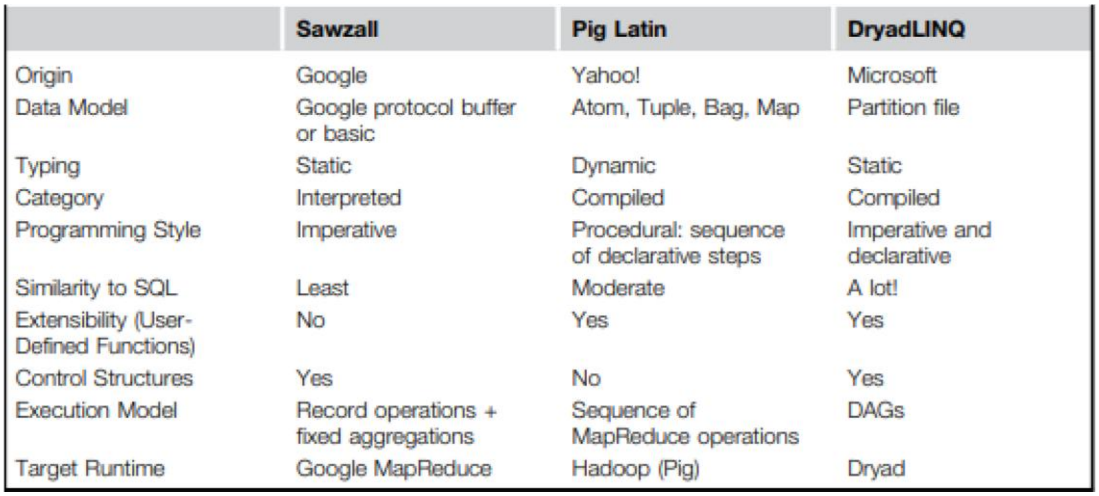
- Pig Latin is a high-level data flow language developed by Yahoo! that has been implemented on top of Hadoop in the Apache Pig project.
- Pig Latin, Sawzall and DryadLINQ are different approaches to building languages on top of MapReduce and its extensions. They are compared in Table 4.1.
- DryadLINQ is building directly on SQL while the other two languages are of NOSQL heritage, although Pig Latin supports major SQL constructs including Join, which is absent in Sawzall.

- Each language automates the parallelism, so you only think about manipulation of individual elements and then invoke supported collective operations.

- This is possible, of course, because needed parallelism can be cleanly implemented by independent tasks with “side effects” only presented insupported collective operations.
- This is an important general approach to parallelism and was seen, for example, a long time ago in High Performance Fortran.

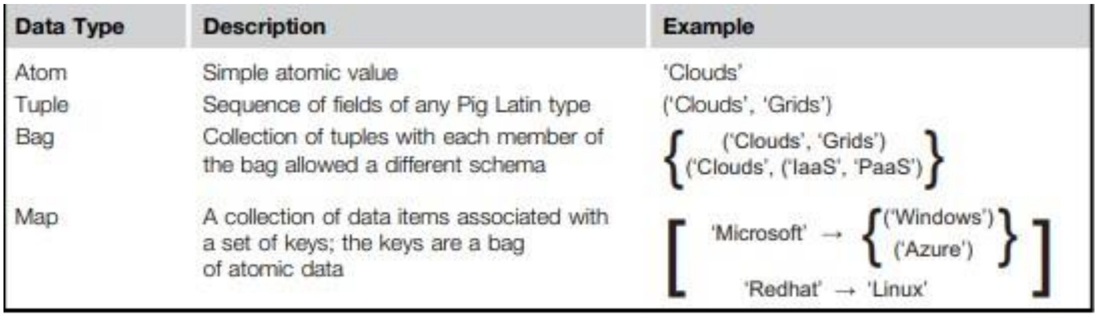
- There are several discussions of Pig and Pig Latin in the literature, and here we summarize the language features. Table 4.2 lists the four data types in Pig Latin and Table 4.3 the 14 operators.

Table 4.1: Comparison of High-Level Data Analysis Languages



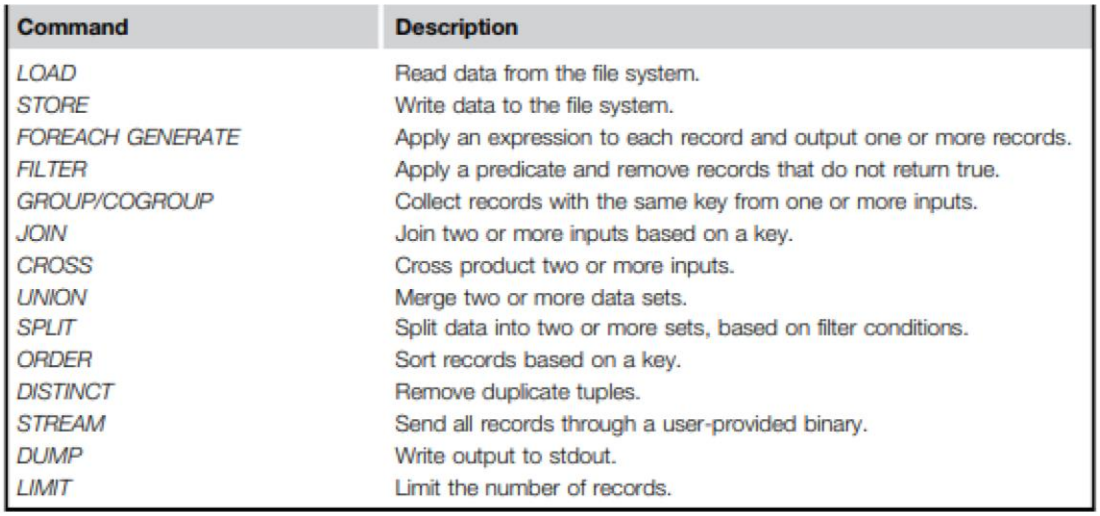
	Sawzall	**Pig Latin**	**DryadLINQ**
Origin	Google	Yahoo!	Microsoft
Data Model	Google protocol buffer or basic	Atom, Tuple, Bag, Map	Partition file
Typing	Static	Dynamic	Static
Category	Interpreted	Compiled	Compiled
Programming Style	Imperative	Procedural: sequence of declarative steps	Imperative and declarative
Similarity to SQL	Least	Moderate	A lot!
Extensibility (User-Defined Functions)	No	Yes	Yes
Control Structures	Yes	No	Yes
Execution Model	Record operations + fixed aggregations	Sequence of MapReduce operations	DAGs
Target Runtime	Google MapReduce	Hadoop (Pig)	Dryad

Table 4.2: Pig Latin Data Types



Data Type	**Description**	**Example**
Atom	Simple atomic value	‘Clouds’
Tuple	Sequence of fields of any Pig Latin type	(‘Clouds’, ‘Grids’)
Bag	Collection of tuples with each member of the bag allowed a different schema	{ (‘Clouds’, ‘Grids’) (‘Clouds’, (‘IaaS’, ‘PaaS’)) }
Map	A collection of data items associated with a set of keys; the keys are a bag of atomic data	[‘Microsoft’ → { (‘Windows’) (‘Azure’) } ‘Redhat’ → ‘Linux’]

Table 4.3: Pig Latin Operators



Command	Description
LOAD	Read data from the file system.
STORE	Write data to the file system.
FOREACH GENERATE	Apply an expression to each record and output one or more records.
FILTER	Apply a predicate and remove records that do not return true.
GROUP/COGROUP	Collect records with the same key from one or more inputs.
JOIN	Join two or more inputs based on a key.
CROSS	Cross product two or more inputs.
UNION	Merge two or more data sets.
SPLIT	Split data into two or more sets, based on filter conditions.
ORDER	Sort records based on a key.
DISTINCT	Remove duplicate tuples.
STREAM	Send all records through a user-provided binary.
DUMP	Write output to stdout.
LIMIT	Limit the number of records.

- Pig Latin operations are performed in the order listed as a data flow pipeline.

  This is in contrast to declarative SQL where one just specifies "what" has to be done, not how it is to be done.

- Pig Latin supports user-defined functions, as illustrated in the preceding code, as first-class operations in the language which could be an advantage over SQL.

- User-defined functions can be placed in Load, Store, Group, Filter, and Foreach operators, depending on user preference.

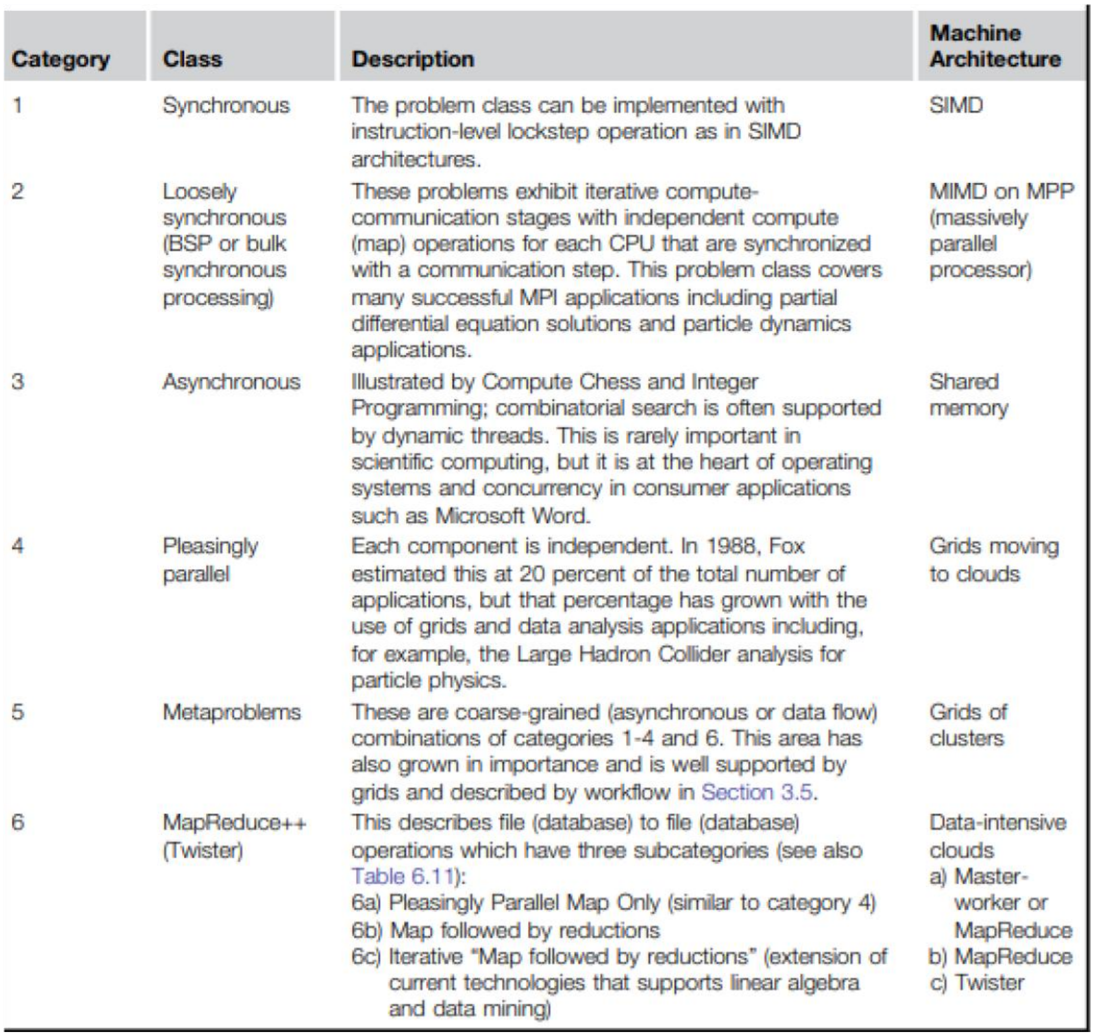
❖ Mapping Applications to Parallel and Distributed Systems

- Fox has discussed mapping applications to different hardware and software in terms of five application architectures.

- These initial five categories are listed in Table 4.4, followed by a sixth emerging category to describe data-intensive computing.

- The original classifications largely described simulations and were not aimed directly at data analysis. It is instructive to briefly summarize them and then explain the new category.

Table 4.4: Application Classification for Parallel and Distributed Systems



Category	Class	Description	Machine Architecture
1	Synchronous	The problem class can be implemented with instruction-level lockstep operation as in SIMD architectures.	SIMD
2	Loosely synchronous (BSP or bulk synchronous processing)	These problems exhibit iterative compute-communication stages with independent compute (map) operations for each CPU that are synchronized with a communication step. This problem class covers many successful MPI applications including partial differential equation solutions and particle dynamics applications.	MIMD on MPP (massively parallel processor)
3	Asynchronous	Illustrated by Compute Chess and Integer Programming; combinatorial search is often supported by dynamic threads. This is rarely important in scientific computing, but it is at the heart of operating systems and concurrency in consumer applications such as Microsoft Word.	Shared memory
4	Pleasingly parallel	Each component is independent. In 1988, Fox estimated this at 20 percent of the total number of applications, but that percentage has grown with the use of grids and data analysis applications including, for example, the Large Hadron Collider analysis for particle physics.	Grids moving to clouds
5	Metaproblems	These are coarse-grained (asynchronous or data flow) combinations of categories 1-4 and 6. This area has also grown in importance and is well supported by grids and described by workflow in Section 3.5.	Grids of clusters
6	MapReduce++ (Twister)	This describes file (database) to file (database) operations which have three subcategories (see also Table 6.11): 6a) Pleasingly Parallel Map Only (similar to category 4) 6b) Map followed by reductions 6c) Iterative "Map followed by reductions" (extension of current technologies that supports linear algebra and data mining)	Data-intensive clouds a) Master-worker or MapReduce b) MapReduce c) Twister

- Category 1 was popular 20 years ago, but is no longer significant. It describes applications that can be parallelized with lock-step operations controlled by hardware.

- Category 1 corresponds to regular problems, whereas category 2 includes dynamic irregular cases with complex geometries for solving partial differential equations or particle dynamics. Note that synchronous problems are still around, but they are run on MIMD machines with the SPMD model. Category 2 consists of compute–communicate phases and the computations are synchronized by communication. No additional synchronization is needed.

- Category 3 consists of asynchronously interacting objects and is often considered the people's view of a typical parallel problem. It probably does describe the concurrent threads in a modern operating system, as well as some important applications, such as event-driven simulations and areas such as search in computer games and graph algorithms.

- Category 4 is the simplest algorithmically, with disconnected parallel components. However, the importance of this category has probably grown since the original 1988 analysis when it was estimated to account for 20 percent of all parallel computing. Both grids and clouds are very natural for this class, which does not need high-performance communication between different nodes.

- Category 5 refers to the coarse-grained linkage of different “atomic” problems. This area is clearly common and is expected to grow in importance. We use a two-level programming model with the meta-problem (workflow) linkage specified in one fashion and the component problems with approaches.

➢ PROGRAMMING SUPPORT OF GOOGLE APP ENGINE

❖ Programming the Google App Engine

- Figure 4.77 summarizes some key features of GAE programming model for two supported languages: Java and Python.

- A client environment that includes an Eclipse plug-in for Java allows you to debug your GAE on your local machine.

- Also, the GWT Google Web Toolkit is available for Java web application developers.

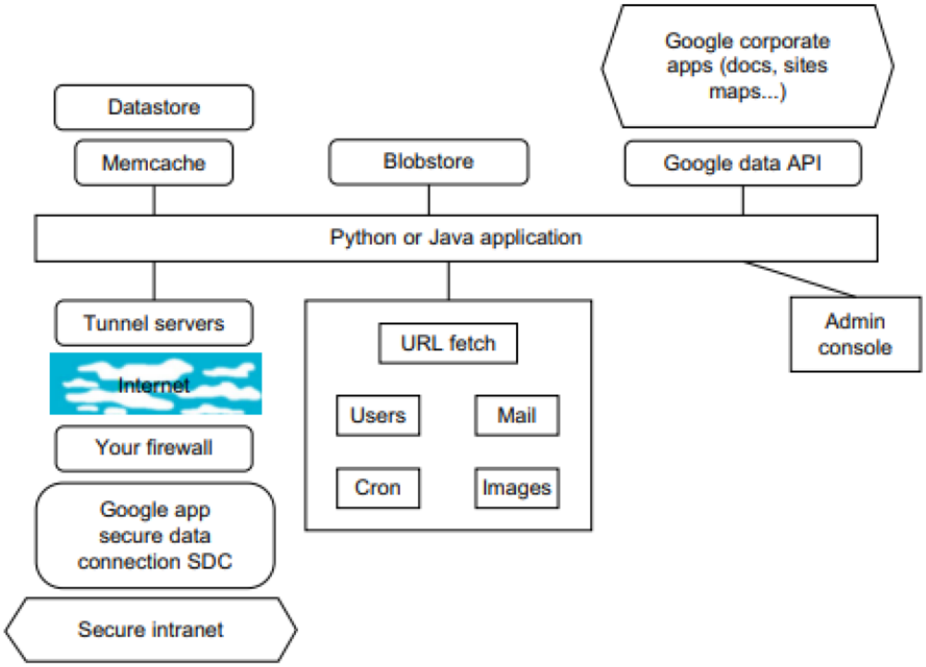


Figure 4.7: Programming environment for Google AppEngine

- Developers can use this, or any other language using a JVMbased interpreter or compiler, such as JavaScript or Ruby.
- Python is often used with frameworks such as Django and CherryPy, but Google also supplies a built in webapp Python environment.

-

There are several powerful constructs for storing and accessing data.

- The data store is a NOSQL data management system for entities that can be, at most, 1 MB in size and are labeled by a set of schema-less properties.
- Queries can retrieve entities of a given kind filtered and sorted by the values of the properties.
- Java offers Java Data Object (JDO) and Java Persistence API (JPA) interfaces implemented by the open source Data Nucleus Access platform, while Python has a SQL-like query language called GQL.
- The data store is strongly consistent and uses optimistic concurrency control.
- The performance of the data store can be enhanced by in-memory caching using the memcache, which can also be used independently of the data store.
- Recently, Google added the blobstore which is suitable for large files as its size limit is 2 GB. There are several mechanisms for incorporating external resources.
- The Google SDC Secure Data Connection can tunnel through the Internet and link your intranet to an external GAE application.
- The URL Fetch operation provides the ability for applications to fetch resources and communicate with other hosts over the Internet using HTTP and HTTPS requests.
- There is a specialized mail mechanism to send e-mail from your GAE application.
- Applications can access resources on the Internet, such as web services or other data, using GAE's URL fetch service.

  The URL fetch service retrieves web resources using the same highspeed Google infrastructure that retrieves web pages for many other Google products.
- There are dozens of Google "corporate" facilities including maps, sites, groups, calendar, docs, and YouTube, among others. These support the Google Data API which can be used inside GAE.

-

- An application can use Google Accounts for user authentication. Google Accounts handles user account creation and sign-in, and a user that already has a Google account (such as a Gmail account) can use that account with your app.
- GAE provides the ability to manipulate image data using a dedicated Images service which can resize, rotate, flip, crop, and enhance images.
- An application can perform tasks outside of responding to web requests.
- Your application can perform these tasks on a schedule that you configure, such as on a daily or hourly basis using "cron jobs," handled by the Cron service.

❖ Google File System (GFS)

- GFS was built primarily as the fundamental storage service for Google's search engine. As the size of the web data that was crawled and saved was quite substantial, Google needed a distributed file system to redundantly store massive amounts of data on cheap and unreliable computers.
- None of the traditional distributed file systems can provide such functions and hold such large amounts of data. In addition, GFS was designed for Google applications, and Google applications were built for GFS.

  In traditional file system design, such a philosophy is not attractive, as there should be a clear interface between applications and the file system, such as a POSIX interface.
- As servers are composed of inexpensive commodity components, it is the norm rather than the exception that concurrent failures will occur all the time.
- Another concerns the file size in GFS. GFS typically will hold a large number of huge files, each 100 MB or larger, with files that are multiple GB in size quite common.

-

- Thus, Google has chosen its file data block size to be 64 MB instead of the 4 KB in typical traditional file systems.
- The I/O pattern in the Google application is also special. Files are typically written once, and the write operations are often the appending data blocks to the end of files. Multiple appending operations might be concurrent.
- Reliability is achieved by using replications (i.e., each chunk or data block of a file is replicated across more than three chunk servers).
- A single master coordinates access as well as keeps the metadata. This decision simplified the design and management of the whole cluster.
- Developers do not need to consider many difficult issues in distributed systems, such as distributed consensus.
- There is no data cache in GFS as large streaming reads and writes represent neither time nor space locality. GFS provides a similar, but not identical, POSIX file system accessing interface.
- Figure 4.8 shows the GFS architecture. It is quite obvious that there is a single master in the whole cluster.

  Other nodes act as the chunk servers for storing data, while the single master stores the metadata.
- The file system namespace and locking facilities are managed by the master.
- The master periodically communicates with the chunk servers to collect management information as well as give instructions to the chunk servers to do work such as load balancing or fail recovery.

-

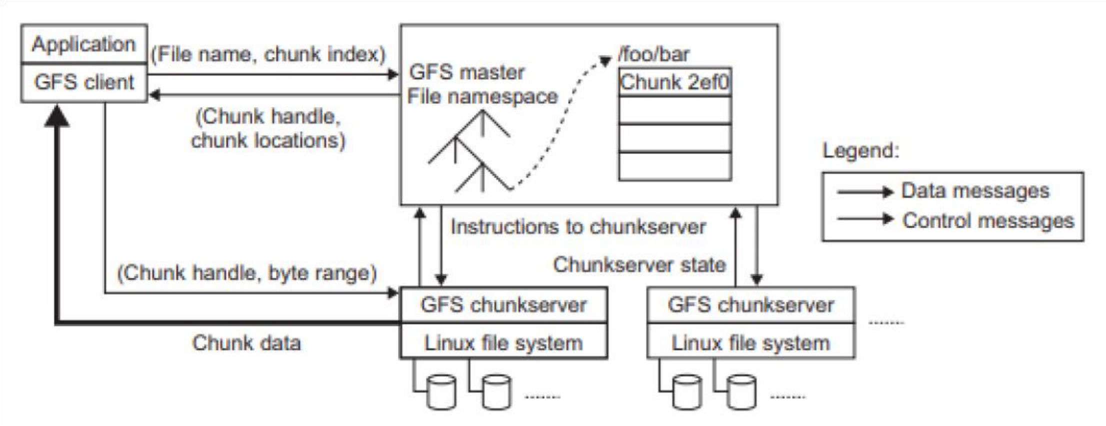


Figure 4.8: Architecture of Google File System (GFS)

- The master has enough information to keep the whole cluster in a healthy state. With a single master, many complicated distributed algorithms can be avoided and the design of the system can be simplified.

- However, this design does have a potential weakness, as the single GFS master could be the performance bottleneck and the single point of failure.

- To mitigate this, Google uses a shadow master to replicate all the data on the master, and the design guarantees that all the data operations are performed directly between the client and the chunk server.

- The control messages are transferred between the master and the clients and they can be cached for future use. With the current quality of

-

commodity servers, the single master can handle a cluster of more than 1,000 nodes.

- Figure 6.9 shows the data mutation (write, append operations) in GFS.

  Data blocks must be created for all replicas.

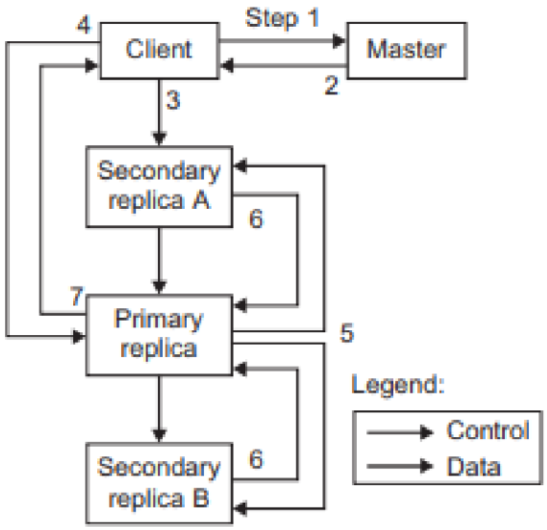


Figure 4.9: Data mutation sequence in GFS

- The goal is to minimize involvement of the master. The mutation takes the following steps:

1. The client asks the master which chunk server holds the current lease for the chunk and the locations of the other replicas. If no one has a lease, the master grants one to a replica it chooses (not shown).

2. The master replies with the identity of the primary and the locations of the other (secondary) replicas. The client caches this data for future mutations. It needs to contact the master again only when the primary becomes unreachable or replies that it no longer holds a lease.

3. The client pushes the data to all the replicas. A client can do so in any order. Each chunk server will store the data in an internal LRU buffer cache until the data is used or aged out. By decoupling the data flow from the control flow, we can improve performance by scheduling

-

the expensive data flow based on the network topology regardless of which chunk server is the primary.

4. Once all the replicas have acknowledged receiving the data, the client sends a write request to the primary. The request identifies the data pushed earlier to all the replicas. The primary assigns consecutive serial numbers to all the mutations it receives, possibly from multiple clients, which provides the necessary serialization. It applies the mutation to its own local state in serial order.

5. The primary forwards the write request to all secondary replicas. Each secondary replica applies mutations in the same serial number order assigned by the primary.

6. The secondaries all reply to the primary indicating that they have completed the operation.

7. The primary replies to the client. Any errors encountered at any replicas are reported to the client. In case of errors, the write corrects at the primary and an arbitrary subset of the secondary replicas. The client request is considered to have failed, and the modified region is left in an inconsistent state. Our client code handles such errors by retrying the failed mutation. It will make a few attempts at steps 3 through 7 before falling back to a retry from the beginning of the write

- GFS was designed for high fault tolerance and adopted some methods to achieve this goal.
- Master and chunk servers can be restarted in a few seconds, and with such a fast recovery capability, the window of time in which the data is unavailable can be greatly reduced.
- The shadow master handles the failure of the GFS master. For data integrity, GFS makes checksums on every 64 KB block in each chunk.
- With the previously discussed design and implementation, GFS can achieve the goals of high availability (HA), high performance, and large scale.

-

❖ BigTable, Google’s NOSQL System

- BigTable was designed to provide a service for storing and retrieving structured and semistructured data.
- BigTable applications include storage of web pages, per-user data, and geographic locations.
- Here we use web pages to represent URLs and their associated data, such as contents, crawled metadata, links, anchors, and page rank values.
- Per-user data has information for a specific user and includes such data as user preference settings, recent queries/search results, and the user’s e-mails.
- Geographic locations are used in Google’s well-known Google Earth software. Geographic locations include physical entities (shops, restaurants, etc.), roads, satellite image data, and user annotations.
- The design and implementation of the BigTable system has the following goals.
- The applications want asynchronous processes to be continuously updating different pieces of data and want access to the most current data at all times.
- The database needs to support very high read/write rates and the scale might be millions of operations per second.
- Also, the database needs to support efficient scans over all or interesting subsets of data, as well as efficient joins of large one-to-one and one-to-many data sets.
- The application may need to examine data changes over time (e.g., contents of a web page over multiple crawls).
- Thus, BigTable can be viewed as a distributed multilevel map. It provides a fault-tolerant and persistent database as in a storage service. T

-

- he BigTable system is scalable, which means the system has thousands of servers, terabytes of in-memory data, petabytes of disk-based data, millions of reads/writes per second, and efficient scans.
- Also, BigTable is a self-managing system (i.e., servers can be added/removed dynamically and it features automatic load balancing).
- The BigTable system is built on top of an existing Google cloud infrastructure.

  BigTable uses the following building blocks:

  1. GFS: stores persistent state
  2. Scheduler: schedules jobs involved in BigTable serving
  3. Lock service: master election, location bootstrapping
  4. MapReduce: often used to read/write BigTable data

Tablet Location Hierarchy

- Figure 4.10 shows how to locate the BigTable data starting from the file stored in Chubby.
- The first level is a file stored in Chubby that contains the location of the root tablet.
- The root tablet contains the location of all tablets in a special METADATA table.
- Each METADATA tablet contains the location of a set of user tablets. The root tablet is just the first tablet in the METADATA table, but is treated specially; it is never split to ensure that the tablet location hierarchy has no more than three levels.

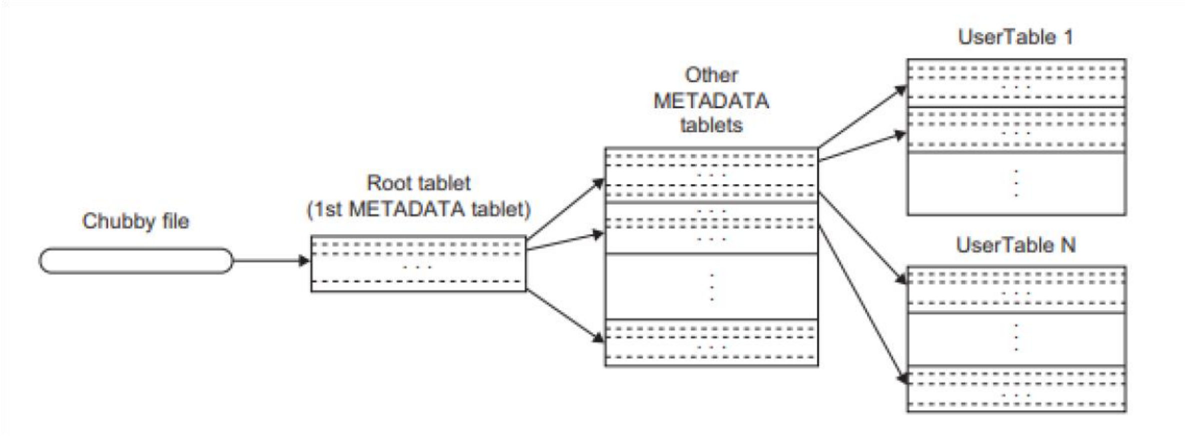


Figure 4.10: Tablet location hierarchy in using the BigTable

- The METADATA table stores the location of a tablet under a row key that is an encoding of the tablet's table identifier and its end row.
- BigTable includes many optimizations and fault-tolerant features. Chubby can guarantee the availability of the file for finding the root tablet.
- The BigTable master can quickly scan the tablet servers to determine the status of all nodes.
- Tablet servers use compaction to store data efficiently.
- Shared logs are used for logging the operations of multiple tablets so as to reduce the log space as well as keep the system consistent.

•

Module 5 - Security in Cloud computing

(I) Security overview

Cloud computing security or, more simply, **cloud security** refers to a broad set of policies, technologies, applications, and controls utilized to protect virtualized IP, data, applications, services, and the associated infrastructure of cloud computing.

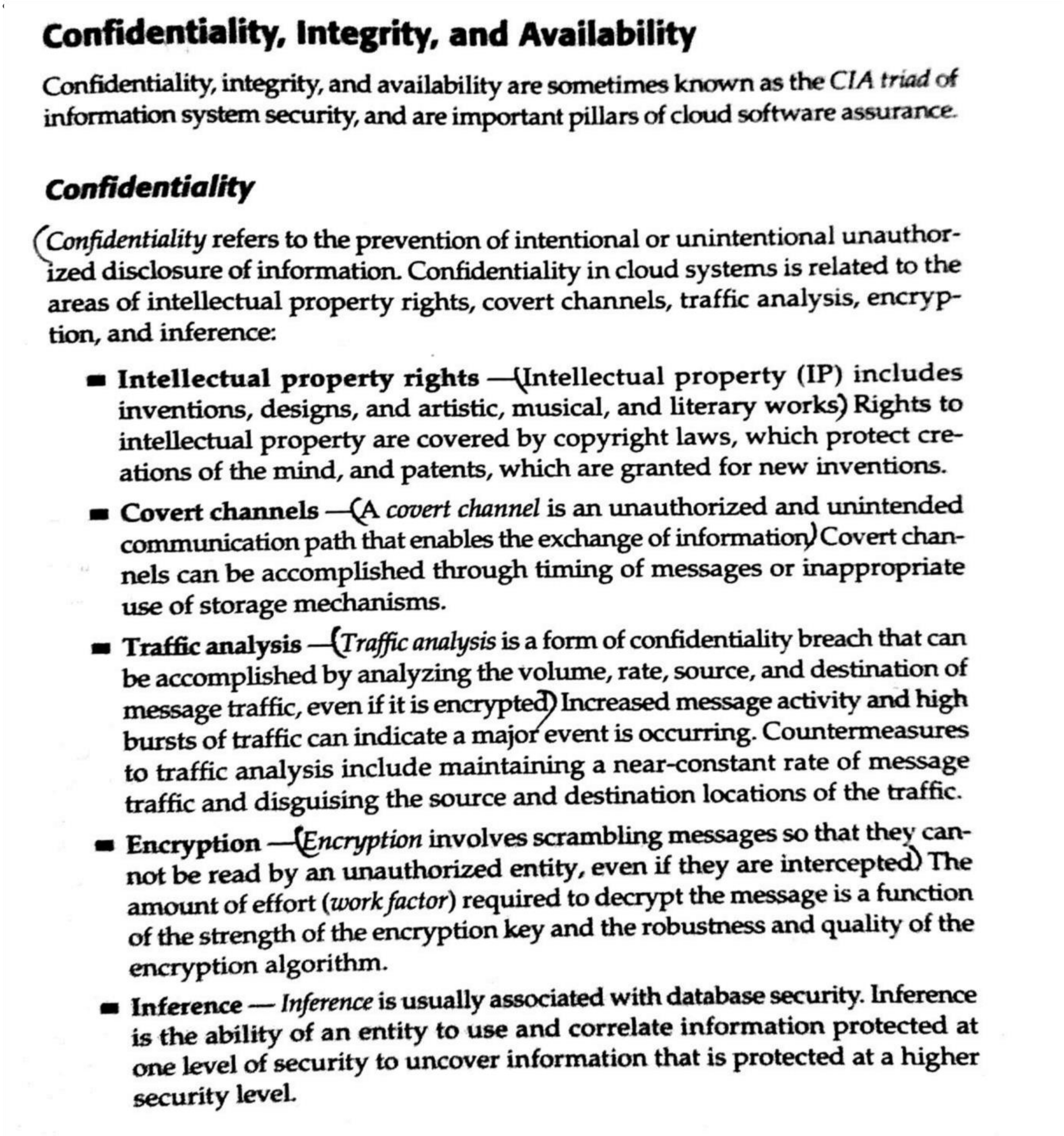
Cloud computing and storage provides users with capabilities to store and process their data in thirdparty data centers. Organizations use the cloud in a variety of different service models (with acronyms such as SaaS, PaaS, and IaaS) and deployment models (private, public, hybrid, and community).

Security concerns associated with cloud computing fall into two broad categories:

- **security issues faced by cloud providers** (organizations providing software-, platform-, or infrastructure-as-a-service via the cloud)
- **security issues faced by their customers** (companies or organizations who host applications or store data on the cloud).

The responsibility is shared, however. The provider must ensure that their infrastructure is secure and that their clients’ data and applications are protected, while the user must take measures to fortify their application and use strong passwords and authentication measures.

Dimensions of cloud security:



Confidentiality, Integrity, and Availability

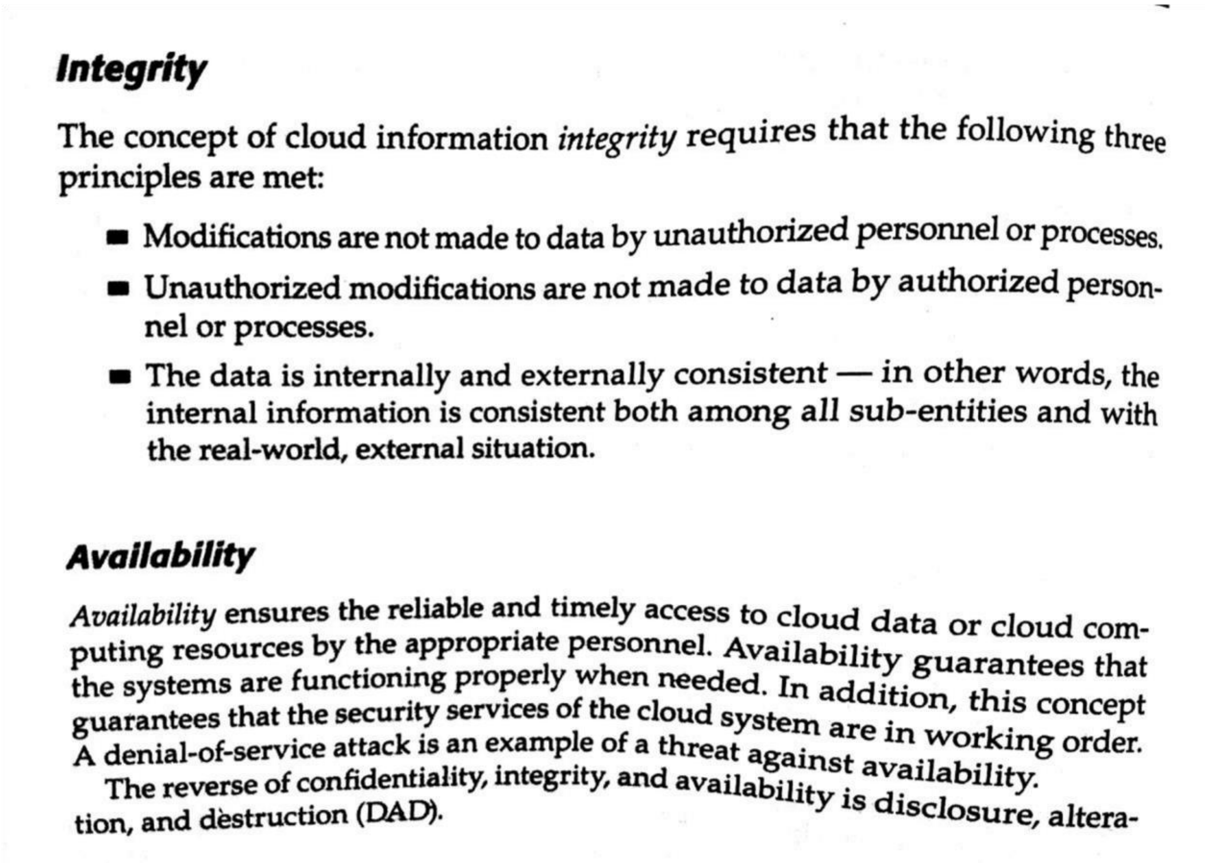
Confidentiality, integrity, and availability are sometimes known as the *CIA triad* of information system security, and are important pillars of cloud software assurance.

Confidentiality

Confidentiality refers to the prevention of intentional or unintentional unauthorized disclosure of information. Confidentiality in cloud systems is related to the areas of intellectual property rights, covert channels, traffic analysis, encryption, and inference:

- **Intellectual property rights** — Intellectual property (IP) includes inventions, designs, and artistic, musical, and literary works. Rights to intellectual property are covered by copyright laws, which protect creations of the mind, and patents, which are granted for new inventions.
- **Covert channels** — A *covert channel* is an unauthorized and unintended communication path that enables the exchange of information. Covert channels can be accomplished through timing of messages or inappropriate use of storage mechanisms.
- **Traffic analysis** — *Traffic analysis* is a form of confidentiality breach that can be accomplished by analyzing the volume, rate, source, and destination of message traffic, even if it is encrypted. Increased message activity and high bursts of traffic can indicate a major event is occurring. Countermeasures to traffic analysis include maintaining a near-constant rate of message traffic and disguising the source and destination locations of the traffic.
- **Encryption** — *Encryption* involves scrambling messages so that they cannot be read by an unauthorized entity, even if they are intercepted. The amount of effort (*work factor*) required to decrypt the message is a function of the strength of the encryption key and the robustness and quality of the encryption algorithm.
- **Inference** — *Inference* is usually associated with database security. Inference is the ability of an entity to use and correlate information protected at one level of security to uncover information that is protected at a higher security level.

•



Integrity

The concept of cloud information *integrity* requires that the following three principles are met:

- Modifications are not made to data by unauthorized personnel or processes.
- Unauthorized modifications are not made to data by authorized personnel or processes.
- The data is internally and externally consistent — in other words, the internal information is consistent both among all sub-entities and with the real-world, external situation.

Availability

Availability ensures the reliable and timely access to cloud data or cloud computing resources by the appropriate personnel. Availability guarantees that the systems are functioning properly when needed. In addition, this concept guarantees that the security services of the cloud system are in working order. A denial-of-service attack is an example of a threat against availability.

The reverse of confidentiality, integrity, and availability is disclosure, alteration, and destruction (DAD).

(II) Security as service in cloud:

Cloud computing security is the set of control-based technologies and policies designed to adhere to regulatory compliance rules and protect information, data applications and infrastructure associated with cloud computing use.

Cloud computing security processes should address the security controls the cloud provider will incorporate to maintain the customer's data security, privacy and compliance with necessary regulations.

The following are 10 security-as-a-service categories are :

1. **Identity and Access Management** should provide controls for assured identities and access management. Identity and access management includes people, processes and systems that are used to manage access to enterprise resources by assuring the identity of an entity is verified and is granted the correct level of access based on this assured identity. Audit logs of activity such as successful and failed authentication and access attempts should be kept by the application/solution.

-

2. **Data Loss Prevention** is the monitoring, protecting and verifying the security of data at rest, in motion and in use in the cloud and on-premises. Data loss prevention services offer protection of data usually by running as some sort of client on desktops/servers and running rules around what can be done. Within the cloud, data loss prevention services could be offered as something that is provided as part of the build, such that all servers built for that client get the data loss prevention software installed with an agreed set of rules deployed.

3. **Web Security** is real-time protection offered either on-premise through software/appliance installation or via the cloud by proxying or redirecting web traffic to the cloud provider. This provides an added layer of protection on top of things like AV to prevent malware from entering the enterprise via activities such as web browsing. Policy rules around the types of web access and the times this is acceptable also can be enforced via these web security technologies.

4. **E-mail Security** should provide control over inbound and outbound e-mail, thereby protecting the organization from phishing and malicious attachments, enforcing corporate policies such as acceptable use and spam and providing business continuity options. The solution should allow for policy-based encryption of e-mails as well as integrating with various e-mail server offerings. Digital signatures enabling identification and non-repudiation are features of many cloud e-mail security solutions.

5. **Security Assessments** are third-party audits of cloud services or assessments of on-premises systems based on industry standards. Traditional security assessments for infrastructure and applications and compliance audits are well defined and supported by multiple standards such as NIST, ISO and CIS. A relatively mature toolset exists, and a number of tools have been implemented using the SaaS delivery model. In the SaaS delivery model, subscribers get the typical benefits of this cloud computing variant elasticity, negligible setup time, low administration overhead and pay-peruse with low initial investments.

6. **Intrusion Management** is the process of using pattern recognition to detect and react to statistically unusual events. This may include reconfiguring system components in real time to stop/prevent an intrusion. The methods of intrusion detection, prevention and response in physical environments are mature; however, the growth of virtualization and massive multi-tenancy is creating new targets for intrusion and raises many questions about the implementation of the same protection in cloud environments.

7. **Security Information and Event Management systems** accept log and event information. This information is then correlated and analyzed to provide real-time reporting and alerting on incidents/events that may require intervention. The logs are likely to be kept in a manner that prevents tampering to enable their use as evidence in any investigations.

8. **Encryption systems** typically consist of algorithms that are computationally difficult or infeasible to break, along with the processes and procedures to manage encryption and decryption, hashing, digital signatures, certificate generation and renewal and key exchange.

9. **Business Continuity and Disaster Recovery** are the measures designed and implemented to ensure operational resiliency in the event of any service interruptions. Business continuity and disaster recovery provides flexible and reliable failover for required services in the

-

event of any service interruptions, including those caused by natural or man-made disasters or disruptions. Cloud-centric business continuity and disaster recovery makes use of the cloud's flexibility to minimize cost and maximize benefits.

10. Network Security consists of security services that allocate access, distribute, monitor and protect the underlying resource services. Architecturally, network security provides services that address security controls at the network in aggregate or specifically addressed at the individual network of each underlying resource.

(III) Security Governance

Cloud security governance refers to the management model that facilitates effective and efficient security management and operations in the cloud environment so that an enterprise's business targets are achieved.

This model incorporates a hierarchy of executive mandates, performance expectations, operational practices, structures, and metrics that, when implemented, result in the optimization of business value for an enterprise.

The following represents key objectives to pursue in establishing a governance model for security in the cloud:

1. **Strategic Alignment**
   Enterprises should mandate that security investments, services, and projects in the cloud are executed to achieve established business goals (e.g., market competitiveness, financial, or operational performance).

2. **Value Delivery**

   Enterprises should define, operationalize, and maintain an appropriate security function/organization with appropriate strategic and tactical representation, and charged with the responsibility to maximize the business value (Key Goal Indicators, ROI) from the pursuit of security initiatives in the cloud.

3. **Risk Mitigation**

   Security initiatives in the cloud should be subject to measurements that gauge effectiveness in

   mitigating risk to the enterprise (Key Risk Indicators). These initiatives should also yield results that progressively demonstrate a reduction in these risks over time.

-

4. **Effective Use of Resources**

It is important for enterprises to establish a practical operating model for managing and performing security operations in the cloud, including the proper definition and operationalization of due processes, the institution of appropriate roles and responsibilities, and use of relevant tools for overall efficiency and effectiveness.

5. **Sustained Performance**

Security initiatives in the cloud should be measurable in terms of performance, value and risk to the enterprise (Key Performance Indicators, Key Risk Indicators), and yield results that demonstrate attainment of desired targets (Key Goal Indicators) over time.

Cloud security governance Deals with :

- Risk management
- Security monitoring

Risk management In Cloud :

- Risk management is one of the most important jobs for a project manager.
- Risk management involves anticipating risk that might affect the project schedule or the quality of the software being developed, and then taking action to avoid these risks.
- Risk is something that prefer not to happen. Risks may threaten the project, the software that is being developed, or the organization.
- The outline of the process of risk management is illustrated in following fig

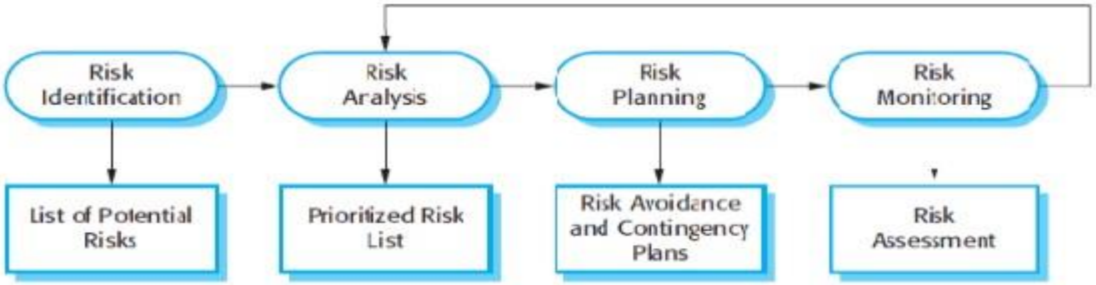


Fig 1. Risk management process

•

- Risk identification
- Risk analysis and evaluation

•Selection of counter measures

- Deployment of suitable counter measures
- Continuous monitoring to assess effectiveness of the solution

Risk management involves following several stages

- Risks identification Risk identification can be done by identifying the known and predictable risk.
- Risk analysis Risk can be analyse by assessing consequences of problem associated with risk.
- Risk planning Making plan to address the risk, either by avoiding it or minimizing its effect on the project.
- Risk monitoring Regularly assessing the risk and plans for risk mitigation.

The risk management process is an iterative process that continues throughout the project. Once initial risk management plan is drawn up, it will help to monitor the situation to detect the emerging risks. As more information about the risks become available, then it will be easier to analyse and decide if the risk priority has changed. Referring it you may then have to change your plans for risk avoidance and contingency management.

1. Risk Identification

- Risk identification is the first stage of the risk management process.
- It is concerned with identifying the risks that could pose a major threat to the software engineering process, the software being developed, or the development organization.
- Risk identification is done by team or sometimes by the project manager.
- The risk item can be identified using following and predictable components.
- After preparing a risk item checklist, a questionnaire is prepared. These set of question should be answered and based on these answer the impact or seriousness of particular risk item can be judged.
- The set of risk components and drivers list is prepared along with their probability of occurrence. Then their impact on the project can be analysed.

2. Risk Analysis

- There are two way by which risk can be rated

1. Probability of the risk is real
2. Consequences of problems associated with the risk

-

- The project planner, technical staff, project manager perform following steps to perform for risk analysis ○ Establish a scale that indicates the probability of risk being real. ○ Enlist the consequences of the risk. ○ Estimate the impact of the risk on the project and product.
    - Maintain the overall accuracy of the risk projection in order to have clear understanding of the software that is to be built.
    - This steps helps to prioritize the risk. And finally, risk table will be built.

3. Risk planning

- The risk planning process considers each of the key risks that have been identified, and develop strategies to manage these risks.
- For each risks, they have to think of actions that they might take to minimize the disruption to the project if the problem identified in the risk occurs.
- There is no simple process that can be followed for contingency planning. It relies on the judgment and experience of the project manager.
- Possible risk management strategies fall into three categories.

1. Avoidance strategies

Using these strategies mean that the probability that the risk will be arise will be reduced. Example of a risk avoidance strategy is the strategy dealing with defective components.

2. Minimization strategies

Using these strategies means that the impact of the risk will be reduced. Example of risk minimization strategy is the strategy for staff illness (recognize team so they can understand each other's job).

3. Contingency strategies

Using these strategies means that team are prepared for the worst and have a strategy in place to deal with it.

Example of contingency strategy is the strategy for organizational financial problem

- Finally, team should have strategies in place to cope with the risk if it arises. These should reduce the overall impact of a risk on the project or product.

4. Risk monitoring

- Risk monitoring is the process of checking that your assumptions about the product, process and business risk have not changed.

- The objective of risk monitoring is

3. To check whether the predicted risks really occur or not.
4. To ensure the step defined to avoid the risk are applied properly or not.
5. To gather the information which can be useful for analysing the risk.

-

Finally, RMMM document is created, in which all the risk analysis activities are described. Sometimes project manager includes this document as a part of overall project plan.

CLOUD SECURITY MONITORING

Monitoring is a critical component of cloud security and management.

Typically relying on automated solutions, cloud security monitoring supervises virtual and physical servers to continuously assess and measure data, application, or infrastructure behaviors for potential security threats. This assures that the cloud infrastructure and platform function optimally while minimizing the risk of costly data breaches.

BENEFITS OF CLOUD SECURITY MONITORING

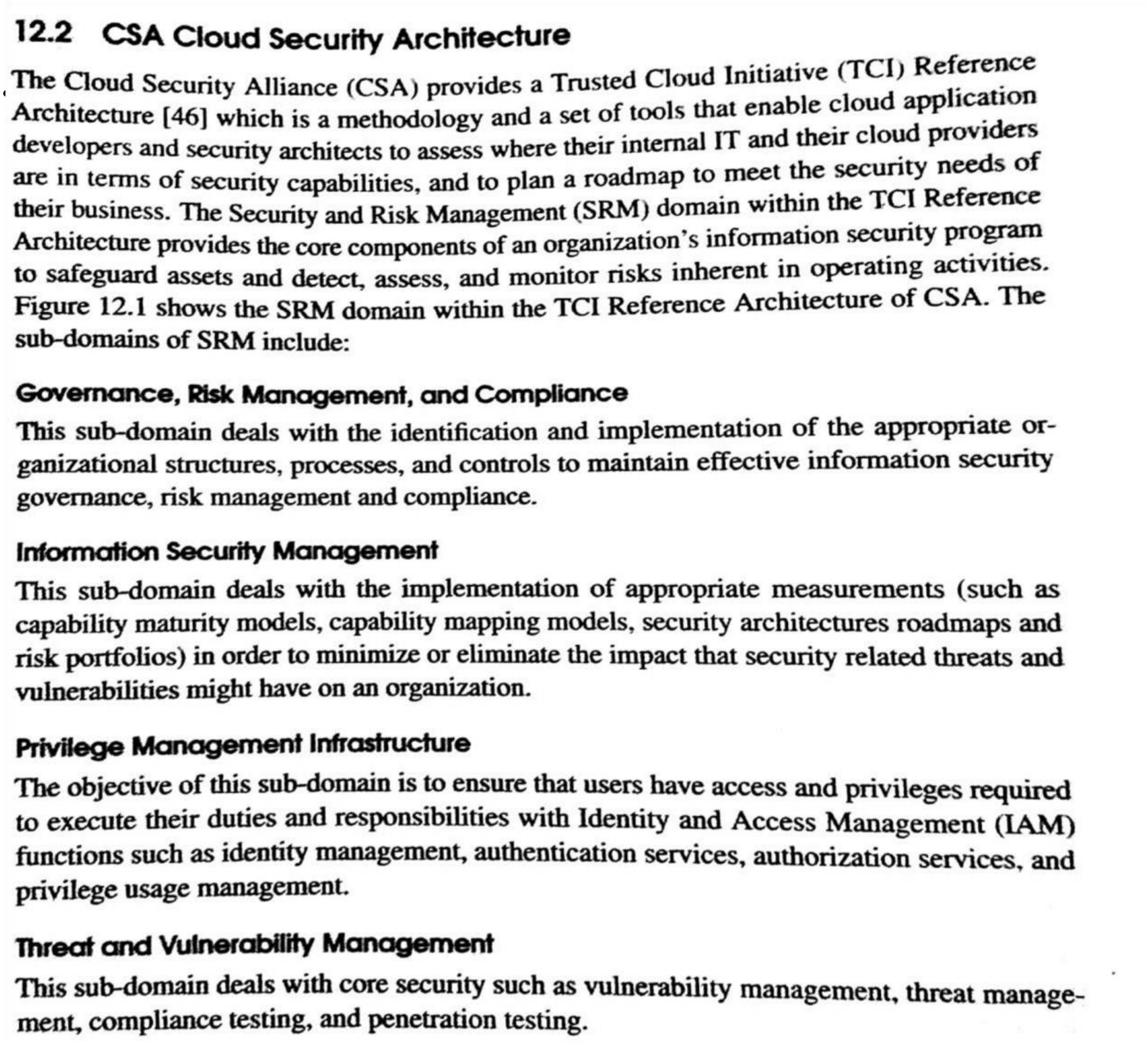
Cloud monitoring provides an easier way to identify patterns and pinpoint potential security vulnerabilities in cloud infrastructure. As there’s a general perception of a loss of control when valuable data is stored in the cloud, effective cloud monitoring can put companies more at ease with making use of the cloud for transferring and storing data.

When customer data is stored in the cloud, cloud monitoring can prevent loss of business and frustrations for customers by ensuring that their personal data is safe. The use of web services can increase security risks, yet cloud computing offers many benefits for businesses, from accessibility to a better customer experience. Cloud monitoring is one initiative that enables companies to find the balance between the ability to mitigate risks and taking advantage of the benefits of the cloud – and it should do so without hindering business processes.

HOW CLOUD SECURITY MONITORING WORKS

There are several approaches to cloud security monitoring. Cloud monitoring can be done in the cloud platform itself, on premises using an enterprise’s existing security management tools, or via a third party service provider. Some of the key capabilities of cloud security monitoring software include:

- **Scalability:** tools must be able to monitor large volumes of data across many distributed locations



12.2 CSA Cloud Security Architecture

The Cloud Security Alliance (CSA) provides a Trusted Cloud Initiative (TCI) Reference Architecture [46] which is a methodology and a set of tools that enable cloud application developers and security architects to assess where their internal IT and their cloud providers are in terms of security capabilities, and to plan a roadmap to meet the security needs of their business. The Security and Risk Management (SRM) domain within the TCI Reference Architecture provides the core components of an organization's information security program to safeguard assets and detect, assess, and monitor risks inherent in operating activities. Figure 12.1 shows the SRM domain within the TCI Reference Architecture of CSA. The sub-domains of SRM include:

Governance, Risk Management, and Compliance

This sub-domain deals with the identification and implementation of the appropriate organizational structures, processes, and controls to maintain effective information security governance, risk management and compliance.

Information Security Management

This sub-domain deals with the implementation of appropriate measurements (such as capability maturity models, capability mapping models, security architectures roadmaps and risk portfolios) in order to minimize or eliminate the impact that security related threats and vulnerabilities might have on an organization.

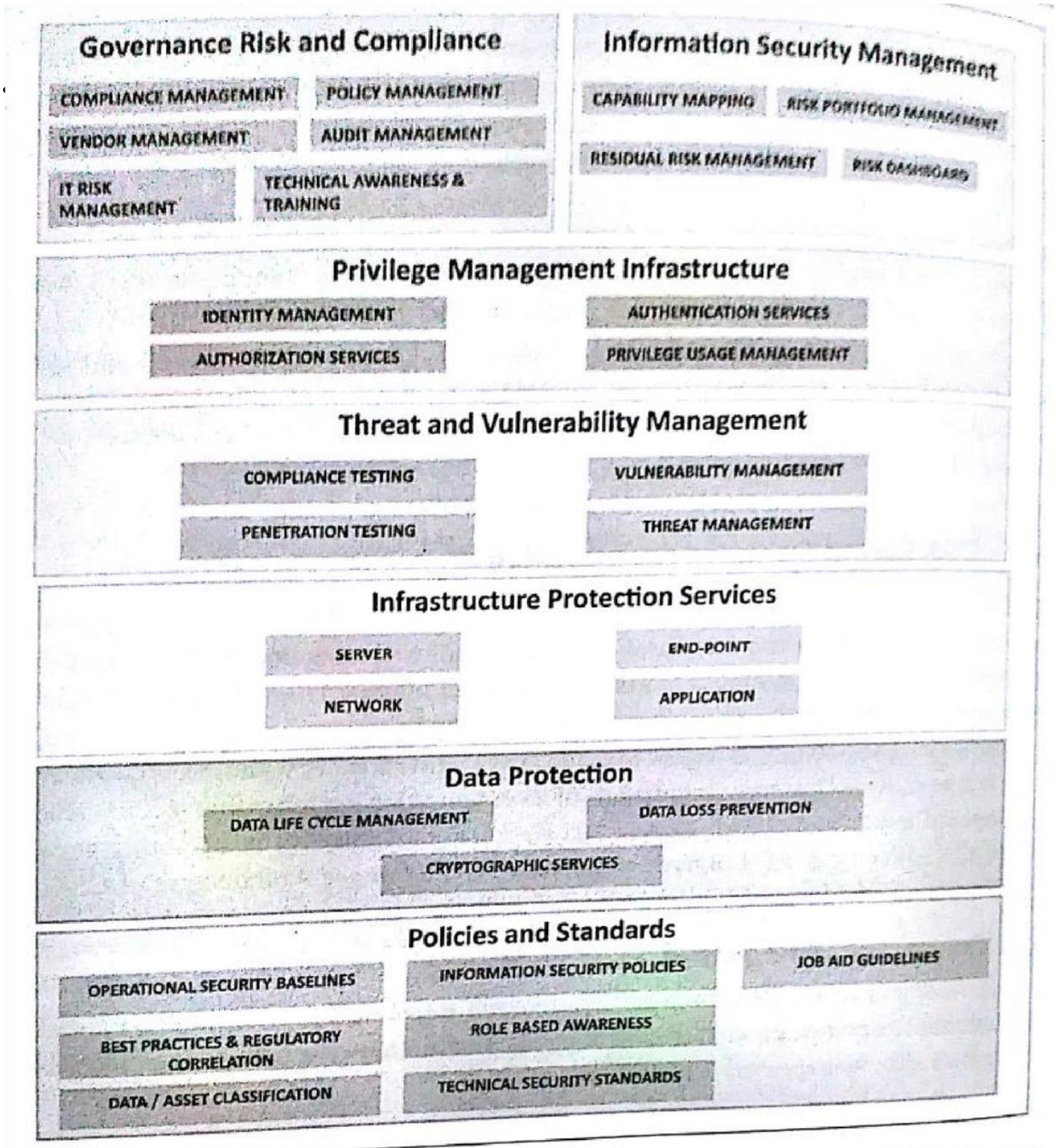
Privilege Management Infrastructure

The objective of this sub-domain is to ensure that users have access and privileges required to execute their duties and responsibilities with Identity and Access Management (IAM) functions such as identity management, authentication services, authorization services, and privilege usage management.

Threat and Vulnerability Management

This sub-domain deals with core security such as vulnerability management, threat management, compliance testing, and penetration testing.

- **Visibility:** the more visibility into application, user, and file behavior that a cloud monitoring solution provides, the better it can identify potential attacks or compromises
- **Timeliness:** the best cloud security monitoring solutions will provide constant monitoring, ensuring that new or modified files are scanned in real time
- **Integration:** monitoring tools must integrate with a wide range of cloud storage providers to ensure full monitoring of an organization's cloud usage
- **Auditing and Reporting:** cloud monitoring software should provide auditing and reporting capabilities to manage compliance requirements for cloud security



gure 12.1: Security and Risk Management (SRM) domain within the TCI Reference rchitecture of CSA [46]

ıfrastructure Protection Services

his objective of this sub-domain is to secure Server, End-Point, Network and Application ıyers.

)ata Protection

:his sub-domain deals with data lifecycle management, data leakage prevention, intellectual ıroperty protection with digital rights management, and cryptographic services such as key nanagement and PKI/symmetric encryption.

Cloud Security challenges :-

Key Security Challanges for Cloud applications include :-

(i) Authentication

(ii) Authorization

(iii) Security of data at rest

(iv) Security of data at motion

(v) data integrity

(vi) Auditing.

(i) Authentication :-

Authentication refers to digitally conforming the identity of the entity rqsting access to some protected information.

(ii) Authorization

Authorization refers to digitally specifying the access right to the protected resources using access policies.

(iii) Security of data at rest

- Data at rest means, that is stored in the data base in the form of tables/records, files on Server or Raw data stored in distributed storage or SAN, etc.

(iv) Security of data in motion:-

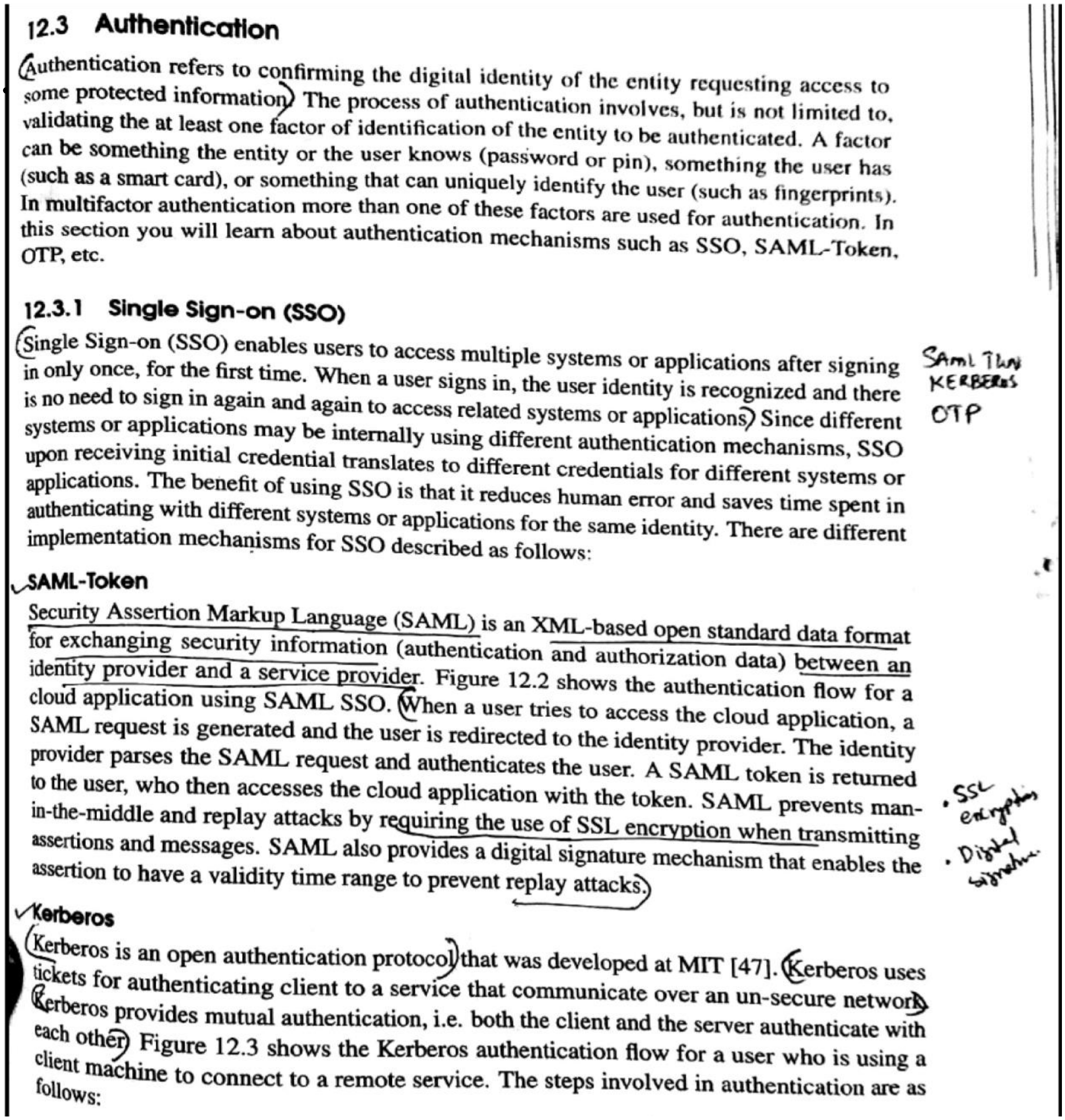
ie when the data flows between a client server over a potentially insecure network.

(v) Data integrity

Data integrity ensures that the data is not altered in an unauthorized manner after it created, transmitted or stored.

(vi) Auditing

For cloud applications, appropriate auditing mech are required to get visibility into the application, data access and actions performed by the application users, including mobile users & devic such as wireless laptop & smart phones.



12.3 Authentication

Authentication refers to confirming the digital identity of the entity requesting access to some protected information. The process of authentication involves, but is not limited to, validating the at least one factor of identification of the entity to be authenticated. A factor can be something the entity or the user knows (password or pin), something the user has (such as a smart card), or something that can uniquely identify the user (such as fingerprints). In multifactor authentication more than one of these factors are used for authentication. In this section you will learn about authentication mechanisms such as SSO, SAML-Token, OTP, etc.

12.3.1 Single Sign-on (SSO)

Single Sign-on (SSO) enables users to access multiple systems or applications after signing in only once, for the first time. When a user signs in, the user identity is recognized and there is no need to sign in again and again to access related systems or applications. Since different systems or applications may be internally using different authentication mechanisms, SSO upon receiving initial credential translates to different credentials for different systems or applications. The benefit of using SSO is that it reduces human error and saves time spent in authenticating with different systems or applications for the same identity. There are different implementation mechanisms for SSO described as follows:

SAML Tkn
KERBEROS
OTP

SAML-Token

Security Assertion Markup Language (SAML) is an XML-based open standard data format for exchanging security information (authentication and authorization data) between an identity provider and a service provider. Figure 12.2 shows the authentication flow for a cloud application using SAML SSO. When a user tries to access the cloud application, a SAML request is generated and the user is redirected to the identity provider. The identity provider parses the SAML request and authenticates the user. A SAML token is returned to the user, who then accesses the cloud application with the token. SAML prevents man-in-the-middle and replay attacks by requiring the use of SSL encryption when transmitting assertions and messages. SAML also provides a digital signature mechanism that enables the assertion to have a validity time range to prevent replay attacks.

- SSL encryption
- Digital signature

✓**Kerberos**

Kerberos is an open authentication protocol that was developed at MIT [47]. Kerberos uses tickets for authenticating client to a service that communicate over an un-secure network. Kerberos provides mutual authentication, i.e. both the client and the server authenticate with each other. Figure 12.3 shows the Kerberos authentication flow for a user who is using a client machine to connect to a remote service. The steps involved in authentication are as follows:

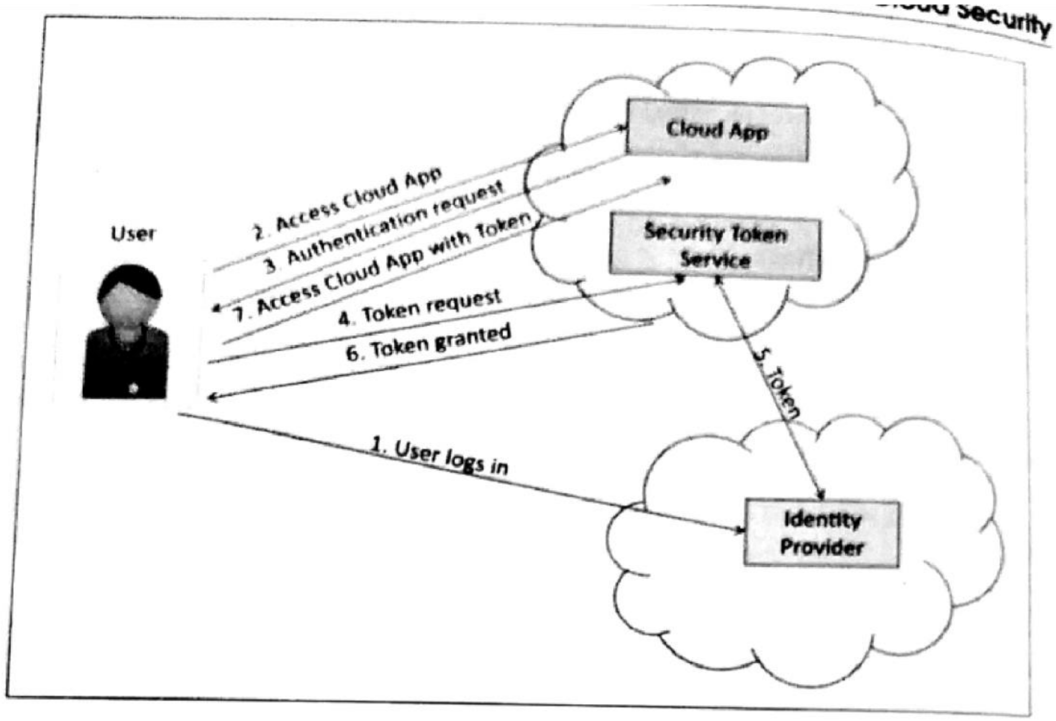


Figure 12.2: SAML-token based SSO authentication

- The client authenticates itself to the Authentication Server (AS) that resides on the Key Distribution Center (KDC). The client does not need to send the user password to the AS. Instead the client sends a clear-text message to the AS containing the user ID to the AS requesting to authenticate the user to the remote service.
- The AS checks if the client is in the database and generates a Client/TGS Session Key that will be used by the client and the remote service. The AS encrypts the session key with the user's password. The KDC also prepares the Ticket Granting Ticket (TGT) (which includes the client ID, client network address, ticket validity period, and the session key) and encrypts it using the secret key of the TGS. KDC then sends both the session key and TGT to the client. On receiving the session key and TGT, the client decrypts the Client/TGS Session Key using its password and extracts the session key. The client cannot decrypt the TGT as it does not know the secret key of the TGS.
- The client encrypts the client ID and current time using the session key to prepare an authenticator. The client then sends the authenticator and the TGT that it received from the KDC to the TGS.
- On receiving the authenticator and TGT from the client, the TGS decrypts the TGT using its own secret key and retrieves the session key. The TGS then uses the session key to decrypt the authenticator and extracts the client ID and time. The TGS then sends two pieces of data to the client. The first piece contains the client-server ticket (which includes the client ID, client network address, validity period and Client/Server Session Key) encrypted using the service's secret key. The second piece of data contains the Client/Server Session Key encrypted with the Client/TGS Session Key.

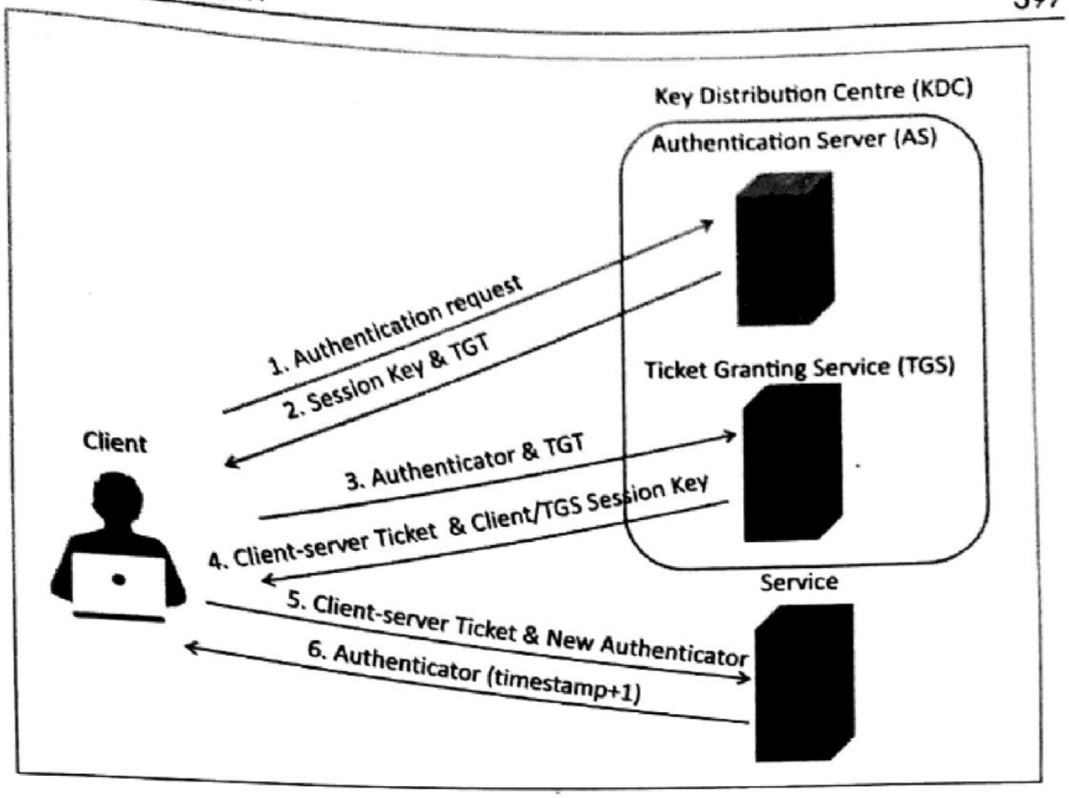

Key Distribution Centre (KDC)
Authentication Server (AS)
Ticket Granting Service (TGS)
Service
Client
1. Authentication request
2. Session Key & TGT
3. Authenticator & TGT
4. Client-server Ticket & Client/TGS Session Key
5. Client-server Ticket & New Authenticator
6. Authenticator (timestamp+1)

Figure 12.3: Kerberos authentication flow

- On receiving the client-server ticket and Client/TGS Session Key from the TGS, the client has enough information to authenticate itself to the remote service. The client then sends two pieces of data to the remote service. The first piece contains the client-server ticket which is encrypted using the service's secret key (that it received from the TGS). The second piece of data contains a new authenticator which includes the client ID, timestamp and is encrypted using Client/Server Session Key.
- On receiving the client-server ticket and new authenticator from the client, the remote service decrypts the client-server ticket using its own secret key and retrieves the Client/Server Session Key. Using the sessions key, remote service decrypts the authenticator. At this point the true identity of the client is confirmed to the remote service and it responds to the client with the timestamp found in client's authenticator plus 1, encrypted using the Client/Server Session Key.
- On receiving the new authenticator from the remote service, the client decrypts it using the Client/Server Session Key and checks the timestamp. If the timestamp is correctly updated, the client can trust the server and start issuing service requests to the server.

One Time Password (OTP)

One time password is another authentication mechanism that uses passwords which are valid for single use only for a single transaction or session. Authentication mechanism based on OTP tokens are more secure because they are not vulnerable to replay attacks. Text messaging (SMS) is the most common delivery mode for OTP tokens. The most common approach for generating OTP tokens is time synchronization. Time-based OTP algorithm (TOTP) is a popular time synchronization based algorithm for generating OTPs [48].

12.4 Authorization

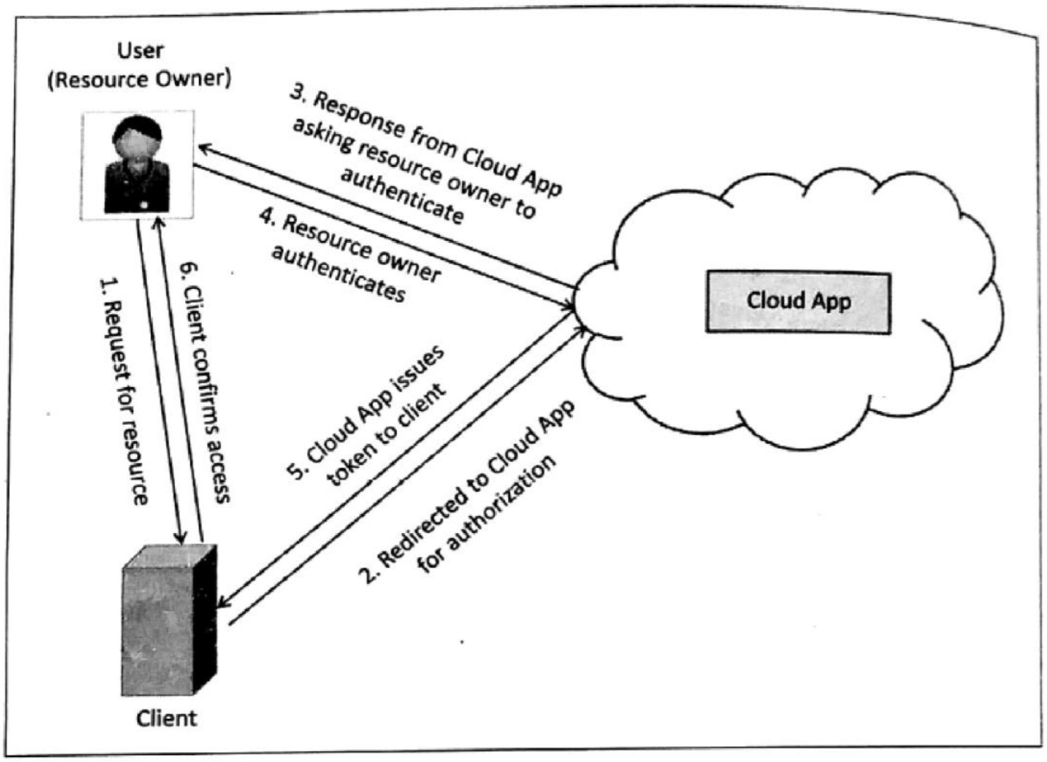

User
(Resource Owner)
3. Response from Cloud App asking resource owner to authenticate
4. Resource owner authenticates
Cloud App
1. Request for resource
6. Client confirms access
5. Cloud App issues token to client
2. Redirected to Cloud App for authorization
Client

Figure 12.4: OAuth authorization flow

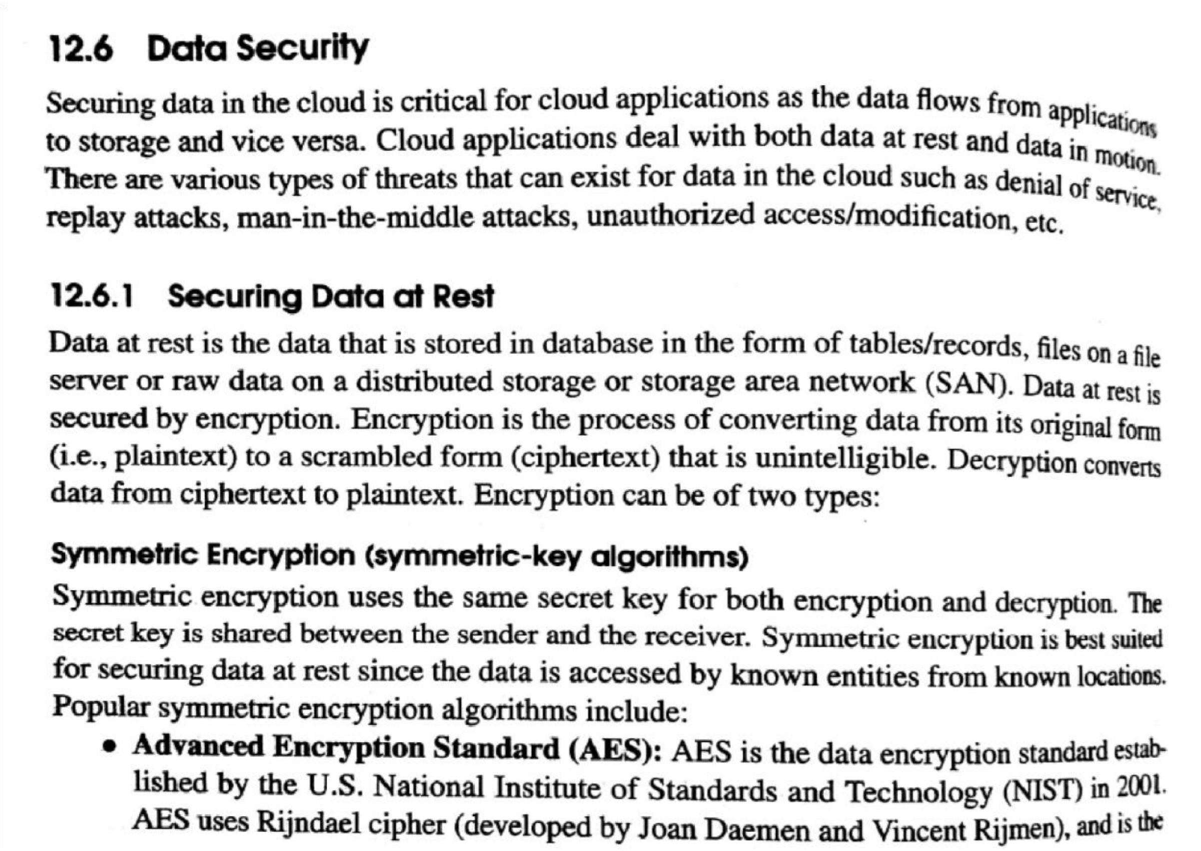
Authorization refers to specifying the access rights to the protected resources using access policies.

OAuth

OAuth is an open standard for authorization that allows resource owners to share their private resources stored on one site with another site without handing out the credentials [49, 50]. OAuth 1.0 protocol was published as an RFC in 2010 and the OAuth 2.0 framework was published in 2012. OAuth 2.0 is not backward compatible with OAuth 1.0. In the OAuth model, an application (which is not the resource owner) requests access to resources controlled by the resource owner (but hosted by the server). The resource owner grants permission to access the resources in the form of a token and matching shared-secret. Tokens make it unnecessary for the resource owner to share its credentials with the application. Tokens can be issued with a restricted scope and limited lifetime, and revoked independently. Figure 12.4 shows the OAuth authorization flow.

Let us look at the an example of an OAuth client. Box 12.1 shows the Python code for an OAuth client that provides methods for fetching request token, fetching access token, authorizing token and accessing resources [51].

•



12.6 Data Security

Securing data in the cloud is critical for cloud applications as the data flows from applications to storage and vice versa. Cloud applications deal with both data at rest and data in motion. There are various types of threats that can exist for data in the cloud such as denial of service, replay attacks, man-in-the-middle attacks, unauthorized access/modification, etc.

12.6.1 Securing Data at Rest

Data at rest is the data that is stored in database in the form of tables/records, files on a file server or raw data on a distributed storage or storage area network (SAN). Data at rest is secured by encryption. Encryption is the process of converting data from its original form (i.e., plaintext) to a scrambled form (ciphertext) that is unintelligible. Decryption converts data from ciphertext to plaintext. Encryption can be of two types:

Symmetric Encryption (symmetric-key algorithms)

Symmetric encryption uses the same secret key for both encryption and decryption. The secret key is shared between the sender and the receiver. Symmetric encryption is best suited for securing data at rest since the data is accessed by known entities from known locations. Popular symmetric encryption algorithms include:

- **Advanced Encryption Standard (AES):** AES is the data encryption standard established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES uses Rijndael cipher (developed by Joan Daemen and Vincent Rijmen), and is the

Virtual Machine Security

Virtualization security is the collective measures, procedures and processes that ensure the protection of a virtualization infrastructure / environment.

It addresses the security issues faced by the components of a virtualization environment and methods through which it can be mitigated or prevented. Compromising the hypervisor could give attackers access to all virtual machines controlled by it and possibly the host, which makes the hypervisor a compelling target.

Typically, virtualization security may include processes such as:

- Implementation of security controls and procedures granularly at each virtual machine.
- Securing virtual machines, virtual network and other virtual appliance with attacks and vulnerabilities surfaced from the underlying physical device.
- Ensuring control and authority over each virtual machine.
- Creation and implementation of security policy across the infrastructure / environment

-

Unauthorized communication between guests is a violation of the isolation principle, but can potentially take place through shared memory.

Like physical machines, VMs are vulnerable to theft and denial of service attacks.

The contents of the virtual disk for each virtual machine are usually stored as a file, which can be run by hypervisors on other machines, allowing attackers to copy the virtual disk and gain unrestricted access to the digital contents of the virtual machine.

Virtual machines security issues:

1 Mobility

Virtual machines are inherently not physical, which means their theft can take place without physical theft of the host machine. The contents of the virtual disk for each VM are stored as a file by most hypervisors, which allows VMs to be copied and run from other physical machines.

2 Hypervisor Intrusion

The hypervisor provides the abstraction and resource allocation between the host and guests. Attackers ultimate goal is to compromise the hypervisor to gain with the ability *to execute arbitrary code on the host with the privileges of the [hypervisor] process*

3 Hypervisor Modification

It does not matter how secure the original hypervisor is if it can be externally modified to use the attacker software. One attack of this form is known as Virtual Machine Based Root Kits (VMBR)

. **4 Communication**

In which attackers use one VM to access or control other VMs on the same hypervisor.

•

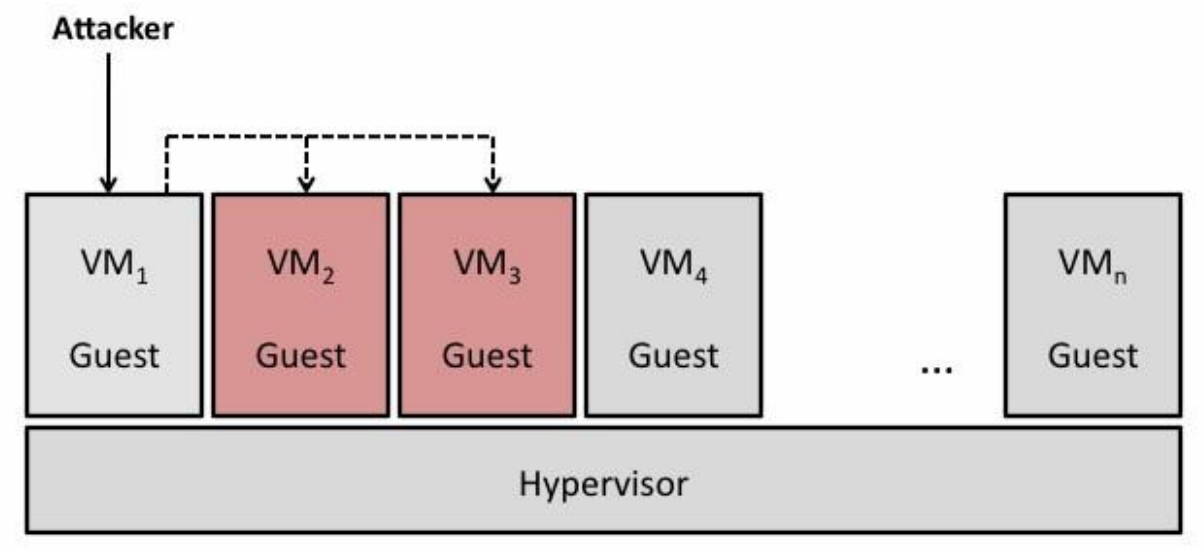
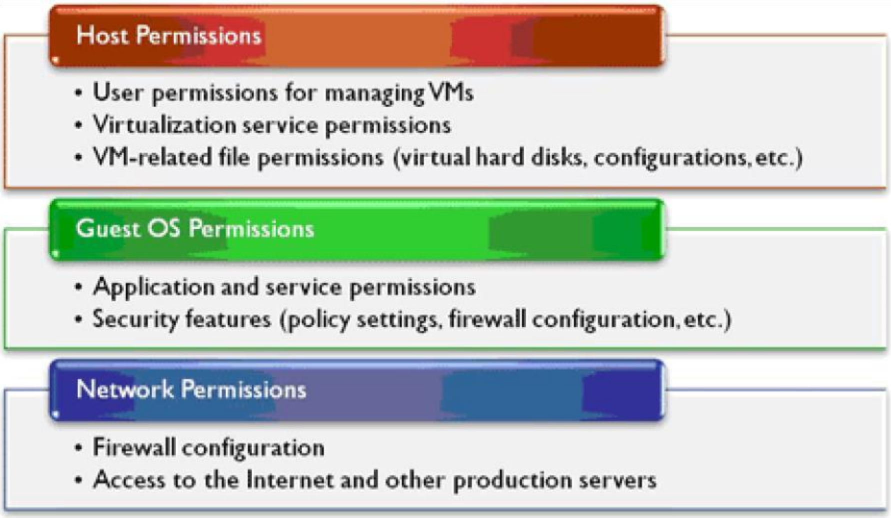


Figure .VM Communication Attack on VM2 and VM3

5 Denial of Service

DoS attacks are a threat to all servers, however an improperly configured hypervisor can allow a single VM to consume all resources, thus starving any other VM running on the same physical machine. DoS attacks make network hosts unable to function since critical processes do not have the hardware resources to execute in a timely manner.

<u>Types of permissions to consider when securing virtualization</u> :



Host Permissions

- User permissions for managing VMs
- Virtualization service permissions
- VM-related file permissions (virtual hard disks, configurations, etc.)

Guest OS Permissions

- Application and service permissions
- Security features (policy settings, firewall configuration, etc.)

Network Permissions

- Firewall configuration
- Access to the Internet and other production servers

-

- • On virtualization hosts only certain staff members should be able to start, stop and reconfigure VMs. It's also important to configure virtual applications and services using limited system accounts. Finally, you should take into account the real requirements for VM configurations.

•

MODULE 6

USING CLOUD

➢ CENTRALIZING EMAIL COMMUNICATIONS

- Pre-cloud computing, your email access was via a single computer, which also stored all your email messages.
- For this purpose, you probably used a program like Microsoft Outlook or Outlook Express, installed on your home computer.
- If you wanted to check your home email from work, it took a bit of juggling and perhaps the use of your ISP’s email access web page. That web page was never in sync with the messages on your home PC, of course, which is just the start of the problems with trying to communicate in this fashion.
- A better approach is to use a web-based email service, such as Google’s Gmail (mail.google.com), Microsoft’s Windows Live Hotmail (mail.live.com), or Yahoo! Mail (mail.yahoo.com).
- These services place your email inbox in the cloud; you can access it from any computer connected to the Internet.
- The messages you receive are stored on the web, as are the messages you send, so nothing depends on a single PC.

-

- The joy of using web-based email is that it doesn’t matter what PC you use, your messages are always where they should be and they’re always in sync.
- It’s easy to check your home email from work, or from anywhere you happen to be—in a coffeehouse, at a hotel, or even in an airport terminal. Use your work PC, your home PC, your notebook PC, or a friend’s PC, it doesn’t matter; your messages are in the cloud, not on any of those PCs.
- You can check your web-based email whether you’re in the office or on the road. Just make sure you’re connected to the Internet, and then open your web browser and log in to the Gmail or Windows Live Hotmail or Yahoo! Mail website.
- Go to your inbox and you’ll find your spouse’s message; reply as necessary and await your spouse’s response. Even if you change locations or computers, your spouse’s message remains in your inbox, and your reply remains in your sent messages folder.

➢ COLLABORATING ON TO-DO LISTS

- A grocery list is just one type of to-do list. If you have a lot of household chores and repairs, it’s likely that you have a larger to-do list for your household.
- Anyone can collaborate on your to-do list by using a web-based word processing application, as we just discussed, or you can use a dedicated web-based planning program.

-

- These applications, such as Zoho Planner (planner.zoho.com) let you create multiple to-do lists on the web, which you and your spouse can both add to from any computer, at any time.
- You can even set email reminders to refresh your memory when a task is due. Add your tasks one at a time, and then mark them off as they're completed.
- If these applications are too advanced for your needs, consider using a simpler web-based to-do list application.
- These applications, such as Remember the Milk (www.rememberthemilk.com) and Ta-da List (www.tadalist.com), operate more like a simple notepad-based list. Some even let you add tasks via email or access the list when you're on the go with your mobile phone.

➢ COLLABORATING ON CONTACT LISTS

- Managing your family's contact list isn't always easy. Yes, you have your most-contacted contacts stored in Microsoft Outlook or some similar program, but that list of names exists only on one computer.
- Some contacts are on your PC, some are on PC, and your lists of work contacts are probably on your work computers. How do you merge and manage all these names—in time to address and mail your cards before the holidays?
- A good solution for managing contacts from multiple family members is to use a web-based program for contact management.

-

- There are few different ways to approach this. First, you can use your web-based email program (Gmail, Yahoo! Mail, and so on) as a contact management program.
- All of these programs let you create and store complete information about your contacts—email address, postal address, phone number, and so forth.
- The only problem with using this approach, however, is that both you and your spouse have to use the same email program and the same email address. So, it might not work for you.
- A more robust and individualized solution is to use a dedicated web-based contact management program.
- Some of these programs, such as MyEvents (www.myevents.com) are targeted at home users and ideal for holiday card lists; other programs, such as Highrise (www.highrisehq.com) will manage your holiday card list and do a lot more.
- These latter programs include the robust customer resource management (CRM) features needed for business and sales force management.

➢ CLOUD COMPUTING FOR THE COMMUNITY

- Cloud computing isn’t just for home users. It has tremendous benefits for the entire community, from neighborhood groups to sports teams to school organizations.

-

- Any time any group of people in the community need to communicate and collaborate, web-based applications are the way to go. This portion, therefore, takes a look at a few typical community uses of cloud computing.

❖ Communicating Across the Community

- One of the key components of any community collaboration is communication. This isn't as easy as it sounds, because many community activities are undertaken by people in their spare time—outside of normal work and home activities.
- Therefore, they might be communicating during office hours on their work computer, after hours on their home computer, or during any spare moment wherever they may happen to be. That makes using traditional desktop email, such as Microsoft Outlook or Windows Mail, problematic.
- The better solution when communication on community issues is to use a web-based email program, such as Gmail (mail.google.com), Microsoft Windows Live Hotmail (mail.live.com), or Yahoo! Mail (mail.yahoo.com).
- These programs can be accessed from any computer connected to the Internet. You use your web browser to send and view email messages hosted on the web.

-

- You can send and receive messages at work, at home, or from wherever you happen to be. Everything you send and receive is stored in the cloud, accessible from anywhere at any time.
- Some community activists go so far as to create a unique web-based email account just for their community communications.

❖ Collaborating on Schedules

- When it comes to coordinating multiple individuals or families in a community activity, you have your work cut out for you. Whether it's a youth sports team, community organization, school event, or some community event, trying to line up who's free and who's not on a given evening takes a lot of effort—unless, that is, you're using web-based scheduling tools.

(i) Sports Team Schedules

- The best way is to use a web-based calendar tool, such as Google Calendar (calendar.google.com), Yahoo! Calendar (calendar.yahoo.com), or CalendarHub (www.calendarhub.com).
- Just create a public calendar and provide the URL to all the team members. After you add all your team activities to the calendar, team members simply have to log in to see what's coming up this week and next.

-

- Also good are dedicated sports team website builders. These sites offer tools designed specifically for sports teams, including home pages complete with schedule, roster, player profiles, box scores, and the like.
- Most of these services even design your site using your team colors and logo.
- There are several of these web-based applications, including eteemz (www.eteemz.com), League Athletics (www.leagueathletics.com), LeagueLineup (www.leaguelineup.com), and TeamSnap (www.teamsnap.com).

(ii) School Schedules

- Web-based calendars are also ideal for keeping track of various school schedules.
- Whether it's homework assignments for a particular class or a schoolwide events schedule, it's easy for a teacher or school to post that schedule on a web-based calendar.
- Make the calendar public (but make sure only authorized personnel can post new events), and then provide the calendar's URL to all students and parents
- . Assuming that all families have Internet access (it helps to check this first), there should be no excuse for missed homework or absence from key events.

-

(iii) Community Group Schedules

- Any community group can benefit from organizing their activities via a webbased calendar. Want to schedule practices for a community theater production? Announce meetings for your local school board? Organize bingo nights for your church?
- Any and all of these group activities can easily be managed online, in the cloud, using a web-based calendar.

(iv) Event Schedules and Management

- You can also use web-based calendars to post dates and schedules for specific public events, such as school plays, or for all events in a given community.
- Although any web-based calendar program can do this job, as well, some event-specific applications are worth noting.
- For example, Zvents (zvents.com) is a web search engine for local events. Upload your event schedule into the Zvents database, and then anyone in your area can find out what's happening in the coming days and weeks.
- Users can also search for events by type, location, and date. Also interesting is the suite of event management software from ServiceU (www. serviceu.com).
- Included in this suite is the EventU application, which offers event, resource, and facility scheduling for organizations small or large.

-

❖ Collaborating on Group Projects and Events

- Community groups often have a lot on their plates. Someone has to schedule the next fundraiser, someone else needs to print up flyers, someone else is in charge of recruiting new members…there's just a lot of stuff to do!
- The best way is with a web-based application— which anyone in the group can access.

(i) Collaborating on To-Do Lists

- Let's start with simple task management, in the form of the old-fashioned todo list. These are web-based lists that multiple group members can access from any web browser.
- Tasks are entered (complete with due date) and checked off when completed.
- Some of the more popular online to-do list applications include Bla-Bla List (www.blablalist.com), Remember the Milk (www.rememberthemilk.com), Ta-da List (www.tadalist.com), Tudu List (www.tudulist.com), and Voo2Do (www.voo2do.com).
- All of these applications are simple enough for even the most techno-phobic group members to use

-

- Some even let you add new tasks via email or access your lists via mobile phone.

(ii) Collaborating on Task Management

- For managing more complex tasks, a simple to-do list application might not work. Instead, consider using a web-based task management application that lets you manage the multiple pieces and parts of large projects.
- Basic task management can be accomplished with applications such as HiTask (www.hitask.com) and Zoho Planner (planner.zoho.com). For the most complicated projects, consider using a dedicated project management application, such as Basecamp (www.basecamphq.com) or Goplan (www.goplan.com).

(iii) Collaborating on Event Management

- When you're putting on a big event such as a concert or conference, you have a whole new set of challenges to face. Not only do you have to manage the tasks involved with putting together the event, you also have to handle attendee registration, event marketing, ticket sales, and the like.

-

  - It’s a massive effort—made somewhat easier by web-based event management tools. With web-based event management applications, the cloud hosts everything you need to schedule and market your events, as well as handle registration, payment, and other important tasks.
  - For example, you can create an online event calendar so that attendees can learn about and sign up for future events via the web; offer web-based event registration and payment; manage requests for hotel rooms, etc.
  - These are very robust applications, capable of handling every last detail over the web. Some of the most popular of these apps include Cvent (www.cvent.com), RegOnline (www.regonline.com), and ViewCentral (www.rmkr.com/viewcentral). Unlike some other web-based apps, these aren’t free; you have to pay for the power you need to manage the details of your particular event.

(iv) Collaborating on Event Marketing

- When it comes to promoting your community events, you want to go beyond the basics to more creative forms of marketing.
- For example, you may want to create a brochure or flyer to announce your event. Fine and dandy, but everybody in the group (including all the community bigwigs) wants input on the final piece.
- This may have been difficult in precloud days, but now you can use a web-based application such as Google Docs (docs.google.com) to create your piece and make it available online for everyone to see and comment on. (Just

-

remember to dole out read-only authorization to these interested parties; you don't want everybody in the group going online and making changes to what you've just created!)

- Naturally, you can also use web-based local search sites, such as Zvents (www.zvents.com), to post announcements of your community events.
- You may even want to use cloud-based social media sites, such as Facebook and MySpace, to promote your event online.
- And, after the event, you can post pictures of the event on community photo-sharing sites, such as Flickr (www.flickr.com). It's all possible because of the cloud!

(v) Collaborating on Budgets

- Every event, small or large, comes with its own set of costs. And with community events, those costs are often managed by a group of people, each responsible for a specific operation or group of operations.
- For simple events, you can collaborate on your budget using web-based spreadsheet programs, such as Google Spreadsheets (part of the Google Docs suite, at docs.google.com). Just create a private spreadsheet, authorize access for each member responsible for the budget, and then start adding data online.

-

- When everyone has finished entering numbers for their line items, the group member responsible for the entire budget can log on and do her thing.
- For larger or more complex events, you may want to use the budget function available in most event management programs. You may also want to consider some of the accounting applications in the Salesforce.com AppExchange (www.salesforce.com/appexchange/).
- Some of these web-based apps are relatively low-priced, which is an attractive asset for most cash-strapped community groups.

➢ COLLABORATING ON CALENDARS, SCHEDULES, AND TASK MANAGEMENT

- This section takes a look at different types of personal and business computing tasks, and at the web-based applications that can facilitate those tasks.
- Whether you want to keep a simple group calendar or to-do list or need something more powerful to schedule appointments and meetings, you’re sure to find some web-based application

❖ Exploring Online Calendar Applications

- Most computer users today have embraced keeping their schedules on their PCs. Not that the old-fashioned wall-hanging calendar is dead, it’s just that

-

it’s a whole lot easier to track appointments and events electronically; the computer does all the busywork for you.

- The problem, however, with using calendar software (such as Microsoft Outlook or Windows Calendar) is that all your appointments have to reside on a single computer.
- If you keep a personal calendar on your home PC, you can’t reference it from work or when you’re traveling. That limits the calculator program’s usefulness.
- That’s why, instead of using a calendar that’s wedded to a single computer, many users are moving to web-based calendars.
- A web-based calendar service stores your calendars on the Internet, where they can be accessed from any computer that has an Internet connection.
- This lets you check your schedule when you’re on the road, even if your assistant in the office or your spouse at home has added new appointments since you left.
- Web-based calendars are also extremely easy to share with other users in any location, which make them great for collaborative projects.

(i) Google Calendar

- Google Calendar The most popular web-based calendar today, no doubt due to its association with the web’s most-used search engine, is Google Calendar (calendar.google.com).

-

- Google Calendar is free, full featured, and easy to use. It lets you create both personal and shared calendars, which makes it ideal for tracking business group, family, and community schedules.
- Like all web-based calendars, all your events are stored in the cloud (in this case, the cloud created by Google's own network of servers), not on your own computer.
- This means that you can access your calendar from any computer anywhere in the world. Just log in to the Google Calendar page and your calendar and all events are there.
- Because Google Calendar is web based, you can use it to create not only a private calendar for yourself, but also public calendars for your company or organization.
- Create a public calendar and all employees or attendees can access it via the web. In addition, special event invitation features make it easy to invite others to an event—public or private.
- In addition, Google allows you to create several different—and different types of—calendars. You can create one calendar for home, another for work, and yet another for your son's soccer team.
- Then you can view all your calendars from the same Google Calendar page, with the events from each calendar color-coded for easy visibility.
- The different types of calendars that can be created with Google Calendar are: (a) Personal calendars, like your default calendar (b) Public calendars, which others can access via the web (c) Friends' calendars, which you

•

import from their Google Calendar web pages (d) Holiday calendars, which add national holidays to a basic calendar.

- Because it’s part of the mighty Google Empire, Google Calendar integrates smoothly with Google’s Gmail application.
- Google Calendar can scan your email messages for dates and times and, with a few clicks of your mouse, create events based on the content of your Gmail messages.

(ii) Yahoo! Calendar

- One of Google Calendar’s primary competitors is Yahoo! Calendar (calendar.yahoo.com), hosted by its search competitor Yahoo!
- This web-based calendar looks, feels, and functions quite similarly to Google Calendar, and is also free for anyone to use.
- One subtle difference in Yahoo! Calendar, however, is the presence of an Add Task button.
- This reflects Yahoo! Calendar’s offering of tasks in addition to events. You can still add individual items to your daily schedule, but you can also add longer-term tasks and have their due dates show up on your calendar. It’s a nice addition.
- Of course, you can share your Yahoo! calendars with other users, in a collaborative environment.

-

- Just click the Sharing link and indicate how you want to share—no sharing, view-only for friends, view-only for anyone, or view-only with special friends allowed to edit.
- At present, Yahoo! Calendar only lets you create a single calendar. All your events, public and private, have to be stored on this calendar; you can't create different calendars for different functions. (That's one advantage that Google has over Yahoo! here.)

(iii) Windows Live Calendar

- Because Google and Yahoo! both offer web-based calendars, it's no surprise that the third-largest search site also has a competitive offering.
- Windows Live Calendar (mail.live.com/mail/calendar.aspx) is Microsoft's web-based calendar, actually part of the Windows Live Hotmail email service.
- Windows Live Calendar looks a lot like both of its primary competitors. It offers tasks, like Yahoo! Calendar, and also lets you schedule meetings with other calendar and Hotmail users.

(iv) Apple MobileMe Calendar

-

- Apple’s MobileMe (www.me.com) is a new competitor in the web-based apps market. It includes online mail, contacts, and calendar, as well as an online photo gallery and file storage.
- The MobileMe Calendar is, of course, a web-based calendar that can be accessed from any computer connected to the Internet, Mac or Windows.
- What makes it more unique and potentially more useful is that it can also be accessed from Apple’s iPhone, which makes it a truly mobile calendar.
- As with competing calendars, you can display MobileMe in daily, weekly, or monthly modes.
- You can also synchronize your MobileMe calendars with Apple’s iCal and Microsoft Outlook calendars.
- Even though MobileMe Calendar doesn’t offer much new or innovative (save for the iPhone interoperability, of course), it’s bound to be a strong competitor in the online apps market, especially for non-business users.
- That’s partly because of Apple’s cachet (everything Steve Jobs does is cool, for some folks), and partly because Apple does tend to get the details right.
- MobileMe Calendar looks and feels a little slicker than all its competitors, Google Calendar included. It’s certainly worth a look—even if you’re already using another online calendar.

(v) AOL Calendar America

-

- Online isn’t quite the powerhouse that it used to be, but it still has millions of users, both paid subscribers and free web users.
- Any registered user can access AOL Calendar (calendar.aol.com), which integrates with the AOL Instant Messenger (AIM) service for both instant messaging and email.
- As with competing calendars, AOL Calendar lets you share calendars with authorized users; your calendars can be either private or public.

(vi) CalendarHub

- CalendarHub offers all the features found in the previously discussed webbased calendars—private/public calendars, sharing/collaboration, multiple calendars, task-based to-do lists, and the like.
- In addition, CalendarHub lets you publish calendars on your blog or website, which makes it great for creating sites for community groups, sports teams, and the like.
- Other users can sign up to receive email notification of new events, or subscribe to RSS feeds for any calendar view. And, of course, it’s completely free.

(vii) Hunt Calendars

- Hunt Calendars (www.huntcal.com) offers event-based web calendars.

-

- Useful features include email reminders, notification of event conflicts, notification of new and updated events, and the like.
- The site lets you add web links and images to calendar events, which is fairly unique.
- Also nice is the ability to customize the color scheme and graphics to reflect your organization's look and feel.

(viii) Famundo

- This site offers Famundo for Organizations, a free webbased calendar ideal for schools, churches, sports teams, and the like.
- After the public calendar has been created, users can subscribe to be notified of new and upcoming events.
- You can also add message boards, blogs, and other features to your calendar.
- The company also offers Famundo for Families, a personal version of their Organizations calendar. This version includes a family address book and message board, to facilitate family communication.

(ix) eStudio Calendar

- eStudio Calendar (www.same-page.com/calendar-software.html) is designed specifically for business use.

-

- You get three types of calendars in a single interface: Member Event calendar helps users manage their personal time, keep track of meetings with others, and so on.
- Team Event calendar is used to schedule activities for a group, as well as schedule facilities. Supervisor calendar provides reports to managers about business activities and schedules.
- In addition, you can use eStudio Calendar to broadcast information about group activities (via email) and to schedule meetings. Information about company events can also be automatically published to your website.

(x) 30Boxes

- The name of 30Boxes (www.30boxes.com) refers to the 30 “boxes” displayed on a typical monthly calendar.
- The site itself offers a slick interface for adding events. All your events can be shared with other designated users, plus you get to-do lists, a link to Google’s Gmail, and similar useful features.

(xi) Trumba

- Trumba (www.trumba.com) offers web-based calendars ideal for community organizations, schools, and similar public entities.

-

- The company lets you embed individualized widgets (dubbed “spuds”), which let users view full calendars, add events to the schedule, receive email notification of events, and such.

(xii) Calendars Net

- Calendars Net (www.calendars.net) is a free web-based calendar designed for companies or individuals who want to add interactive calendars to their websites.
- A typical calendar fits into a frame on your website, with little coding required.
- The site also hosts personal calendars in the cloud. You can employ four different levels of security (so that different users can view the calendar), add events, edit events, and even change universal calendar settings.

(xiii) Jotlet

- Here’s another way to add web-based calendar functionality to your website.
- Jotlet (www.jotlet.net) is a JavaScript API and library that you can use to build rich calendar functionality into any web page.
- If you’re skilled in HTML programming, this is a good way to build a calendar-based page.

-

- The Jotlet API is free for noncommercial use, and also available (for a fee) for commercial sites.

❖ Exploring Online Scheduling Applications

- A web-based online scheduling app takes much of the pain out of scheduling meetings, for both large and small groups.
- The typical app requires all users to enter their individual calendars beforehand.
- When you schedule a meeting, the app checks attendees' schedules for the first available free time for all.
- The app then generates automated email messages to inform attendees of the meeting request (and the designated time), followed by automatic confirmation emails when attendees accept the invitation.

(i) Jiffle

- Jiffle (www.jifflenow.com), which schedules meetings, appointments, and the like for the enterprise environment.

-

- To track employees’ free time, it synchronizes seamlessly with both Microsoft Outlook and Google Calendar.
- It also offers its own Jiffle Calendar application.
- Jiffle allows the originating user to mark available time slots on his calendar, and then share them with proposed attendees via a Jiffle-generated email invitation.

(ii) Presdo

- Unlike Jiffle, Presdo (www.presdo.com) is a scheduling tool that isn’t limited to a single company.
- Presdo lets you schedule meetings and events with anyone who has an email address.
- Adding an event is as simple as entering a description into a box.
- You then enter the email addresses of other participants, and Presdo emails out the appropriate invites.
- When an attendee responds, he’s automatically added to the event’s guest list.

-

(iii) Diarised

- Diarised (www.diarised.com) is, like Presdo, a web-based meeting maker that users across different companies can use.
- It helps you pick the best time for a meeting by sending out emails to invitees, letting them choose the best times for them, and then sending you a summary of those best dates.
- You pick the final date, Diarised notifies everyone via email, and your meeting is scheduled.

(iv) Windows Live Events

- Event scheduling is now part of Microsoft’s bag of tricks. Microsoft’s Windows Live Events (home.services.spaces.live.com/events/) is a customized version of its Live Spaces offering; it lets Live Spaces users organize events and share activities between participants.
- To schedule an event, you set up a list of invitees and then send out a mass email with a link back to your Live Event site. (All the event details are also available as an RSS feed.)

-

- Information about the event is posted on the site itself, which also serves as a place for attendees to come back after the event and share their photos, videos, and blog posts about the event.
- With its user-friendly consumer features, Live Events isn't robust enough (or professional enough) for most business users. It is, however, a nice way to plan more personal and informal events.

(v) Schedulebook

- Schedulebook (www.schedulebook.com) offers several different types of web-based scheduling services.
- Depending on the application, you can use Schedulebook to schedule employees, customers, or other interested parties.
- The company's three offerings are:
  - (a) Schedulebook Professionals, which is a business-oriented schedule/calendar/planning application
  - (b) Schedulebook Office, which schedules the use of any shared resource, such as company meeting rooms or even vacation homes
  - (c) Schedulebook Aviation, which is used by the aviation industry to schedule aircraft, flight training, and similar services

(vi) Acuity Scheduling

-

- Acuity Scheduling (www.acuityscheduling.com) can help ease your scheduling operations.
- Acuity Scheduling lets you clients schedule their own appointments 24/7 via a web-based interface, you don’t have to manually schedule any appointment.

(vii) AppointmentQuest

- Like Acuity Scheduling, AppointmentQuest (www.appointmentquest.com) is designed to solve the scheduling problems of busy professionals.
- This application not only enables clients to make and you to accept appointments over the web, it also lets you manage personnel, schedules, and other calendar related items.

(viii) hitAppoint

- Our last scheduling application, hitAppoint (www.hitappoint.com), also enables online client booking.
- Like the previous similar application, it’s ideal for any business that requires the making of customer appointments—barbershops, hair salons, doctor and dentist offices, consultants, financial advisors, car repair shops, computer technicians, and the like.

-

❖ Exploring Online Planning and Task Management

- Planning and task applications let you manage everything from simple to-do lists to complex group tasks, all over the Internet and collaboratively with other users.

(i) iPrioritize

- iPrioritize (www.iprioritize.com) is a good basic to-do list manager. \
- The authorized users can create a new to-do list, add items to the list, prioritize tasks by dragging them up and down the list, and mark items complete when finished.
- And, because it's web based, you can access your lists anytime and anyplace.

(ii) Bla-Bla List

- Bla-Bla List (www.blablalist.com) is another simple to-do list manager.
- It's web based, of course, so you can access your lists from any location at any time.
- You can even publish your lists via RSS so that family and coworkers can get instant updates.

-

(iii) Hiveminder

- Hiveminder (www.hiveminder.com) is similar to all the previously discussed to-do list managers.
- What's nice about Hiveminder is that you can enter list items in a kind of freeform fashion, and it will help you create and prioritize lists based on your "brain dumps."

(iv) Remember the Milk

- When you need to "remember the milk" at the grocery store, check out the aptly named Remember the Milk (www.rememberthemilk.com) web-based todo list manager.
- Once you create a list, you can arrange reminders via email, instant messaging, or text messages to your mobile phone.

(v) Ta-da List

- Here's another web-based to-do list manager.

-

- Ta-da List (www.tadalist.com) lets you make all sorts of lists, share them with friends, family, and coworkers, and then check off items as they're completed.

(vi) Tudu List

- Tudu List (www.tudulist.com) is a little different from other to-do list managers in that it also includes a web-based calendar.
- Items are added both to the appropriate to-do list and to your calendar, on the date they're due.

(vii) TaskTHIS

- TaskTHIS (taskthis.darthapo.com) is similar to most other to-do list managers, but offers the ability to add extended notes to any individual task.
- You can publish your tasks via RSS or share with others via the web.

(viii) Vitalist

- Like other to-do list managers, Vitalist (www.vitalist.com) organizes all sorts of tasks and projects.
- It's unique in that it uses the Getting Things Done (GTD) workflow methodology popularized by management consultant David Allen.

-

(ix) TracksLife

- Trackslife (www.trackslife.com) is a database-oriented task manager.
- Each “track” is a separate database that combines columns of money, numbers, words, paragraphs, and yes/no responses.
- The application sends out reminders of critical events via email or RSS.

(x) Voo2Do

- Voo2Do (www.voo2do.com) moves beyond simple to-do list management into more sophisticated priority management.
- This web-based application lets you set up different projects, organize tasks by project, track time spent and remaining on a given task or project, publish task lists, and even add tasks via email.

(xi) HiTask

- More sophisticated task management can be had with HiTask (www.hitask.com), a business-oriented task manager.
- Tasks are added to your calendar and color tagged for easy viewing.

-

- The task manager and scheduler both utilize drag-and-drop editing, and you can share and assign tasks and projects to a group of people via the web.

(xii) Zoho Planner

- Zoho Planner (planner.zoho.com) is perhaps the most sophisticated task planner evaluated here.
- Its features and functionality approach those of the project management applications.
- Zoho Planner is ideal for anyone managing small- to medium-sized projects.
- It’s probably overkill for simple to-do list management (try iPrioritize or Remember the Milk, instead), and not powerful enough for large corporate projects.
- But for the average home or community project, it’s an ideal solution—just enough versatility to handle disparate types of projects, but not so complex as to scare off nontechnical users.

➢ COLLABORATING ON EVENT MANAGEMENT

-

  - To stage a successful event, you have to market it to potential attendees, sign up those attendees, process their fee payments, make sure that the event space and conference rooms are properly scheduled, handle travel and hotel arrangements, register attendees when they arrive onsite, manage event workers, and make sure everything runs on time during the event. It’s a tremendous undertaking.

- ❖ **Understanding Event Management Applications**

  - Less sophisticated apps may focus on one or two operations, such as event registration or facilities booking.
  - The more full-featured apps include management of everything from pre-event marketing to post-event analysis.

(a) Event Planning and Workflow Management

- A successful event starts well in advance of its opening date.
- There are tons of details involved in an event of any size, and managing all those tasks takes quite a bit of computing horsepower—just the thing cloud computing can help you out with.
- Most event management applications include robust task planning modules, similar to what you’d find in higher-end task management applications or lower-end project management apps.

-

(b) Event Marketing

- Unless you let people know about your event, you could be disappointed with the final attendance.
- To that end, many event management applications include modules to help you market your event.

(c) Event Calendar

- Another part of your event marketing mix is an event calendar—an online calendar that displays all the happenings within your overall event.
- This proves particularly useful if you’re hosting a conference or trade show made of lots of individual panels, sessions, or meetings.
- You can post each individual event on the main event calendar, easily accessed by any attendee or potential attendee with a web browser.

(d) Facilities Scheduling

- Unless you’re running a one-room meeting, chances are your event involves multiple rooms and maybe event multiple locations.
- If so, you need to be able to schedule different rooms for different components of your event; when a participant or group asks for a room, you need to be able to see what’s available and when.

-

- To that end, most event management apps include a facilities scheduling module. Ideally, this module ties into the event host’s systems, giving you complete power over room or hall scheduling.

(e) Advance Registration

- Larger events require or encourage advance registration of participants.
- To that end, most event management apps include a web-based registration module, where attendees can sign up (and, in most cases, pay) for the event.
- Attendee information is entered into a web form, and that data is then stored on the application provider’s cloud servers.
- You then access attendee data from your own computer, wherever you may be.

(f) Payment Processing

- Collecting payment for your advance and onsite registrants is a key part of the event management experience.
- You want the event management software to tie payment processing into the registration process, letting you accept payment via credit card, PayPal, or whatever other payment methods you accept.

(g) Travel Management

-

- If you’re running a real “hands-on” event, you might want to consider offering travel services to select attendees.
- This may be as simple as arranging ground transfer services (taxis, buses, and so on) between your local airport and the event hotel, or as advanced as linking into an online travel site or airline reservations system to provide flight reservations.
- Although not all event management applications offer this type of functionality, it is available with some apps if you need it.

(h) Housing Management

- More common is a housing management module that helps match event attendees with available rooms at your event hotel.
- Many attendees prefer to have the event host handle their hotel reservations, so that you serve as kind of a “one-stop shop” for all your attendees needs.
- The best event management apps link directly from advance registration and payment into the hotel’s reservation system—and then let you confirm rooms and such at the event site.

(i) Onsite Registration

-

- Your attendees sign up (and probably pay) for your event in advance. But when they arrive on opening day, you need to sign them in, print out badges, provide a welcoming packet, and so forth.
- All of these tasks are managed by the event management application's onsite registration module.
- Ideal onsite registration ties into the advance registration and, optionally, the housing management modules of the application.

(j) Contact Management

- Using the master database of event guests, you can provide contact management services.
- At the very least, your event management application should let you print out (or host online) a master directory of attendees, which can then be provided as part of the welcoming packet of materials.

(k) Budget Management

- Running an event is an expensive and complex undertaking; your overall budget includes hundreds of individual expense items.
- To that end, your event management application should include a robust accounting or budget management module, to track both your expenses and your income.

-

(l) Post-Event Reporting and Analysis

- When the event is (finally!) over, your job isn't quite done yet. Not only do you have to balance the books, you also need to look back on the entire event and determine how successful it was.
- That's why most event management applications include some form of post-event reporting and analysis.
- Some apps even let you send and process attendee surveys, which can provide valuable feedback from those who were there.

❖ Exploring Event Management Applications

(i) 123Signup

- The company offers four different applications: Event Manager, Association Manager, Training Manager, and Member Directory. Of these, the one in which we're interested is the aptly named Event Manager.
- 123 Event Manager is scalable, so it can be used for both smaller (employee meetings, stockholder meetings, alumni meetings, and so forth) and larger (trade shows, fundraisers, conferences, and so on) events.

-

(ii) Acteva

- Acteva (www.acteva.com) offers online event registration and payments.
- Using Acteva's web-based solutions, you can handle event registration, ticketing, and payment handling (via any major credit card) directly from your own website.
- You can then sort and manage all event registration data online.

(iii) Conference.com

- Conference.com (www.conference.com) offers one of the most full-featured web-based event management applications available today.
- By using Conference.com's cloud servers, even small events can utilize the company's powerful event management tools, designed to serve the needs of the largest events.
- Your data (and the behind-the-scenes application modules) are hosted by Conference.com's secure servers, accessible from any Internet-enabled location.

(iv) Cvent

- Competing directly with Conference.com is Cvent (www.cvent.com), with its Event Management system.

-

- Like Conference.com, Cvent's Event Management system is a suite of interrelated tools.

(v) Event Wax

- Event Wax (www.eventwax.com) isn't quite as full featured as other event management solutions.
- In fact, it really isn't designed to handle large-scale events such as trade shows and conferences.
- Instead, Event Wax is for smaller-scale in-house events, such as company meetings, parties, open houses, and the like.

(vi) RegOnline

- Like eventsbot, RegOnline (www.regonline.com) offers online event registration and payment.
- You use RegOnline to create a website for your event, create web-based registration forms, accept credit card payments, send automatic email reminders and confirmations, print name badges and room signs, and generate all manner of custom reports.

-

- The application also handles the reservations of individual hotel rooms and room blocks.

(vii) Setdot

- Setdot (www.setdot.com) isn’t really for large corporate events; it’s more of a stylish web-based way to schedule and manage smaller personal events and activities.
- Setdot lets use choose from various preset themes for your event web page.
- It even displays maps and directions to events. And, although it’s mainly for smaller events, it does manage guest responses and messages.

(viii) Tendenci

- Here’s another unique approach to event management.
- Tendenci (www.tendenci.com) combines a web-based calendar application with online registration and payment.
- You create an event calendar, which you embed in your own website. When an interested party clicks an event link, he’s taken to a dedicated page for that event, where he can see.

➢ COLLABORATING ON PROJECT MANAGEMENT

-

- Managing a large project can be an exhaustive task. Even the smallest project has numerous pieces and parts, all of which have to be completed in a precise order and on an exacting timetable for the project to come in on time and on budget.
- If just one piece slips, the whole project goes out of whack. The process of managing a project gets even more complex when the participants are in different locations.
- When you employ a web-based project management application, you can more easily manage all the pieces and parts, no matter where the players are located.
- Your project is turned into a single database hosted in the cloud, accessible by all from any Internet-connected computer

❖ Understanding Project Management

- The project itself can be anything, from creating a product brochure to implementing a new hiring process to launching a new product line.
- The challenge, of course, is completing the project by the assigned date—and to the agreed-upon budget.
- Key to this is the tight management of each and task that comprises the project; if all the component tasks are completed on time and on budget, the entire project will be completed as planned.
- If one or more tasks slip—and you can't make up the lost time elsewhere—your project will come in late.

-

- To manage the individual tasks within a project requires managing a larger set of resources—people, of course, but also money, materials, space, communications, and the like.
- This resource management is crucial to ensuring the eventual success of a project.
- Project management professionals like to think in terms of juggling a certain set of constraints: scope (what must be done to produce the end result), time (the amount of time available to complete the project), and cost (the budgeted amount available for the project).
- The key to effective project management is to use all available tools and techniques that enable the project team to organize their work to meet these constraints.

❖ Exploring Project Management

- Applications Traditional project management software helps project managers and team members organize and track all the various tasks in a project.
- To do this, the software typically includes scheduling, budget management, and resource-allocation components. Web-based project

-

management applications do all this online, with a centralized project file accessible to all team members.

- This enables improved communication and collaboration between members of the project team.
- The scheduling component of a project management application helps the project manager schedule the series of events that comprise the total project.
- After the project has been planned, it then has to be executed.
- The project management application should enable this execution by creating task lists for team members, allocation schedules for project resources, overview information for the team manager, and, as the project progresses, an early warning of any risks to the project's completion.

(i) @task

- The web-based project management program known as @task (www.attask.com) offers a variety of traditional projection management functions.
- The application includes an interactive drag-and-drop Gantt chart critical path analysis, project milestones, planned/projected/ estimated comparisons, resource scheduling, issue management, and calendar views for project tasks.

-

- Tasks can even be managed remotely via a special software widget for Apple’s iPhone.

(ii) AceProject

- AceProject (www.aceproject.com) is an easy-to-use web-based project management application.
- It lets users manage multiple projects using multiple resources and share those resources across projects.
- Tasks can be tracked via a variety of filters that fine-tune the results, and the application offers a number of different project reports and statistics.
- AceProject also offers time tracking, email notification of task deadlines, and a monthly project calendar.

(iii) Basecamp

- One of the most popular project management applications today is Basecamp (www.basecamphq.com).
- Its web-based nature makes it viable for both internal and external (client) projects.

(iv) Copper Project

-

- Copper Project (www.copperproject.com) is a project management application that can be hosted either on the company's servers or on your own server. Either version enables web-based collaboration.
- Copper includes useful features such as a drag-and-drop weekly or monthly timeline, resource management, email alerts, statistical reports, and a unique personal time management tool.

(v) eStudio TaskTracker

- TaskTracker from eStudio (www.same-page.com/online-project-management07.html) is an easy-to-use online project management application.
- This program includes features such as task lists, work logs, issue management, automatic task dependencies, subproject capability, budget and expense tracking, Gantt charts, and a full set of management reports.

(vi) onProject

- Another company offering online projection management solutions is onProject (www.onproject.com).
- The company's myonProject application is a subscription service that offers collaborative project management functionality.

-

- The application’s Workspace page provides one-screen access to all key operations.

(vii) Project Drive

- The Project Drive (www.project-drive.net) application includes communication and collaboration features in addition to basic project management functionality.
- Users get a customizable overview dashboard, templates for fast project setup, Gantt charts, task management, resource allocation, document sharing and management, automated communication tools, a group calendar, cost analysis and budgeting, and a large number of management reports.

(viii) Vertabase

- Vertabase (www.vertabase.com) is a popular web-based project management application.
- It offers a summary executive dashboard, multiple schedule views, project portfolio, cross-project Gantt charts, resource planning, budget control, issue tracking, and a detailed project schedule.

(ix) Wrike

-

- Wrike (www.wrike.com) is a project management application that offers a unique way to create project tasks.
- The application is email based; emails from project members are automatically converted into tasks in the appropriate project.
- Wrike then automatically reminds employees about overdue tasks, creates individual schedules for employees, and generates Gantt charts for each project.

(x) Zoho Projects

- Our final web-based project management application is Zoho Projects (projects.zoho.com), another popular product from the Zoho cloud combine.
- Zoho Projects is a standard project management application, complete with tasks and milestones, a project calendar, Gantt charts and other reports, time tracking, and group file sharing.

➢ COLLABORATING ON WORD PROCESSING

- Just about everyone who uses a computer uses a word processing program. You use your word processor— most likely some version of Microsoft Word—to write memos, letters, thank you notes, fax coversheets, reports, newsletters, you name it.

-

- The word processor is an essential part of our computing lives.
- There are a number of web-based replacements for Microsoft's venerable Word program.
- All of these programs let you write your letters and memos and reports from any computer, no installed software necessary, as long as that computer has a connection to the Internet.
- And every document you create is housed on the web, so you don't have to worry about taking your work with you.
- Web-based word processors, in contrast, are hosted in the cloud, not on your hard drive—as are the documents you create with these applications.
- By being web based, you can easily share your documents with others. That makes real-time workgroup collaboration possible from anywhere around the globe, which is something you don't have with Microsoft Word and similar desktop software programs.
- Another benefit of being web based is that you can't lose your work—theoretically, anyway. After you've named the document you're working on, the web-based word processor saves your file on its cloud of servers.
- Best of all, most of these web-based applications are free. That's free, as in it costs zero dollars, unlike the ever increasingly expensive Microsoft Office suite. Being free makes it easy to take for a test drive, and even easier to add to your bag of applications.

❖ Exploring Web-Based Word Processors

-

(i) Google Docs

- Google Docs (docs.google.com) is the most popular web-based word processor available today.
- Docs is actually a suite of applications that also includes Google Spreadsheets and Google Presentations; the Docs part of the Docs suite is the actual word processing application.
- As with all web-based word processors, when you create a Google Docs document, you’re actually creating an HTML document—just like a web page.
- All HTML-type formatting is available for your documents, through the Google Docs interface.
- The document is also saved in HTML format, although you can export (download) the document in a number of other formats, including Microsoft Word DOC format and Adobe PDF.
- Of course, one of the most useful features of Google Docs is the capability to share a document with other Google Docs users, either for viewing or for collaborative editing.

(ii) Adobe Buzzword

-

- Buzzword (buzzword.acrobat.com) is Adobe's entry into the web-based word processor marketplace.
- Unlike Google Docs, Buzzword runs in Flash, which might be problematic for users with older PCs or those with slow Internet connections.
- That said, Flash implementation gives Buzzword a snazzy interface and some advanced editing and formatting features.
- The Buzzword interface is head and shoulders above the more utilitarian interface of Google Docs.
- In addition, Buzzword gives you full text and paragraph formatting, headers and footers, page numbering, endnotes, and keyboard shortcuts, none of which are currently available with Google Docs.

(iii) ajaxWrite

- Unlike most other web-based word processors, ajaxWrite (www.ajaxwrite.com) doesn't work with Internet Explorer. Instead, you have to use the Firefox web browser.
- This not unimportant caveat aside, ajaxWrite's simple interface and clean workspace makes it well liked by many users.
- ajaxWrite looks a lot like Microsoft Word, which makes it easy to start using the program right away.
- New documents open in their own windows, complete with Word-like pull-down menus and toolbars.

-

(iv) Docly

- Docly (www.docly.com) is an interesting application, designed especially for professional writers.
- What sets Docly apart from other web-based word processors is its focus on copyright management, including the ability to assign a document a Creative Commons license or a traditional “all rights reserved” license.
- This means that not only can you share and publish your Docly documents, you can also offer them for sale.

(v) Glide Write

- Glide Write (www.glidedigital.com) is part of the Glide Business suite of webbased applications.
- Glide Write itself is an elegant word processor that just happens to integrate seamlessly with other Glide applications, including email and chat.
- In addition, Glide documents can be viewed on a number of smartphones, including the iPhone, T-Mobile SideKick, and a handful of Treo and BlackBerry models.

(vi) iNetWord

-

- The iNetWord (www.inetword.com) web-based word processor is a full-featured application.
- iNetWord features a tabbed interface, with each open document appearing on its own tab.
- You get support for page backgrounds, borders, page numbering, tables, images, and the like.
- It even comes with a number of predesigned templates for common types of documents.
- For group collaboration, iNetWord lets you share individual documents or entire folders.

(vii) KBdocs

- KBdocs (www.kbdocs.com) is a no-frills online word processor. There are only limited formatting options, and it doesn’t have any sharing or collaboration features.
- That said, it’s probably the easiest-to-use web-based word processor, especially for newbies; just pick a username and password, click Enter, and you’re ready to go.

(viii) Peepel WebWriter

-

- Peepel WebWriter (www.peepel.com) is part of a multi-application web-based office suite.
- The Peepel interface is a trifle unusual: The document you're editing appears in its own window, on top of the larger home window that holds the toolbar and tabs that you use to edit and format the document.

(ix) ThinkFree Write

- ThinkFree Write (www.thinkfree.com) is a Java-based online word processor. That lets ThinkFree mimic the Word 2003 interface.
- Each new document opens in its own window, each of which has a Word-like pull-down menu and toolbar.
- The editing and formatting functions are also quite Word-like, complete with styles, editing marks, fields, an autocorrect function, and the like.

(x) WriteBoard

- If collaboration is your game, consider WriteBoard (www.writeboard.com), a web-based word processor designed with group collaboration in mind.
- WriteBoard isn't the most full-featured word processor on the web, but it does make collaboration between multiple users remarkably easy.

-

- After you create a document and share it with others, it’s easy to compare different versions of the document; every time you or someone else saves an edit, a new version of the document is created and linked to in the sidebar.

(xi) Zoho Writer

- Zoho’s web-based applications always end up being the last discussed in this book, thanks to the company’s last-of-the-alphabet name.
- Case in point: Zoho Writer (writer.zoho.com), which easily holds its own, if not surpasses, Google Docs in the web-based word processor race.
- Naturally, Zoho Writer offers robust sharing and collaboration features. You can share a document with individuals or with groups on either a read-only or read/write basis. Sharing is as easy as clicking the Share tab.

•

CONTENT BEYOND SYLLABUS

CLOUD SECURITY CHALLENGES

The worldwide public cloud services market is forecast to grow 17% in 2020 to total $266.4 billion, up from $227.8 billion in 2019 according to Gartner. As the cloud continues to be more and more heavily adopted, it's important to be aware of the challenges organizations are faced with when leveraging cloud computing. Recently the Cloud Security Alliance presented the following major cloud challenges in its report "Top Threats to Cloud Computing: Egregious Eleven." In this blog, I will be summarizing each threat covered in the report and discuss its implications to organizations today.

1. Data Breaches

Consequences of a data breach may include:

1. Impact to reputation and trust of customers or partners
2. Loss of intellectual property (IP) to competitors, which may impact products release
3. Regulatory implications that may result in monetary loss
4. Brand impact which may cause a market value decrease due to previously listed reasons
5. Legal and contractual liabilities
6. Financial expenses incurred due to incident response and forensics

2. Misconfiguration and Inadequate Change Control

This is one of the most common challenges of the cloud. In 2017, a misconfigured AWS Simple Storage Service (S3) cloud storage bucket exposed detailed and private data of 123 million American households. The data set belonged to Experian, a credit bureau, which sold the data to an online marketing and data analytics company called Alteryx. It was Alteryx that exposed the file. Such instances can be disastrous.

3. Lack of Cloud Security Architecture and Strategy

Worldwide, organizations are migrating portions of their IT infrastructure to public clouds. One of the biggest challenges during this transition is the implementation of appropriate security architecture to withstand cyberattacks. Unfortunately, this process is still a mystery for many organizations. Data are exposed to different threats when organizations assume that cloud migration is a "lift-and-shift" endeavor of simply porting their existing IT stack and security controls to a cloud environment. A lack of understanding of the shared security responsibility model is also another contributing factor.

-

4. Insufficient Identity, Credential, Access and Key Management

Cloud computing introduces multiple changes to traditional internal system management practices related to identity and access management (IAM). It isn’t that these are necessarily new issues. Rather, they are more significant issues when dealing with the cloud because cloud computing profoundly impacts identity, credential and access management. In both public and private cloud settings, CSPs and cloud consumers are required to manage IAM without compromising security.

5. Account Hijacking

Account hijacking is a threat in which malicious attackers gain access to and abuse accounts that are highly privileged or sensitive. In cloud environments, the accounts with the highest risks are cloud service accounts or subscriptions. Phishing attacks, exploitation of cloud-based systems, or stolen credentials can compromise these accounts.

6. Insider Threat

The Netwrix 2018 Cloud Security Report indicates that 58 percent of companies attribute security breaches to insiders. Insider negligence is the cause of most security incidents. Employee or contractor negligence was the root cause of 64 percent of the reported insider incidents, whereas 23 percent were related to criminal insiders and 13 percent to credential theft, according to the Ponemon Institute’s 2018 Cost of Insider Threats study. Some common scenarios cited include: misconfigured cloud servers, employees storing sensitive company data on their own insecure personal devices and systems, and employees or other insiders falling prey to phishing emails that led to malicious attacks on company assets.

7. Insecure Interfaces and APIs

Cloud computing providers expose a set of software user interfaces (UIs) and APIs to allow customers to manage and interact with cloud services. The security and availability of general cloud services are dependent on the security of these APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent the security policy. Poorly designed APIs could lead to misuse or—even worse—a data breach. Broken, exposed, or hacked APIs have caused some major data breaches. Organizations must understand the security requirements around designing and presenting these interfaces on the internet.

8. Weak Control Plane

-

Moving from the data center to the cloud poses some challenges for creating a sufficient data storage and protection program. The user must now develop new processes for data duplication, migration and storage and—if using multi-cloud—it gets even more complicated. A control plane should be the solution for these problems, as it enables the security and integrity that would complement the data plane that provides stability and runtime of the data. A weak control plane means the person in charge—either a system architect or a DevOps engineer—is not in full control of the data infrastructure's logic, security and verification. In this scenario, controlling stakeholders don't know the security configuration, how data flows and where architectural blind spots and weak points exist. These limitations could result in data corruption, unavailability, or leakage.

9. Metastructure and Applistructure Failures

Cloud service providers routinely reveal operations and security protections that are necessary to implement and protect their systems successfully. Typically, API calls disclose this information and the protections are incorporated in the metastructure layer for the CSP. The metastructure is considered the CSP/customer line of demarcation—also known as the waterline. Failure possibilities exist at multiple levels in this model. For example, poor API implementation by the CSP offers attackers an opportunity to disrupt cloud customers by interrupting confidentiality, integrity, or availability of the service.

10. Limited Cloud Usage Visibility

Limited cloud usage visibility occurs when an organization does not possess the ability to visualize and analyze whether cloud service use within the organization is safe or malicious. This concept is broken down into two key challenges. Un-sanctioned app use: This occurs when employees are using cloud applications and resources without the specific permission and support of corporate IT and security. This scenario results in a self-support model called Shadow IT. When insecure cloud services activity does not meet corporate guidelines, this behavior is risky— especially when paired with sensitive corporate data. Gartner predicts that by 2020, one-third of all successful security attacks on companies will come through shadow IT systems and resources.

Sanctioned app misuse: Organizations are often unable to analyze how their approved applications are being leveraged by insiders who use a sanctioned app. Frequently, this use occurs without the explicit permission of the company, or by external threat actors who target the service using methods such as credential theft, Structured Query Language (SQL) injection, Domain Name System (DNS) attacks and more.

-

11. Abuse and Nefarious Use of Cloud Services

Malicious actors may leverage cloud computing resources to target users, organizations or other cloud providers. Malicious attackers can also host malware on cloud services. Cloud services that host malware can seem more legitimate because the malware uses the CSP’s domain. Furthermore, cloud-hosted malware can use cloud-sharing tools as an attack vector to further propagate itself.

-

RECENT TRENDS IN CLOUD COMPUTING

Let’s take a close look at these breakthrough technologies that work really well in Cloud Computing, unlocking new features and increasing capabilities for businesses:

Artificial Intelligence (AI)

Artificial intelligence has seen massive growth in the past few years, and in combination with other modern technologies such as Cloud it gives countless opportunities to industries.

Businesses demand AI solutions, including deep learning and machine learning from cloud companies. Here cloud-based AI plays a major role, enabling enterprises to innovate, scale, and grow.

It empowers companies to extend their capabilities in a cost-effective way. Excellent computing and storage features make cloud-based AI an ideal solution for any kind of business.

DevOps

DevOps and Cloud together offer leading solutions. The terrifying majority of cloud development employs DevOps to offer impactful services. The advantages of using DevOps with Cloud are speedily increasing, that’s why developers prefer to use both together.

-

Cloud computing provides DevOps automation with a standard and centralized platform for testing, deployment, and production.

Internet of Things (IoT)

Most of the IoT devices are already utilizing the cloud to connect these systems and generate useful data on the habits and patterns of users. You might know this, each major cloud platform offers an IoT solution, that's why multiple companies have started ameliorating their existing apps with IoT connectivity.

IoT technology generates humongous amounts of data that need to be stored in cloud storage or cloud-based applications.

Edge Computing

Edge computing greatly supports IoT applications by providing increased processing power and fast response. It has an excellent ability to process and store data while reducing latency.

Edge computing is critical because it generates new and improved ways for the enterprise. It is used to maximize operational efficiency, enhance performance and safety, automate all core business processes, etc.

As a result, edge computing helps create businesslike applications for companies. By integrating edge computing with cloud computing, businesses can unlock new possibilities in this modern digital world.

Blockchain

Blockchain technology has benefited various sectors, including banking & finance, logistics, healthcare, and more. It can be effectively applied to any industry involving the tracking of digitally-recordable transactions. Improved transparency, traceability, and security are the core factors that make this technology crucial for businesses.

-

Today, major **cloud service providers** are making it easy for enterprises to adopt blockchain technology, which is a perfect fit for cloud solutions. With the cloud, all sorts of organizations can leverage the benefit of blockchain, such as large scale data analysis, and data security.

Virtual Reality (VR) and Augmented Reality (AR)

When it comes to improved customer experience, applications of Virtual Reality (VR) and Augmented Reality can be effectively applied. But it requires appropriate computing resources such as the Cloud, which has the capability to deal with AR/VR applications.

Cloud tools make it easy for developers to create, test, and deploy these applications. Using the Cloud in combination with AR/VR, companies can offer impressive experiences to its customers, along with meeting the core business purposes.

Microservices

Microservices are software architecture, collection of various independent services. Each service has a definite business focus and interacts with the other language-agnostic etiquette, such as REST.

The microservice architecture allows the speedy, frequent, and guaranteed delivery of large, complex applications. When microservices are combined with Cloud computing, it offers amazing functionalities and enhances app performance.

Containers

Containers are a sort of operating system (OS) virtualization. A single container might be utilized to run small microservice or larger applications. Containers hold various elements such as files, libraries, and environment variables required to drive desired Software.

The Rightscale State of the Cloud 2019 stated that 66% of companies adopted containers. Similarly, 60% adopted Kubernetes, the container management system created by Google.

-

Serverless

Serverless computing is a cloud computing performance model in which the cloud provider handles the server and dynamically controls the allocation of machine resources.

It offers lots of benefits, that’s why nowadays enterprises prefer to use it. It can alter the way you do work in your business and empower it to be brisker and faster.

Omni-Cloud

Omni-cloud computing makes the application more robust and portable. It also offers top-notch connectivity; this allows multiple platforms to streamline their data.

Quantum Computing

Quantum computing focuses on building computer technology based on the origins of quantum theory. This describes the behavior of energy and material on the atomic and subatomic levels. Classical computers that we employ today can only encode information in bits.

Kubernetes

Kubernetes is an open-source container orchestration system for automating computer application deployment, scaling, and management. It was designed by Google and is now managed by the Cloud Native Computing Foundation.