CS8791 - CLOUD COMPUTING

SYLLABUS

UNIT I INTRODUCTION

Introduction to Cloud Computing –Definition of Cloud –Evolution of Cloud Computing –Underlying Principles of Parallel and Distributed Computing –Cloud Characteristics –Elasticity in Cloud –On-demand Provisioning.

UNIT II CLOUD ENABLING TECHNOLOGIES

Service Oriented Architecture – RESTful Systems – Web Services – Publish-Subscribe Model – Basics of Virtualization – Types of Virtualization – Implementation Levels of Virtualization – Virtualization Structures – Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices – Virtualization Support and Disaster Recovery.

UNIT III CLOUD ARCHITECTURE, SERVICES AND STORAGE

Layered Cloud Architecture Design – NIST Cloud Computing Reference Architecture – Public, Private and Hybrid Clouds - laaS – PaaS – SaaS – Architectural Design Challenges – Cloud Storage – Storage-as-a-Service – Advantages of Cloud Storage – Cloud Storage Providers – S3.

UNIT IV RESOURCE MANAGEMENT AND SECURITY IN CLOUD

Inter Cloud Resource Management – Resource Provisioning and Resource Provisioning Methods – Global Exchange of Cloud Resources – Security Overview – Cloud Security Challenges – Software-as-a-Service Security – Security Governance – Virtual Machine Security – IAM – Security Standards.

UNIT V CLOUD TECHNOLOGIESAND ADVANCEMENTS

Hadoop – MapReduce – Virtual Box -- Google App Engine – Programming Environment for Google App Engine – Open Stack – Federation in the Cloud – Four Levels of Federation – Federated Services and Applications – Future of Federation.

REFERENCES

1. Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012
2. Rittinghouse, JohnW., and James F. Ransome, “Cloud Computing: Implementation, Management and Security”, CRC Press, 2017.
3. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi, “Mastering Cloud Computing”, Tata Mcgraw Hill, 2013.
4. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing -A Practical Approach”, Tata Mcgraw Hill, 2009.
5. George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice)”, O'Reilly, 2009
6. Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger and Dawn Leaf, “Recommendations of the National Institute of Standards and Technology”, Special publication, NIST, U.S. Department of Commerce, 500-292.
7. Oracle Virtual Box official documentation
8. https://en.wikipedia.org/wiki/VirtualBox
9. https://docs.openstack.org/train/install/
10. https://en.wikipedia.org/wiki/OpenStack
11. https://cacoo.com/examples/network-diagram-software
12. https://www.supraits.com/infrastructure/managed-cloud/hybrid-cloud-3/cloud-computing/
13. https://www.researchgate.net/figure/Components-make-up-of-Cloud-Computing-Solution_fig1_289259494
14. https://www.telegraph.co.uk/technology/connecting-britain/colossus-bletchley-computer-broke-hitler-codes/
15. https://medium.com/penn-engineering/on-eniacs-anniversary-a-nod-to-its-female-computers-267c97a0a17
16. https://arstechnica.com/information-technology/2011/11/the-40th-birthday-ofmaybethe-first-microprocessor/
17. https://newatlas.com/anniversary-of-vannavar-bushs-famous-essay-describing-the-memex-machine/4303/
18. https://wiki.xenproject.org/wiki/Book/HelloXenProject/1-Chapter
19. https://www.safe.com/industry/natural-resources-solutions/
20. https://www.researchgate.net/figure/The-cases-of-over-provisioning-under-provisioning-and-delay-caused-by-under-provisioning_fig5_283948945
21. https://www.researchgate.net/figure/An-Enterprise-Inter-cloud-Architecture-Adapted-from-Dayananda-and-Kumar-2012_fig2_314057960
22. https://www.researchgate.net/figure/The-Hadoop-Master-Slave-Architecture-232-MapReduce-MapReduce-is-a-Hadoop-computational_fig1_274069405
23. https://mindmajix.com/hadoop-mapreduce
24. https://docs.openstack.org/train/install/
25. https://www.researchgate.net/figure/Results-of-IDC-ranking-security-challenges-3Q2009-n263_fig4_224162841

UNIT I INTRODUCTION

Introduction to Cloud Computing – Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed Computing – Cloud Characteristics – Elasticity in Cloud – On-demand Provisioning.

Introduction to Cloud Computing

- Over the last three decades, businesses that use computing resources have learned to face a vast array of buzzwords like grid computing, utility computing, autonomic computing, on-demand computing and so on.

- A new buzzword named cloud computing is presently in state-of-the-art and it is generating all sorts of confusion about what it actually means.

- In history, the term cloud has been used as a metaphor for the Internet.

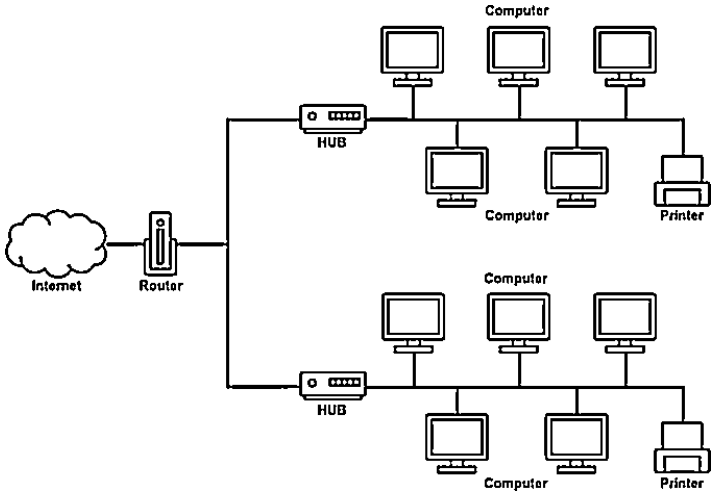


Figure 1.1 illustration of network diagram

- This usage of the term was originally derived from its common illustration in network diagrams as an outline of a cloud and the symbolic representation used to represent the transport of data across the network to an endpoint location on the other side of the network.

- Figure 1.1 illustrates the network diagram which includes the symbolic representation of cloud

- The cloud computing concepts were initiated in 1961, when Professor John McCarthy suggested that computer time-sharing technology might lead to a future where computing power and specific applications might be sold through a utility-type business model.

- This idea became very popular in the late 1960s, but in mid 1970s the idea vanished away when it became clear that the IT Industries of the day were unable to sustain such a innovative computing model. However, since the turn of the millennium, the concept has been restored.

- Utility computing is the provision of computational resources and storage resources as a metered service, similar to those provided by a traditional public utility company. This is not a new idea. This form of computing is growing in popularity, however, as companies have begun to extend the model to a cloud computing paradigm providing virtual servers that IT departments and users can access on demand.

- In early days, enterprises used the utility computing model primarily for non-mission-critical requirements, but that is quickly changing as trust and reliability issues are resolved.

- Research analysts and technology vendors are inclined to define cloud computing very closely, as a new type of utility computing that basically uses virtual servers that have been made available to third parties via the Internet.

- Others aimed to describe the term cloud computing using a very broad, all-inclusive application of the virtual computing platform. They confront that anything beyond the network firewall limit is in the cloud.

- A more softened view of cloud computing considers it the delivery of computational resources from a location other than the one from which the end users are computing.

- The cloud sees no borders and thus has made the world a much smaller place. Similar to that the Internet is also global in scope but respects only established communication paths.

- People from everywhere now have access to other people from anywhere else.

- Globalization of computing assets may be the major contribution the cloud has made to date. For this reason, the cloud is the subject of many complex geopolitical issues.

- Cloud computing is viewed as a resource available as a service for virtual data centers. Cloud computing and virtual data centers are different one.

- For example, Amazon's S3 is Simple Storage Service. This is a data storage service designed for use across the Internet. It is designed to create web scalable computing easier for developers.

- Another example is Google Apps. This provides online access via a web browser to the most common office and business applications used today. The Google server stores all the software and user data.

- Managed service providers (MSPs) offers one of the oldest form of cloud computing.

- A managed service is an application that is accessible to an organization's IT infrastructure rather than to end users which include virus scanning for email, anti spam services such as Postini, desktop management services offered by CenterBeam or Everdream, and application performance monitoring.

- Grid computing is often confused with cloud computing. Grid computing is a form of distributed computing model that implements a virtual supercomputer made up of a cluster of networked or Inter networked computers involved to perform very large tasks.

- Most of the cloud computing deployments in market today are powered by grid computing implementations and are billed like utilities, but cloud computing paradigm is evolved next step away from the grid utility model.

- The majority of cloud computing infrastructure consists of time tested and highly reliable services built on servers with varying levels of virtualized technologies, which are delivered via large scale data centers operating under various service level agreements that require 99.9999% uptime.

Definition of cloud

- Cloud computing is a model for delivering IT services in which resources are retrieved from the internet through web based tools and applications rather than a direct connection to the server.

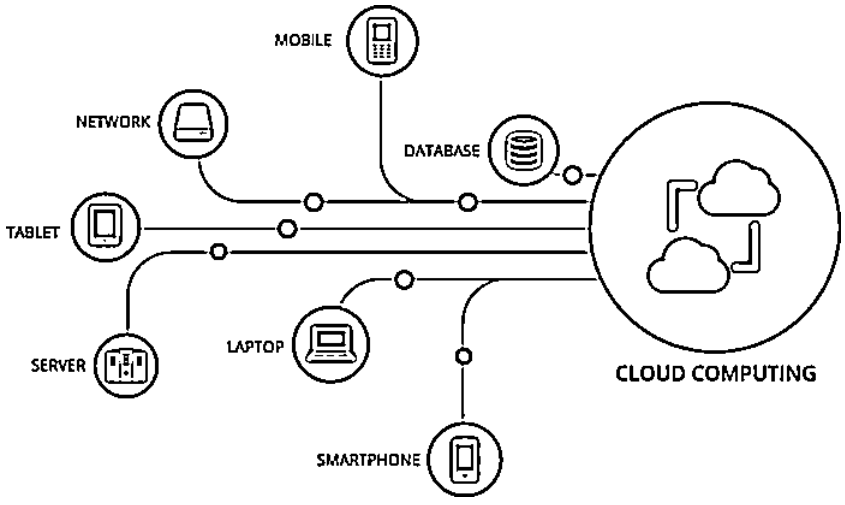


Figure 1.2 Cloud Computing Paradigm

- In other words, cloud computing is a distributed computing model over a network and means the ability to run a program on many connected components at a same time

- In the cloud computing environment, real server machines are replaced by virtual machines. Such virtual machines do not physically exist and can therefore be moved around and scaled up or down on the fly without affecting the cloud user as like a natural cloud.

- Cloud refers to software, platform, and Infrastructure that are sold as a service. The services accessed remotely through the Internet

- The cloud users can simply log on to the network without installing anything. They do not pay for hardware and maintenance. But the service providers pay for physical equipment and maintenance.

- The concept of cloud computing becomes much more understandable when one begins to think about what modern IT environments always require scalable capacity or additional capabilities to their infrastructure dynamically, without investing money in the purchase of new infrastructure, all the while without needing to conduct training for new personnel and without the need for licensing new software.

- The cloud model is composed of three components.

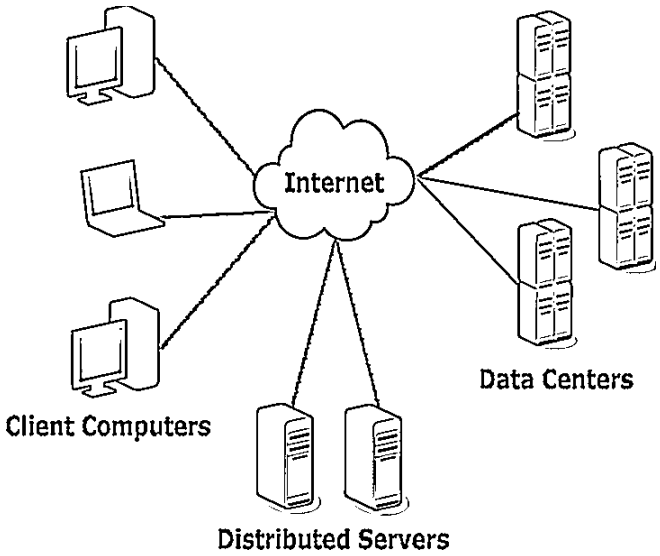


Figure 1.3 Cloud Components

- **Clients** are simple computers might be laptop, tablet, mobile phone.
- Categories of clients are Mobile clients, Thin clients and Thick clients.
- Mobile clients which includes smartphones and PDAs
- Thin clients which include servers without internal hardware. Usage of this type of clients leads to Low hardware cost, Low IT Cost, Less power consumption and less noise.
- Thick clients which includes regular computers.
- **Data Center** is a collection of servers and it contains clients requested applications.
- **Distributed Server** in which server is distributed in different geographical locations

Evolution of Cloud Computing

- It is important to understand the evolution of computing in order to get an appreciation of how IT based environments got into the cloud environment. Looking at the evolution of the computing hardware itself, from the first generation to the fourth generation of computers, shows how the IT industry's got from there to here.

- The hardware is a part of the evolutionary process. As hardware evolved, so did the software. As networking evolved, so did the rules for how computers communicate. The development of such rules or protocols, helped to drive the evolution of Internet software.

- Establishing a common protocol for the Internet led directly to rapid growth in the number of users online.

- Today, enterprises discuss about the uses of IPv6 (Internet Protocol version 6) to ease addressing concerns and for improving the methods used to communicate over the Internet.

- Usage of web browsers led to a stable migration away from the traditional data center model to a cloud computing based model. And also, impact of technologies such as

server virtualization, parallel processing, vector processing, symmetric multiprocessing, and massively parallel processing fueled radical change in IT era.

1.3.1 Hardware Evolution

- The first step along with the evolutionary path of computers was occurred in 1930, when the first binary arithmetic was developed and became the foundation of computer processing technology, terminology, and programming languages.

- Calculating devices date back to at least as early as 1642, when a device that could mechanically add numbers was invented.

- Adding devices were evolved from the abacus. This evolution was one of the most significant milestones in the history of computers.

- In 1939, the Berry brothers were invented an electronic computer that capable of operating digital aspects. The computations were performed using vacuum tube technology.

- In 1941, the introduction of Z3 at the German Laboratory for Aviation purpose in Berlin was one of the most significant events in the evolution of computers because Z3 machine supported both binary arithmetic and floating point computation. Because it was a “Turing complete” device, it is considered to be the very first computer that was fully operational.

1.3.1.1 First Generation Computers

- The first generation of modern computers traced to 1943, when the Mark I and Colossus computers were developed for fairly different purposes.

- With financial support from IBM, the Mark I was designed and developed at Harvard University. It was a general purpose electro, mechanical, programmable computer.

- Colossus is an electronic computer built in Britain at the end 1943. Colossus was the world's first programmable, digital, electronic, computing device.

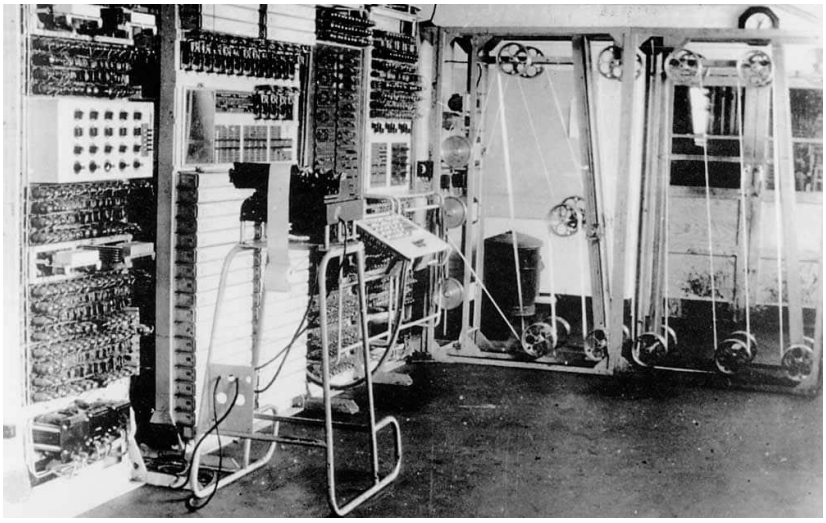


Figure 1.4 Colossus

- In general, First generation computers were built using hard-wired circuits and vacuum tubes.

- Data were stored using paper punch cards.

1.3.1.2 Second Generation Computers

- Another general-purpose computer of this era was ENIAC (Electronic Numerical Integrator and Computer), which was built in 1946. This was the first Turing complete, digital computer that capable of reprogramming to solve a full range of computing problems.

- ENIAC composed of 18,000 thermionic valves, weighed over 60,000 pounds, and consumed 25 kilowatts of electrical power per hour. ENIAC was capable of performing one lakh calculations a second.

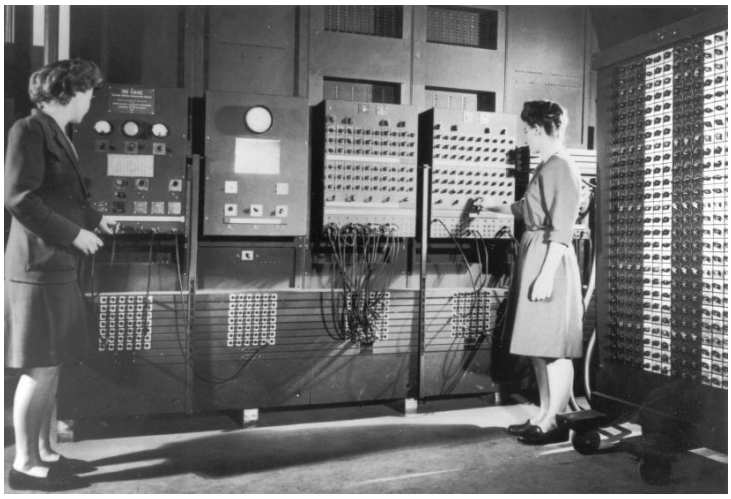


Figure 1.5 ENIAC

- Transistorized computers marked the initiation of second generation computers, which dominated in the late 1950s and early 1960s. The computers were used mainly by universities and government agencies.

- The integrated circuit or microchip was developed by Jack St. Claire Kilby, an achievement for which he received the Nobel Prize in Physics in 2000.

1.3.1.3 Third Generation Computers

- Claire Kilby's invention initiated an explosion in third generation computers. Even though the first integrated circuit was produced in 1958, microchips were not used in programmable computers until 1963.

- In 1971, Intel released the world's first commercial microprocessor called Intel 4004.

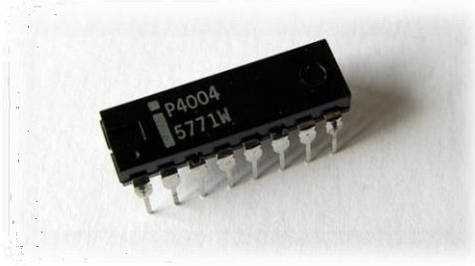


Figure 1.6 Intel 4004

- Intel 4004 was the first complete CPU on one chip and became the first commercially available microprocessor. It was possible because of the development of new silicon gate technology that enabled engineers to integrate a much greater number of transistors on a chip that would perform at a much faster speed.

1.3.1.4 Fourth Generation Computers

- The fourth generation computers that were being developed at this time utilized a microprocessor that put the computer's processing capabilities on a single integrated circuit chip.

- By combining random access memory, developed by Intel, fourth generation computers were faster than ever before and had much smaller footprints.

- The first commercially available personal computer was the MITS Altair 8800, released at the end of 1974. What followed was a flurry of other personal computers to market, such as the Apple I and II, the Commodore PET, the VIC-20, the Commodore 64, and eventually the original IBM PC in 1981. The PC era had begun in earnest by the mid-1980s.

- Even though microprocessing power, memory and data storage capacities have increased by many orders of magnitude since the invention of the 4004 processor, the technology for Large Scale Integration (LSI) or Very Large Scale Integration (VLSI) microchips has not changed all that much.

- For this reason, most of today's computers still fall into the category of fourth generation computers.

1.3.2 Internet Software Evolution

- The Internet is named after the evolution of Internet Protocol which is the standard communications protocol used by every computer on the Internet.

- Vannevar Bush was written a visionary description of the potential uses for information technology with his description of an automated library system called MEMEX.

- Bush introduced the concept of the MEMEX in late 1930s as a microfilm based device in which an individual can store all his books and records.

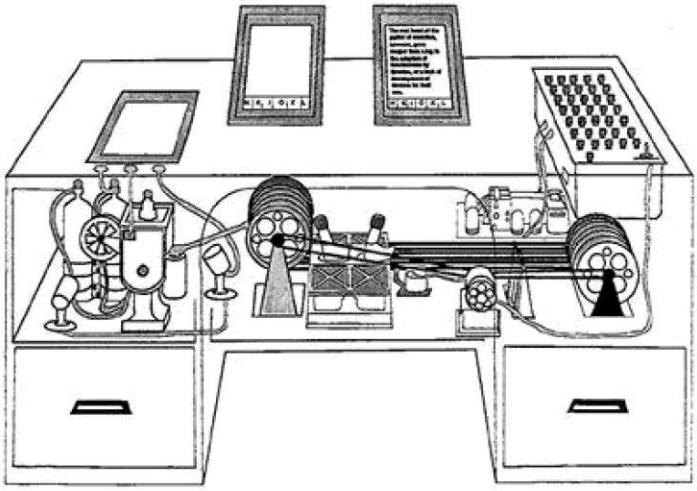


Figure 1.7 MEMEX Systems

- The second individual who has shaped the Internet was Norbert Wiener.

- Wiener was an early pioneer in the study of stochastic and noise processes. Norbert Wiener work in stochastic and noise processes was relevant to electronic engineering, communication, and control systems.

- SAGE refers Semi Automatic Ground Environment. SAGE was the most ambitious computer project and started in the mid 1950s and became operational by 1963. It remained in continuous operation for over 20 years, until 1983.

- A minicomputer was invented specifically to realize the design of the Interface Message Processor (IMP). This approach provided a system independent interface to the ARPANET.

- The IMP would handle the interface to the ARPANET network. The physical layer, the data link layer, and the network layer protocols used internally on the ARPANET were implemented using IMP.

- Using this approach, each site would only have to write one interface to the commonly deployed IMP.

- The first networking protocol that was used on the ARPANET was the Network Control Program (NCP). The NCP provided the middle layers of a protocol stack running on an ARPANET connected host computer.

- The lower-level protocol layers were provided by the IMP host interface, the NCP essentially provided a transport layer consisting of the ARPANET Host-to-Host Protocol (AHHP) and the Initial Connection Protocol (ICP).

- The AHHP defines how to transmit a unidirectional and flow controlled stream of data between two hosts.

- The ICP specifies how to establish a bidirectional pair of data streams between a pair of connected host processes.

- Robert Kahn and Vinton Cerf who built on what was learned with NCP to develop the TCP/IP networking protocol commonly used nowadays. TCP/IP quickly became the most widely used network protocol in the world.

- Over time, there evolved four increasingly better versions of TCP/IP (TCP v1, TCP v2, a split into TCP v3 and IP v3, and TCP v4 and IPv4). Now, IPv4 is the standard protocol, but it is in the process of being replaced by IPv6.

- The amazing growth of the Internet throughout the 1990s caused a huge reduction in the number of free IP addresses available under IPv4. IPv4 was never designed to scale to global levels. To increase available address space, it had to process data packets that were larger.

- After examining a number of proposals, the Internet Engineering Task Force (IETF) settled on IPv6, which was released in early 1995 as RFC 1752. IPv6 is sometimes called the Next Generation Internet Protocol (IPNG) or TCP/IP v6.

1.3.3 Server Virtualization

- Virtualization is a method of running multiple independent virtual operating systems on a single physical computer. This approach maximizes the return on investment for the computer.

- The creation and management of virtual machines has often been called platform virtualization.

- Platform virtualization is performed on a given computer (hardware platform) by software called a control program.

- Parallel processing is performed by the simultaneous execution of multiple program instructions that have been allocated across multiple processors with the objective of running a program in less time.

- The next advancement in parallel processing was multiprogramming.

- In a multiprogramming system, multiple programs submitted by users are allowed to use the processor for a short time, each taking turns and having exclusive time with the processor in order to execute instructions.

- This approach is called as round robin scheduling (RR scheduling). It is one of the oldest, simplest, fairest, and most widely used scheduling algorithms, designed especially for time-sharing systems.

- Vector processing was developed to increase processing performance by operating in a multitasking manner.

- Matrix operations were added to computers to allow a single instruction to manipulate two arrays of numbers performing arithmetic operations. This was valuable in certain types of applications in which data occurred in the form of vectors or matrices.

- The next advancement was the development of symmetric multiprocessing systems (SMP) to address the problem of resource management in master or slave models. In SMP systems, each processor is equally capable and responsible for managing the workflow as it passes through the system.

- Massive parallel processing (MPP) is used in computer architecture circles to refer to a computer system with many independent arithmetic units or entire microprocessors, which run in parallel.

1.2 Principles of Parallel and Distributed Computing

- The two fundamental and dominant models of computing environment are sequential and parallel. The sequential computing era was begun in the 1940s. The parallel and distributed computing era was followed it within a decade.

- The four key elements of computing developed during these eras are architectures, compilers, applications, and problem solving environments.

- Every aspect of this era will undergo a three phase process.

    - Research and Development (R&D)
    - Commercialization
    - Commoditization

1.4.1 Parallel vs distributed computing

- The terms parallel computing and distributed computing are often used interchangeably, even though which meant somewhat different things.

- The term parallel implies a tightly coupled system, whereas distributed refers to a wider class of system which includes tightly coupled systems.

- More specifically, the term parallel computing refers to a model in which the computation is divided among several processors which sharing the same memory.

- The architecture of a parallel computing system is often characterized by the homogeneity of components.

- In parallel computing paradigm, each processor is of the same type and it has the same capability. The shared memory has a single address space, which is accessible to all the processors.

- Processing of multiple tasks simultaneously on multiple processors is called as parallel processing.

- The parallel program consists of multiple active processes or tasks simultaneously solving a given problem.

- A given task is divided into multiple subtasks using a divide and conquer technique, and each subtask is processed on a different Central Processing Unit (CPU).

- Programming on a multiprocessor system using the divide and conquer technique is called parallel programming.

- The term distributed computing encompasses any architecture or system that allows the computation to be broken down into units and executed concurrently on different computing elements, whether these are processors on different nodes, processors on the same computer, or cores within the same processor.

- Therefore, distributed computing includes a wider range of systems and applications than parallel computing and is often considered a most common term.

1.4.2 Elements of parallel computing

- The core elements of parallel processing are CPUs. Based on the number of instruction streams and data streams that can be processed simultaneously, computing systems are classified into four categories proposed by Michael J. Flynn in 1966.

- Single Instruction Single Data systems (SISD)
- Single Instruction Multiple Data systems (SIMD)
- Multiple Instruction Single Data systems (MISD)
- Multiple Instruction, Multiple Data systems (MIMD)

- An SISD computing system is a uniprocessor system capable of executing a single instruction, which operates on a single data stream.

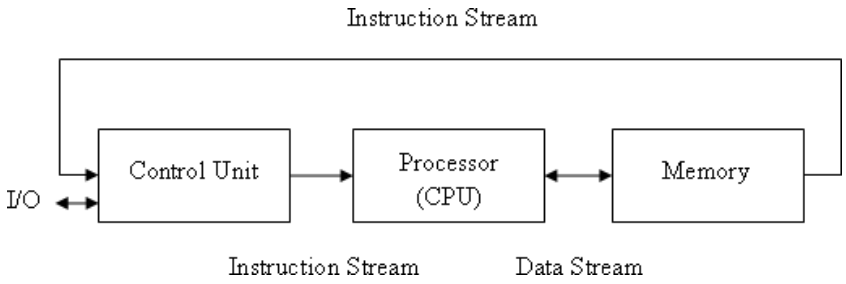


Figure 1.8 SISD

- An SIMD computing system is a multiprocessor system capable of executing the single instruction on all the CPUs but operating on different data streams.

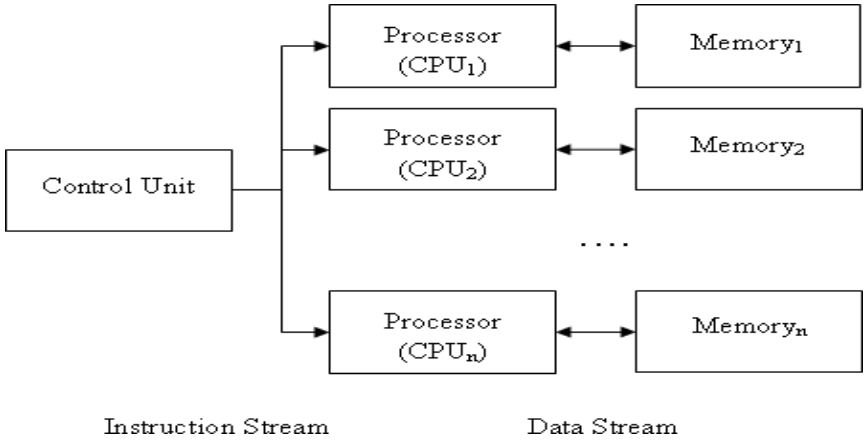


Figure 1.9 SIMD

- An MISD computing system is a multiprocessor system capable of executing different instructions on different processing elements but all of them operating on the same data streams.

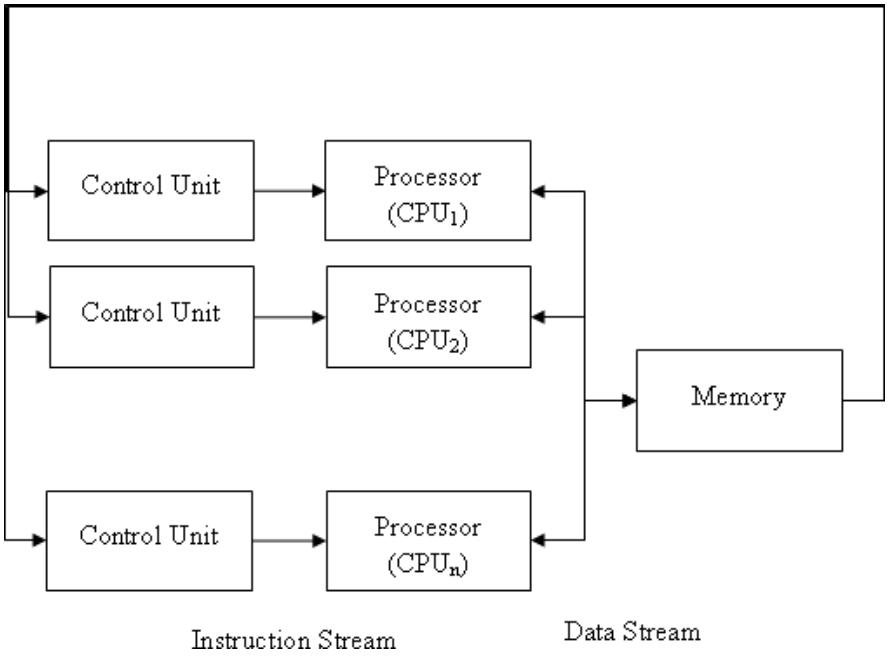


Figure 1.10 MISD

- An MIMD computing system is a multiprocessor system capable of executing multiple instructions on multiple data streams.

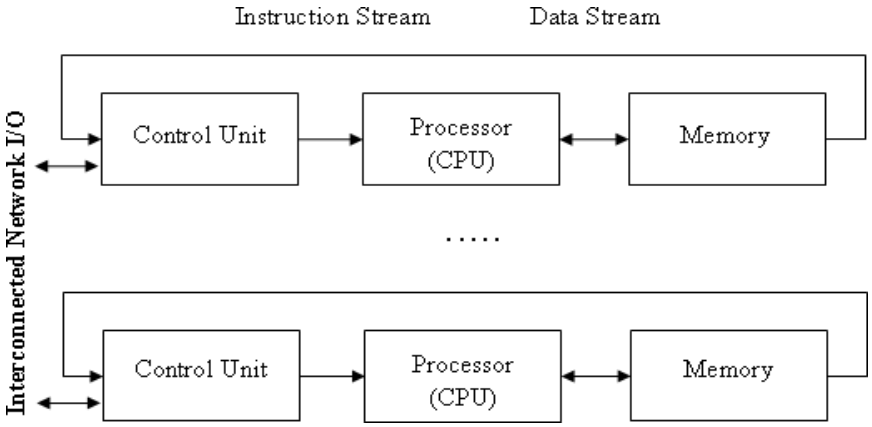


Figure 1.11 MIMD

- MIMD systems are broadly categorized into shared memory MIMD and distributed memory MIMD based on the way processing elements are coupled to the main memory.

- In the shared memory MIMD model, all the processing elements are connected to a single global memory and they all have access to it.

- In the distributed memory MIMD model, all processing elements have a local memory. Systems based on this model are also called loosely coupled multiprocessor systems.

- In general, Failures in a shared memory MIMD affects the entire system, where as this is not the case of the distributed model, in which each of the processing elements can be easily isolated.

- A wide variety of parallel programming approaches are available in computing environment. The most prominent among them are the following:

    - Data parallelism
    - Process parallelism
    - Farmer-and-worker model

- In data parallelism, the divide and conquer methodology is used to split data into multiple sets, and each data set is processed on different processing elements using the same instruction.

- In process parallelism, a given operation has multiple distinct tasks that can be processed on multiple processors.

- In farmer and worker model, a job distribution approach is used in which one processor is configured as master and all other remaining processing elements are designated as slaves. The master assigns jobs to slave processing elements and, on completion, they inform the master, which in turn collects results.

- Parallelism within an application can be detected at several levels such as Large grain (or task level), Medium grain (or control level), Fine grain (data level), Very fine grain (multiple-instruction issue)

- Speed of computation is never increase linearly. It is proportional to the square root of system cost. Therefore, the faster a system becomes, the more expensive it is to increase its speed.

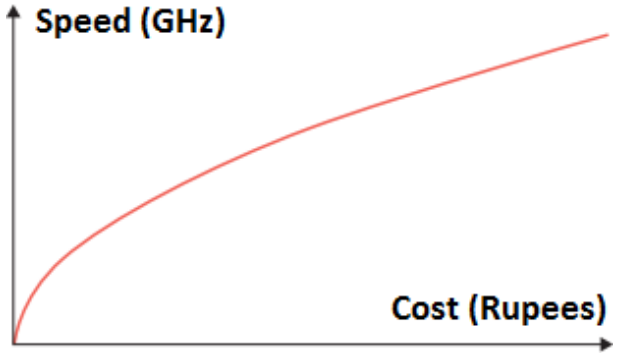


Figure 1.12 Cost versus Speed

- Speed by a parallel computer increases as the logarithm of the number of processors (i.e.,y= klog(N)).

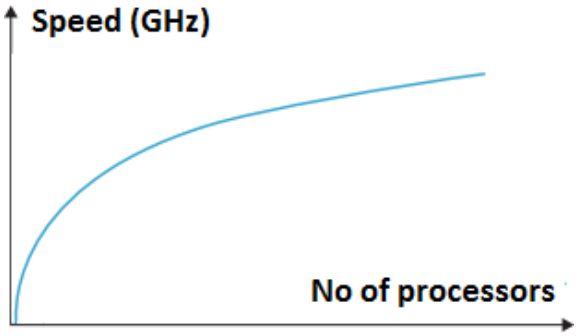


Figure 1.13 No of processors versus Speed

1.4.3 Elements of distributed computing

- A distributed system is the collection of independent computers that appears to its users as a single coherent system.

- A distributed system is the result of the interaction of several components that pass through the entire computing stack from hardware to software.

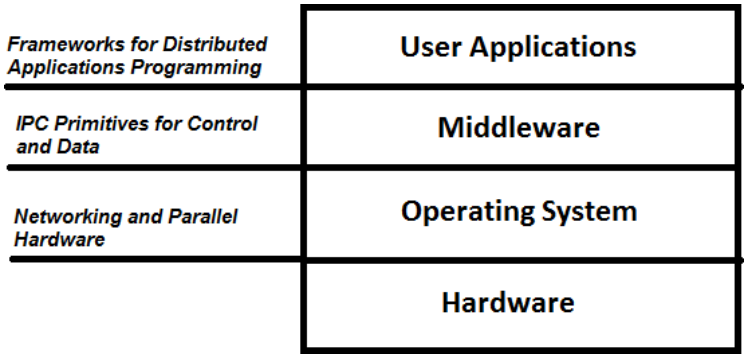


Figure 1.14 A layered view of a distributed system

- At the very bottom layer, computer and network hardware constitute the physical infrastructure.

- The hardware components are directly managed by the operating system, which provides the basic services for inter process communication (IPC), process scheduling and management, and resource management in terms of file system and local devices.

- The use of well-known standards at the operating system level and even more at the hardware and network levels allows easy harnessing of heterogeneous components and their organization into a coherent and uniform system.

- The middleware layer leverages such services to build a uniform environment for the development and deployment of distributed applications.

- The top of the distributed system stack is represented by the applications and services designed and developed to use the middleware.

- In distributed computing, Architectural styles are mainly used to determine the vocabulary of components and connectors that are used as instances of the style together with a set of constraints on how they can be combined.

- Architectural styles are classified into two major classes.

    - Software architectural styles
    - System architectural styles

- The first class relates to the logical organization of the software.

- The second class includes all those styles that describe the physical organization of distributed software systems in terms of their major components.

- A component represents a unit of software that encapsulates a function or a feature of the system. Examples of components can be programs, objects, processes, pipes, and filters.

- A connector is a communication mechanism that allows cooperation and coordination among components. Differently from components, connectors are not encapsulated in a single entity, but they are implemented in a distributed manner over many system components.

- Software architectural styles are based on the logical arrangement of software components.

- According to Garlan and Shaw, architectural styles are classified as shown in Table 1.1

Category	**Most Common Architectural Styles**
Data-centered	• Repository • Blackboard
Data flow	• Pipe and filter • Batch sequential
Virtual machine	• Rule-based system • Interpreter

Call and return	• Top down systems • Object oriented systems • Layered systems
Independent components	• Communicating processes • Event system

Table 1.1 Software Architectural Styles

- The repository architectural style is the most relevant reference model in this category. It is characterized by two main components: the central data structure, which represents the current state of the system, and a collection of independent components, which operate on the central data.

- The batch sequential style is characterized by an ordered sequence of separate programs executing one after the other. These programs are chained together by providing as input for the next program the output generated by the last program after its completion, which is most likely in the form of a file.

- The pipe and filter style is a variation of the previous style for expressing the activity of a software system as a sequence of data transformations. Each component of the processing chain is called a filter, and the connection between one filter and the next is represented by a data stream.

- Rule-Based Style architecture is characterized by representing the abstract execution environment as an inference engine. Programs are expressed in the form of rules or predicates that hold true.

- The core feature of the interpreter style is the presence of an engine that is used to interpret a pseudo code expressed in a format acceptable for the interpreter. The interpretation of the pseudo-program constitutes the execution of the program itself.

- Top Down Style is quite representative of systems developed with imperative programming, which leads to a divide and conquer approach to problem resolution.

- Object Oriented Style encompasses a wide range of systems that have been designed and implemented by leveraging the abstractions of object oriented programming

- The layered system style allows the design and implementation of software systems in terms of layers, which provide a different level of abstraction of the system.

- Each layer generally operates with at most two layers: the one that provides a lower abstraction level and the one that provides a higher abstraction layer.

- In Communicating Processes architectural style, components are represented by independent processes that leverage IPC facilities for coordination management.

- On the other hand, Event Systems architectural style where the components of the system are loosely coupled and connected.

- System architectural styles cover the physical organization of components and processes over a distributed infrastructure. They provide two fundamental reference styles: client/server and peer-to-peer.

- The client/server model features two major components: a server and a client. These two components interact with each other through a network connection using a given protocol. The communication is unidirectional. The client issues a request to the server, and after processing the request the server returns a response.

- The important operations in the client-server paradigm are request, accept (client side), and listen and response (server side).

- The client/server model is suitable in many-to-one scenarios.

- In general, multiple clients are interested in such services and the server must be appropriately designed to efficiently serve requests coming from different clients. This consideration has implications on both client design and server design.

- For the client design, there are two models: Thin client model and Fat client model.

- Thin client model, the load of data processing and transformation is put on the server side, and the client has a light implementation that is mostly concerned with retrieving and returning the data it is being asked for, with no considerable further processing.

- Fat client model, the client component is also responsible for processing and transforming the data before returning it to the user, whereas the server features a fairly light implementation that is mostly concerned with the management of access to the data.

- The three major components in the client-server model are presentation, application logic, and data storage.

- Presentation, application logic, and data maintenance can be seen as conceptual layers, which are more appropriately called tiers.

- The mapping between the conceptual layers and their physical implementation in modules and components allows differentiating among several types of architectures, which go under the name of multi-tiered architectures.

- Two major classes are Two-tier architecture and Three-tier architecture.

- Two-tier architecture partitions the systems into two tiers, which are located one in the client component and the other on the server. The client is responsible for the presentation tier by providing a user interface. The server concentrates the application logic and the data store into a single tier.

- Three-tier architecture separates the presentation of data, the application logic, and the data storage into three tiers. This architecture is generalized into an N-tier model in case it is necessary to further divide the stages composing the application logic and storage tiers.

- The peer-to-peer model introduces a symmetric architecture in which all the components are called as peers, play the same role and incorporate both client and server capabilities of the client/server model.

- The most relevant example of peer-to-peer systems is constituted by file sharing applications such as Gnutella, BitTorrent, and Kazaa.

1.4.4 Models for inter process communication

- There are several different models in which processes can interact with each other; these map to different abstractions for IPC. Among the most relevant models are shared memory, remote procedure call (RPC), and message passing.

- Message passing introduces the concept of a message as the main abstraction of the model. The entities exchanging information explicitly encode in the form of a message the data to be exchanged. The structure and the content of a message vary according to the model. Examples of this model are the Message-Passing Interface (MPI) and OpenMP.

- Remote procedure call paradigm extends the concept of procedure call beyond the boundaries of a single process, thus triggering the execution of code in remote processes. In this case, underlying client/server architecture is implied. A remote process hosts a server component, thus allowing client processes to request the invocation of methods, and returns the result of the execution.

1.4.5 Models for message-based communication

Point-to-point message model

- This model organizes the communication among single components. Each message is sent from one component to another, and there is a direct addressing to identify the message receiver. In a point-to-point communication model it is necessary to know the location of or how to address another component in the system.

Publish-and-subscribe message model

- This model introduces a different strategy, one that is based on notification among components.

- There are two major roles: the publisher and the subscriber.

- There are two major strategies for dispatching the event to the subscribers:
    - Push strategy. In this case it is the responsibility of the publisher to notify all the subscribers. For example, with a method invocation.
    - Pull strategy. In this case the publisher simply makes available the message for a specific event, and it is the responsibility of the subscribers to check whether there are messages on the events that are registered.

Request-reply message model

- The request-reply message model identifies all communication models in which, for each message sent by a process, there is a reply.

- This model is quite popular and provides a different classification that does not focus on the number of the components involved in the communication but rather on how the dynamic of the interaction evolves.

1.3 Technologies for distributed computing

Remote procedure call

- RPC is the fundamental abstraction enabling the execution of procedures on client's request.

- RPC allows extending the concept of a procedure call beyond the boundaries of a process and a single memory address space.

- The called procedure and calling procedure may be on the same system or they may be on different systems in a network.

- An important aspect of RPC is marshaling, which identifies the process of converting parameters and return values into a form that is more suitable to be transported over a network through a sequence of bytes. The term unmarshaling refers to the opposite procedure.

Distributed object frameworks

- Distributed object frameworks extend object-oriented programming systems by allowing objects to be distributed across a heterogeneous network and provide facilities so that they can coherently act as though they were in the same address space.

Service-oriented computing

- Service-oriented computing organizes distributed systems in terms of services, which represent the major abstraction for building systems.

- Service orientation expresses applications and software systems as aggregations of services that are coordinated within a service-oriented architecture (SOA).

- SOA is an architectural style supporting service orientation. It organizes a software system into a collection of interacting services.

- SOA encompasses a set of design principles that structure system development and provide means for integrating components into a coherent and decentralized system.

- SOA-based computing packages functionalities into a set of interoperable services, which can be integrated into different software systems belonging to separate business domains.

- There are two major roles within SOA: the service provider and the service consumer.

1.5 Cloud Characteristics

From the cloud computing's various definitions; a certain set of key characteristics emerges. Figure 1.15 illustrates various key characteristics related to cloud computing paradigm.

1.5.1 On-demand Provisioning

- On-demand provisioning is the single most important characteristic of cloud computing, it allows the users to request or release resources whenever they want.

- These demands are thereafter automatically granted by a cloud provider's service and the users are only charged for their usage, i.e., the time they were in possession of the resources.

- The reactivity of a cloud solution, with regard to resource provisioning is indeed of prime importance as it is closely related to the cloud's pay-as-you-go business model.

- It is one of the important and valuable features of Cloud Computing as the user can continuously monitor the server uptime, capabilities, and allotted network storage. With this feature, the user can also monitor the computing capabilities.

1.5.2 Universal Access

- Resources in the cloud need not only be provisioned rapidly but also accessed and managed universally, using standard Internet protocols, typically via RESTful web services.

- This enables the users to access their cloud resources using any type of devices, provided they have an Internet connection.

- Universal access is a key feature behind the cloud's widespread adoption, not only by professional actors but also by the general public that is nowadays familiar with cloud based solutions such as cloud storage or media streaming.

- Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms such as mobile phones, tablets, laptops, and workstations.

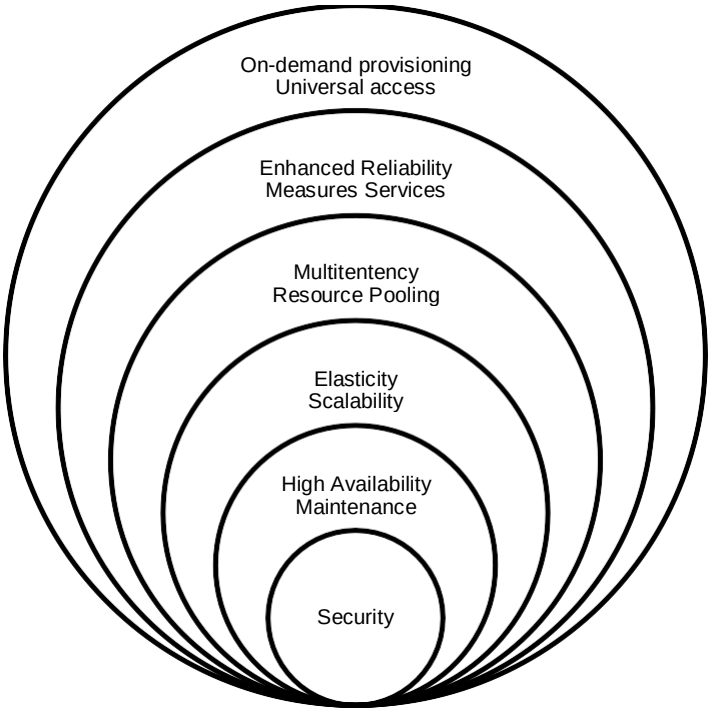


Figure 1.15 Cloud Characteristics

1.5.3 Enhanced Reliability

- Cloud computing enables the users to enhance the reliability of their applications.

- Reliability is already built in many cloud solutions via storage redundancy.

- Cloud providers usually have more than one data center and further reliability can be achieved by backing data up in different locations.

- This can also be used to ensure service availability, in the case of routine maintenance operations or the rarer case of a natural disaster.

- The user can achieve further reliability using the services of different cloud providers.

1.5.4 Measured Services

- Cloud computing refers generally to paid services.
- The customers are entitled to a certain quality of service, guaranteed by the Service Level Agreement that they should be able to supervise.
- Therefore, cloud providers offer monitoring tools, either using a graphical interface or via an API.
- These tools also help the providers themselves for billing and management purposes.

1.5.5 Multitenancy

- As the grid before, the cloud's resources are shared by different simultaneous users. These users had to reserve in advance a fixed number of physical machines for a fixed amount of time.
- In virtualized data centers, a user's provisioned resources no longer correspond to the physical infrastructure and can be dispatched over multiple physical machines.
- They can also run alongside other users' provisioned resources thus requiring a lesser amount of physical resources. Consequently, important energy savings can be made by shutting down the unused resources or putting them in energy saving mode.

1.5.6 Resource pooling

- The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

- There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).

- Examples of resources include storage, processing, memory, and network bandwidth.

1.5.7 Rapid elasticity and Scalability

- Elasticity is the ability of a system to include and exclude resources like CPU cores, memory, Virtual Machine and container instances to adapt to the load variation in real time.

- Elasticity is a dynamic property for cloud computing. There are two types of elasticity. Horizontal and Vertical.

- Horizontal elasticity consists in adding or removing instances of computing resources associated with an application.

- Vertical elasticity consists in increasing or decreasing characteristics of computing resources, such as CPU time, cores, memory, and network bandwidth.

- There are other terms such as scalability and efficiency, which are associated with elasticity but their meaning is different from elasticity while they are used interchangeably in some cases.

- Scalability is the ability of the system to sustain increasing workloads by making use of additional resources, it is time independent and it is similar to the provisioning state in elasticity but the time has no effect on the system (static property).

- The following equation that summarizes the elasticity concept in cloud computing.

  Auto scaling = Scalability +Automation
  Elasticity = Auto scaling + Optimization

- It means that the elasticity is built on top of scalability. It can be considered as an automation of the concept of scalability, however, it aims to optimize at best and as quickly as possible the resources at a given time.

- Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.

- To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

1.5.8 Easy Maintenance

- The servers are easily maintained and the downtime is very low and even in some cases, there is no downtime.

- Cloud Computing comes up with an update every time by gradually making it better. The updates are more compatible with the devices and perform faster than older ones along with the bugs which are fixed.

1.5.9 High Availability

- The capabilities of the Cloud can be modified as per the use and can be extended a lot. It analyzes the storage usage and allows the user to buy extra Cloud storage if needed for a very small amount.

1.5.10 Security

- Cloud Security is one of the best features of cloud computing. It creates a snapshot of the data stored so that the data may not get lost even if one of the servers gets damaged.

- The data is stored within the storage devices, which cannot be hacked and utilized by any other person. The storage service is quick and reliable.

TWO MARK QUESTIONS

1. Define utility computing.

   - Utility computing is the provision of computational resources and storage resources as a metered service, similar to those provided by a traditional public utility company.
   - This is not a new idea.
   - This form of computing is growing in popularity, however, as companies have begun to extend the model to a cloud computing paradigm providing virtual servers that IT departments and users can access on demand.

2. What is Grid Computing?

   - Grid computing is often confused with cloud computing.
   - Grid computing is a form of distributed computing model that implements a virtual supercomputer made up of a cluster of networked or Inter networked computers involved to perform very large tasks.

3. Define Cloud computing.

   - Cloud computing is a model for delivering IT services in which resources are retrieved from the internet through web based tools and applications rather than a direct connection to the server.

4. Define Cloud.

   - Cloud refers to software, platform, and Infrastructure that are sold as a service. The services accessed remotely through the Internet

   - The cloud users can simply log on to the network without installing anything. They do not pay for hardware and maintenance. But the service providers pay for physical equipment and maintenance.

5. What is the purpose of NCP?

    - The first networking protocol that was used on the ARPANET was the Network Control Program (NCP).
    - The NCP provided the middle layers of a protocol stack running on an ARPANET connected host computer.

6. How to increase the performance using multiprogramming?

    - In a multiprogramming system, multiple programs submitted by users are allowed to use the processor for a short time, each taking turns and having exclusive time with the processor in order to execute instructions.
    - This approach is called as round robin scheduling

7. Differentiate between Vector processing and Massive parallel processing

    - Vector processing was developed to increase processing performance by operating in a multitasking manner.
    - Massive parallel processing (MPP) is used in computer architecture circles to refer to a computer system with many independent arithmetic units or entire microprocessors, which run in parallel.

8. List the four key elements in parallel and distributed computing.

    - The four key elements of computing developed during these eras are architectures, compilers, applications, and problem solving environments.

9. Differentiate between parallel and distributed computing.

    - The terms parallel computing and distributed computing are often used interchangeably, even though which meant somewhat different things. Parallel implies a tightly coupled system, whereas distributed refers to a wider class of system which includes tightly coupled systems.

- The term distributed computing encompasses any architecture or system that allows the computation to be broken down into units and executed concurrently on different computing elements, whether these are processors on different nodes, processors on the same computer, or cores within the same processor.

10. Categorize computing systems base on Flynn's classification.

- Single Instruction Single Data systems (SISD)
- Single Instruction Multiple Data systems (SIMD)
- Multiple Instruction Single Data systems (MISD)
- Multiple Instruction, Multiple Data systems (MIMD)

11. List the most prominent parallel programming approaches.

- Data parallelism
- Process parallelism
- Farmer-and-worker model

12. What is farmer and worker model?

- A job distribution approach is used in which one processor is configured as master and all other remaining processing elements are designated as slaves.
- The master assigns jobs to slave processing elements and, on completion, they inform the master, which in turn collects results.

13. Differentiate between component and connector.

- A component represents a unit of software that encapsulates a function or a feature of the system.
  - A connector is a communication mechanism that allows cooperation and coordination among components.

14. Classify architectural styles according to Garlan and Shaw.

- Data-centered
- Data flow
- Virtual machine
- Call and return
- Independent components

15. What is repository architectural style?

- The repository architectural style is the most relevant reference model in this category.
- It is characterized by two main components: the central data structure, which represents the current state of the system, and a collection of independent components, which operate on the central data.

16. When the computing paradigm adapt client / server mode?

- The client/server model is suitable in many-to-one scenarios.
- The client/server model features two major components: a server and a client.
- These two components interact with each other through a network connection using a given protocol.
- The communication is unidirectional.

17. Differentiate between Thin client and Fat client model.

- Thin client model, the load of data processing and transformation is put on the server side, and the client has a light implementation that is mostly concerned with retrieving and returning the data it is being asked for, with no considerable further processing.

- Fat client model, the client component is also responsible for processing and transforming the data before returning it to the user.

18. Differentiate between two tier and three tier architecture.

- Two-tier architecture partitions the systems into two tiers, which are located one in the client component and the other on the server.

- Three-tier architecture separates the presentation of data, the application logic, and the data storage into three tiers.

19. What is point-to-point model?

- This model organizes the communication among single components.
- Each message is sent from one component to another, and there is a direct addressing to identify the message receiver.
- In a point-to-point communication model it is necessary to know the location of or how to address another component in the system.

20. List the strategies for dispatching the event to the subscribers

- Push strategy. In this case it is the responsibility of the publisher to notify all the subscribers.
- Pull strategy. In this case the publisher simply makes available the message for a specific event.

21. What is request-reply model?

- The request-reply message model identifies all communication models in which, for each message sent by a process, there is a reply.

- This model is quite popular and provides a different classification that does not focus on the number of the components involved in the communication but rather on how the dynamic of the interaction evolves.

22. What is the purpose of Distributed object frameworks?

- Distributed object frameworks extend object-oriented programming systems by allowing objects to be distributed across a heterogeneous network and provide facilities so that they can coherently act as though they were in the same address space.

23. List the key characteristics of cloud.

- On-demand provisioning Universal access
- Enhanced Reliability Measures Services
- Multitentency Resource Pooling
- Elasticity Scalability
- High Availability Maintenance
- Security

UNIT II CLOUD ENABLING TECHNOLOGIES

Service Oriented Architecture –REST and Systems of Systems –Web Services –Publish-Subscribe Model –Basics of Virtualization –Types of Virtualization –Implementation Levels of Virtualization –Virtualization Structures –Tools and Mechanisms –Virtualization of CPU –Memory –I/O Devices –Virtualization Support and Disaster Recovery.

2.1 Service Oriented Architecture

- A service encapsulates a software component that gives a set of coherent and related functionalities that can be reused and integrated into larger and more complex applications.

- The term service is a general abstraction that encompasses several different implementations using different technologies and protocols.

- Don Box identifies four major characteristics with the intention of identify a service.

- *Boundaries are explicit*

    - A service oriented applications are generally composed of services that are spread across different domains, trust authorities and execution environments.

- *Services are autonomous*

    - Services are components that exist to offer functionality.
    - Services are aggregated and coordinated to build more complex system.
    - Services are not designed to be part of a specific system but they can be integrated in several software systems.
    - The notion of autonomy also affects the way services handle failures.

- *Services share schema and contracts*

    - Services never share class and interface definitions.
    - In object oriented systems, services are not expressed in terms of classes or interfaces but they define in terms of schemas and contracts.
    - Technologies such as XML and SOAP provide the appropriate tools to support such features rather than class definition and an interface declaration.

- *Services compatibility is determined based on policy*

    - Service orientation separates structural compatibility from semantic compatibility.
    - Structural compatibility is based on contracts and schema and can be validated by machine based techniques.
    - Semantic compatibility is expressed in the form of policies that define the capabilities and requirements for a service.

- Service Oriented architecture is an architectural style supporting service orientation.

- This architectural style organizes a software system into a collection of interacting services.

- SOA encompasses a set of design principles that structure system development and provide means for integrating components into a coherent and decentralized system.

- SOA based computing packages functionalities into a set of interoperable services, which can be integrated into different software systems belonging to separate business domains.

- There are two major roles exist in SOA

    - Service provider
    - Service consumer

- First, the service provider is the maintainer of the service and the organization that makes available one or more services for others to use.

- To advertise services, the provider can publish them in a registry along with a service contract that specifies the nature of the service, how to use the service, the requirements for the service and the fees charged.

- Second, the service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.

- Service providers and consumers can belong to different organization bodies.

- It is very common in SOA based computing systems that components play the roles of both service provider and service consumer.

- Services might aggregate information and data retrieved from other services or create workflows of services to satisfy the request of a given service consumer. This practice is called as service orchestration, which more generally describes the automated arrangement, coordination and management of more complex computer systems, middleware and services.

- Another important interaction pattern is service composition is the coordinated interaction of services without a single point of control.

- SOA provides a reference model for architecting several software systems primarily for enterprise business applications and systems.

- Interoperability, standards and service contracts plays a fundamental role.

- In particular, the following list of guiding principles characterize SOA platforms:

    - *Standardized service contract*

        - Services adhere to a given communication agreement, which is specified through one or more service description documents.

- *Loose coupling*
    - Services are designed as self-contained components, maintain relationships that minimize dependencies and only require being aware of each other.
    - Service contracts will enforce the required interaction among services.
    - This simplifies the flexible aggregation of services and enables a more agile design strategy that supports the evolution of the enterprise business.
- *Abstraction*
    - A service is completely defined by service contracts and description documents.
    - Abstraction hiding the logic, which is encapsulated within their implementation.
    - The use of service description documents and contracts removes the need to consider the technical implementation details.
    - It provides a more intuitive framework to define software systems within a business context.
- *Reusability*
    - Designed as components, services can be reused more efficiently, thus reducing development time and the associated costs.
    - Reusability allows for a more agile design and cost effective system implementation and deployment.
- *Autonomy*
    - Services have control over the logic they encapsulate and do not know about their implementation.
- *Lack of state*

        - By providing a stateless interaction pattern, services increase the chance of being reused and aggregated, particularly in a scenario in which a single service is used by multiple consumers that belong to different administrative and business domains.

    Discoverability

    -
        - Services are defined by description documents that constitute supplemental metadata through which they can be effectively discovered.
        - Service discovery provides an effective means for utilizing third party resources.

    - *Composability*

        - Using services as building blocks, difficult operations can be implemented.
        - Service orchestration and choreography provide a solid support for composing services and achieving desired business goals.

- Together with these principles, other resources guide the use of SOA for enterprise application integration (EAI).

- The SOA manifest integrates the previously described principles with general considerations about the overall goals of a service oriented approach to enterprise application software design and what is valued in SOA.

- Modeling frameworks and methodologies, such as the Service Oriented Modeling Framework (SOMF) and reference architectures introduced by the Organization for Advancement of Structured Information Standards (OASIS), provide means for effectively realizing service oriented architectures.

- SOA can be realized through several technologies.

- The first implementations of SOA have leveraged distributed object programming technologies such as CORBA and DCOM.

- CORBA has been a suitable platform for realizing SOA systems because it provides interoperability among different implementations and has been designed as a specification supporting the development of industrial applications.

- Nowadays, SOA is mostly realized through Web services technology, which provides an interoperable platform for connecting systems and applications.

2.2 Web Services

- Web services are the prominent technology for implementing SOA systems and applications.

- They leverage Internet technologies and standards for building distributed systems. Several aspects make Web services the technology of choice for SOA.

    - First, they allow for interoperability across different platforms and programming languages.
    - Second, they are based on well-known and vendor independent standards such as HTTP, SOAP, XML and WSDL.
    - Third, they provide an intuitive and simple way to connect heterogeneous software systems, enabling the quick composition of services in a distributed environment.
    - Finally, they provide the features required by enterprise business applications to be used in an industrial environment.

- They define facilities for enabling service discovery, which allows the system architect to more efficiently compose SOA applications and service metering to assess whether a specific service complies with the contract between the service provider and the service consumer.

- The concept behind a Web service is very simple.

- Using as a basis the object oriented abstraction, a Web service exposes a set of operations that can be invoked by leveraging Internet based protocols.

- The semantics for invoking Web service methods is expressed through interoperable standards such as XML and WSDL, which also provide a complete framework for expressing simple and complex types in a platform independent manner.

- Web services are made accessible by being hosted in a Web server

- HTTP is the most popular transport protocol used for interacting with Web services.

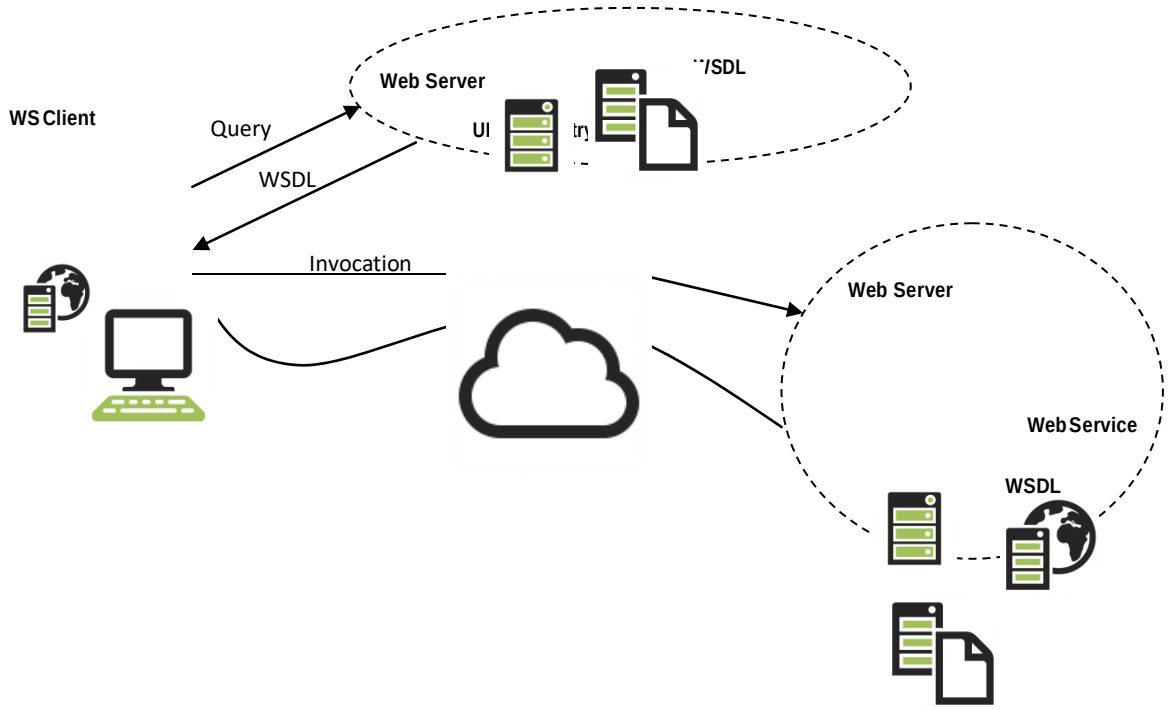


Figure 2.1 Reference scenario for Web Services

- Figure 2.1 describes the common use case scenarios for Web services.

- System architects develop a Web service with their technology of choice and deploy it in compatible Web or application servers.

- The service description document is expressed by means of Web Service Definition Language (WSDL), can be either uploaded to a global registry or attached as a metadata to the service itself.

- Service consumers can look up and discover services in global catalogs using Universal Description Discovery and Integration (UDDI).

- The Web service description document allows service consumers to automatically generate clients for the given service and embed them in their existing application.

- Web services are now extremely popular, so bindings exist for any mainstream programming language in the form of libraries or development support tools.

- This makes the use of Web services seamless and straightforward with respect to technologies such as CORBA that require much more integration effort.

- Moreover, being interoperable, Web services constitute a better solution for SOA with respect to several distributed object frameworks, such as .NET Remoting, Java RMI, and DCOM/COM1, which limit their applicability to a single platform or environment.

- Besides the main function of enabling remote method invocation by using Web based and interoperable standards, Web services encompass several technologies that put together and facilitate the integration of heterogeneous applications and enable service oriented computing.

- Figure 2.2 shows the Web service technologies stack that lists all the components of the conceptual framework describing and enabling the Web services abstraction.

- These technologies cover all the aspects that allow Web services to operate in a distributed environment, from the specific requirements for the networking to the discovery of services.

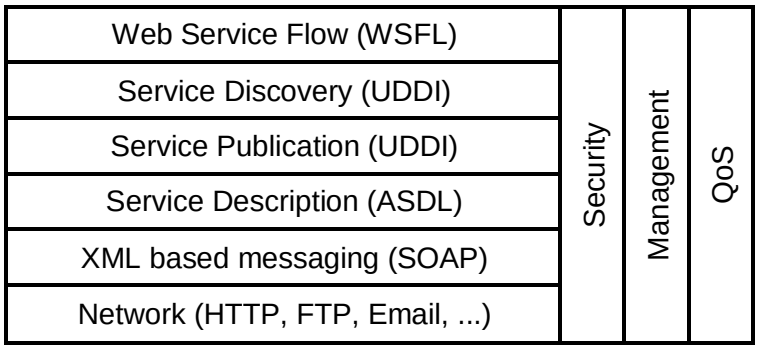


Figure 2.2 Web services technologies stack

- The backbone of all these technologies is XML, which is also one of the causes of Web service's popularity and ease of use.

- XML based languages are used to manage the low level interaction for Web service method calls (SOAP), for providing metadata about the services (WSDL), for discovery services (UDDI), and other core operations.

- In practice, the core components that enable Web services are SOAP and WSDL.

- Simple Object Access Protocol (SOAP) is an XML based language for exchanging structured information in a platform-independent manner, constitutes the protocol used for Web service method invocation.

- Within a distributed context leveraging the Internet, SOAP is considered an application layer protocol that leverages the transport level, most commonly HTTP, for IPC.

- SOAP structures the interaction in terms of messages that are XML documents mimicking the structure of a letter, with an envelope, a header, and a body.

- The envelope defines the boundaries of the SOAP message.

- The header is optional and contains relevant information on how to process the message.

```
Host : www.sample.com
Content-Type: application/soap+xml; charsetutf-8
Content-Length: <Size>

<?xml version= “1.0”>
<soap: Envelope xmlns:soap= “http://www.w3.org/2001/12/soap-envelope”
soap:encodingStyle= “http://www.w3.org/2001/12/soap-enoding” >
<soap:Header></soap:Header>
<soap:Body xmlns=http://www.sample.com/stock>
<m:GetPrice>
<m: StockName>DELL</m:StockName>
</m:GetPrice>
</soap:Body>
</soap: Envelope>

POST /StockPrice HTTP/1.1
Host : www.sample.com
Content-Type: application/soap+xml; charsetutf-8
Content-Length: <Size>

<?xml version= “1.0”>
<soap: Envelope xmlns:soap= “http://www.w3.org/2001/12/soap-envelope”
soap:encodingStyle= “http://www.w3.org/2001/12/soap-enoding” >
<soap:Header></soap:Header>
<soap:Body xmlns=http://www.sample.com/stock>
<m:GetPriceResponse>
<m: Price>58.5</m:Price>
</m:GetPriceResponse>
</soap:Body>
</soap: Envelope>
```

Figure 2.3 SOAP Message

- In addition to that it contains information such as routing and delivery settings, authentication, transaction contexts and authorization assertions.

- The body contains the actual message to be processed.

- The main uses of SOAP messages are method invocation and result retrieval.

- Figure 2.3 shows an example of a SOAP message used to invoke a Web service method that retrieves the price of a given stock and the corresponding reply.

- Despite the fact that XML documents are easy to produce and process in any platform or programming language, SOAP has often been considered quite inefficient because of the excessive use of markup that XML imposes for organizing the information into a well-formed document.

- Therefore, lightweight alternatives to the SOAP/XML pair have been proposed to support Web services.

2.3 REST and Systems of Systems

- The most relevant alternative to SOAP/XML pair is Representational State Transfer (REST), which provides a model for designing network based software systems utilizing the client / server model and leverages the facilities provided by HTTP for IPC without additional burden.

- In a RESTful system, a client sends a request over HTTP using the standard HTTP methods (PUT, GET, POST, and DELETE) and the server issues a response that includes the representation of the resource.

- By relying on this minimal support, it is possible to provide whatever it needed to replace the basic and most important functionality provided by SOAP, which is method invocation.

- The GET, PUT, POST, and DELETE methods constitute a minimal set of operations for retrieving, adding, modifying and deleting the data.

- Together with an appropriate URI organization to identify resources, all the atomic operations required by a Web service are implemented.

- The content of data is still transmitted using XML as part of the HTTP content, but the additional markup required by SOAP is removed.

- For this reason, REST represents a lightweight alternative to SOAP, which works effectively in contexts where additional aspects beyond those manageable through HTTP are absent.

- RESTful Web services operate in an environment where no additional security beyond the one supported by HTTP is required.

- This is not a great limitation, and RESTful Web services are quite popular and used to deliver functionalities at enterprise scale:

    - Twitter
    - Yahoo! (search APIs, maps, photos, etc)
    - Flickr
    - Amazon.com

- Web Service Description Language (WSDL) is an XML based language for the description of Web services.

- It is used to define the interface of a Web service in terms of methods to be called and types and structures of the required parameters and return values.

- In Figure 2.3 we notice that the SOAP messages for invoking the GetPrice method and receiving the result do not have any information about the type and structure of the parameters and the return values.

- This information is stored within the WSDL document attached to the Web service.

- Therefore, Web service consumer applications already know which types of parameters are required and how to interpret results.

- As an XML based language, WSDL allows for the automatic generation of Web service clients that can be easily embedded into existing applications.

- Moreover, XML is a platform and language independent specification, so clients for web services can be generated for any language that is capable of interpreting XML data.

- This is a fundamental feature that enables Web service interoperability and one of the reasons that make such technology a solution of choice for SOA.

- Besides those directly supporting Web services, other technologies that characterize Web 2.0 and contribute to enrich and empower Web applications and then SOA based systems.

- These fall under the names of Asynchronous JavaScript and XML (AJAX), JavaScript Standard Object Notation (JSON) and others.

- AJAX is a conceptual framework based on JavaScript and XML that enables asynchronous behavior in Web applications by leveraging the computing capabilities of modern Web browsers.

- This transforms simple Web pages in complete applications and used to enrich the user experience.

- AJAX uses XML to exchange data with Web services and applications

- An alternative to XML is JSON, which allows representing objects and collections of objects in a platform independent manner.

- Often it is preferred to transmit data in an AJAX context because compared to XML, it is a lighter notation and therefore allows transmitting the same amount of information in a more concise form.

2.4 Publish-Subscribe Model

- Publish-and-subscribe message model introduces a different message passing strategy, one that is based on notification among components.

- There are two major roles:

    - The publisher and the subscriber

        - The publisher provides facilities for the subscriber to register its interest in a specific topic or event.
        - Specific conditions holding true on the publisher side can trigger the creation of messages that are attached to a specific event.
        - A message will be available to all the subscribers that registered for the corresponding event.

- There are two major strategies for dispatching the event to the subscribers:

    - Push strategy

        - In this case it is the responsibility of the publisher to notify all the subscribers using method invocation.
    - Pull strategy

        - In this case the publisher simply makes available the message for a specific event and it is responsibility of the subscribers to check whether there are messages on the events that are registered.

- Publish and subscribe model is very suitable for implementing systems based on the one to many communication model and simplifies the implementation of indirect communication patterns.

- It is, in fact, not necessary for the publisher to know the identity of the subscribers to make the communication happen.

2.5 Basics of Virtualization

- Virtualization technology is one of the fundamental components of cloud computing, especially in regard to infrastructure based services.

- Virtualization allows the creation of a secure, customizable and isolated execution environment for running application.

- Virtualization is a large umbrella of technologies and concepts that are meant to provide an abstract environment whether virtual hardware or an operating system to run applications.

- The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure as a Service (IaaS) solutions for cloud computing.

- Virtualization technologies have gained renewed interested recently due to the confluence of several phenomena:

    - Increased performance and computing capacity.
    - Underutilized hardware and software resources

- Lack of space
- Greening initiatives
- Rise of administrative costs

- Virtualization is a broad concept that refers to the creation of a virtual version of something, whether hardware, a software environment, storage and a network.

- In a virtualized environment, there are three major components:

  - Guest
  - Host
  - Virtualization layer

- The guest represents the system component that interacts with the virtualization layer rather than with the host, as would normally happen.

- The host represents the original environment where the guest is supposed to be managed.

- The virtualization layer is responsible for recreating the same or a different environment where the guest will operate.

2.5.1 Characteristics of virtualized environments

- Increased security

  - The ability to control the execution of a guest in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment.
  - The virtual machine represents an emulated environment in which the guest is executed.
  - This level of indirection allows the virtual machine manager to control and filter the activity of the guest, thus preventing some harmful operations from being performed.

- Managed execution Virtualization of the execution environment not only allows increased security, but a wider range of features also can be implemented.

- In particular, sharing, aggregation, emulation, and isolation are the most relevant features

- Sharing

    - Virtualization allows the creation of a separate computing environment within the same host.
    - In this way it is possible to fully exploit the capabilities of a powerful guest, which would otherwise be underutilized.

- Aggregation

    - Not only is it possible to share physical resource among several guests but virtualization also allows aggregation, which is the opposite process.
    - A group of separate hosts can be tied together and represented to guests as a single virtual host.

- Emulation

    - Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program.
    - This allows for controlling and tuning the environment that is exposed to guests.

- Isolation

    - Virtualization allows providing guests whether they are operating systems, applications, or other entities with a completely separate environment, in which they are executed.
    - The guest program performs its activity by interacting with an abstraction layer, which provides access to the underlying resources.

    - Benefits of Isolation
        - First it allows multiple guests to run on the same host without interfering with each other.
        - Second, it provides a separation between the host and the guest.

- Another important capability enabled by virtualization is performance tuning.

- This feature is a reality at present, given the considerable advances in hardware and software supporting virtualization.

- It becomes easier to control the performance of the guest by finely tuning the properties of the resources exposed through the virtual environment.

- This capability provides a means to effectively implement a quality of service (QoS) infrastructure that more easily fulfills the service level agreement (SLA) established for the guest.

- Portability

    - The concept of portability applies in different ways according to the specific type of virtualization considered.
    - In the case of a hardware virtualization solution, the guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines

2.6 Types of Virtualization

- Virtualization is mainly used to emulate execution environments, storage and networks.

- Execution virtualization techniques into two major categories by considering the type of host they require.

- Process level techniques are implemented on top of an existing operating system, which has full control of the hardware.

- System level techniques are implemented directly on hardware and do not require or require a minimum of support from existing operating system.

- Within these two categories we can list various techniques that offer the guest a different type of virtual computation environment:

    - Bare hardware
    - Operating system resources
    - Low level programming language
    - Application libraries

- Execution virtualization includes all techniques that aim to emulate an execution environment that is separate from the one hosting the virtualization layer.

- All these techniques concentrate their interest on providing support for the execution of programs, whether these are the operating system, a binary specification of a program compiled against an abstract machine model or an application.

- Therefore, execution virtualization can be implemented directly on top of the hardware by the operating system, an application and libraries (dynamically or statically) linked to an application image.

- Modern computing systems can be expressed in terms of the reference model described in Figure 2.4.

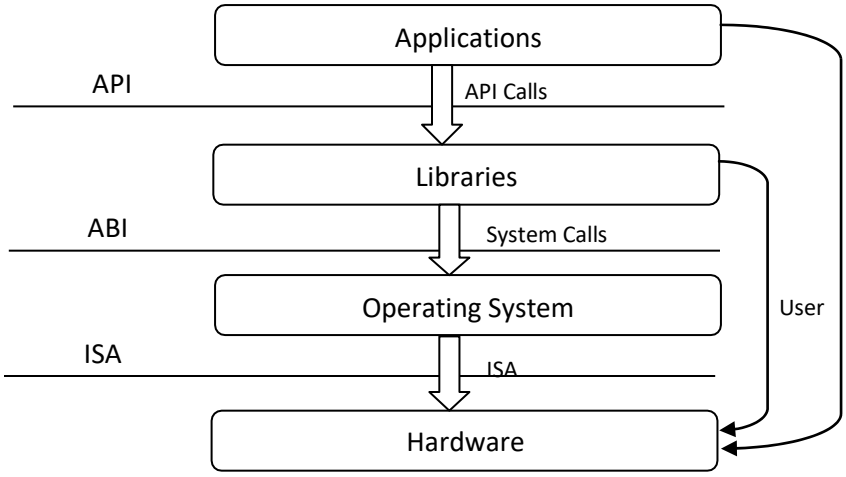


Figure 2.4 Machine reference model

- At the bottom layer, the model for the hardware is expressed in terms of the Instruction Set Architecture (ISA), which defines the instruction set for the processor, registers, memory and an interrupt management.

- ISA is the interface between hardware and software.

- ISA is important to the operating system (OS) developer (System ISA) and developers of applications that directly manage the underlying hardware (User ISA).

- The application binary interface (ABI) separates the operating system layer from the applications and libraries, which are managed by the OS.

- ABI covers details such as low level data types, alignment, call conventions and defines a format for executable programs.

- System calls are defined at this level.

- This interface allows portability of applications and libraries across operating systems that implement the same ABI.

- The highest level of abstraction is represented by the application programming interface (API), which interfaces applications to libraries and the underlying operating system.

- For this purpose, the instruction set exposed by the hardware has been divided into different security classes that define who can operate with them.

- The first distinction can be made between privileged and non privileged instructions.

    - Non privileged instructions are those instructions that can be used without interfering with other tasks because they do not access shared resources.
    - This category contains all the floating, fixed-point, and arithmetic instructions.

- Privileged instructions are those that are executed under specific restrictions and are mostly used for sensitive operations, which expose (behavior-sensitive) or modify (control-sensitive) the privileged state.

- Some types of architecture feature more than one class of privileged instructions and implement a finer control of how these instructions can be accessed.

- For instance, a possible implementation features a hierarchy of privileges illustrate in the figure 2.5 in the form of ring-based security: Ring 0, Ring 1, Ring 2, and Ring 3;

    - Ring 0 is in the most privileged level and Ring 3 in the least privileged level.
    - Ring 0 is used by the kernel of the OS, rings 1 and 2 are used by the OS level services, and Ring 3 is used by the user.
    - Recent systems support only two levels, with Ring 0 for supervisor mode and Ring 3 for user mode.

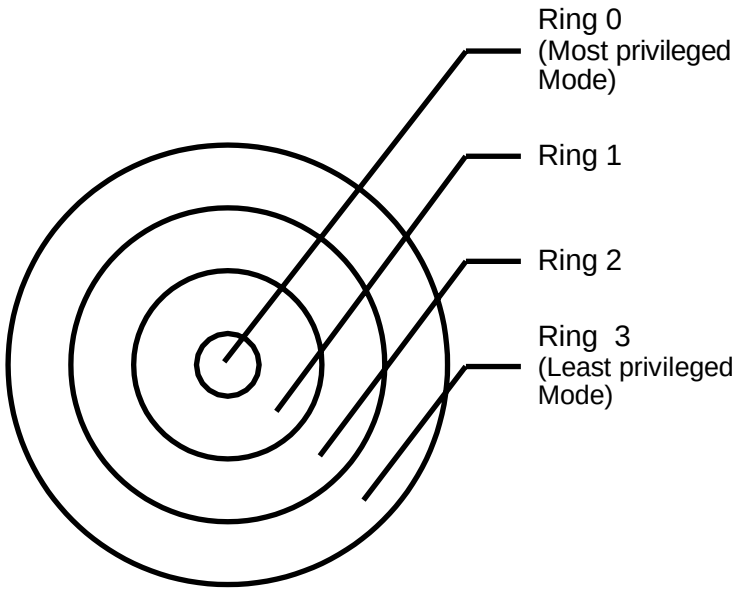


Figure 2.5 Security rings

- All the current systems support at least two different execution modes: supervisor mode and user mode.
    - The supervisor mode denotes an execution mode in which all the instructions (privileged and non privileged) can be executed without any restriction.
    - This mode, also called master mode or kernel mode, is generally used by the operating system (or the hypervisor) to perform sensitive operations on hardware level resources.
    - In user mode, there are restrictions to control the machine level resources.

- The distinction between user and supervisor mode allows us to understand the role of the hypervisor and why it is called that.

- Conceptually, the hypervisor runs above the supervisor mode and from here the prefix “hyper” is used.

- In reality, hypervisors are run in supervisor mode and the division between privileged and non privileged instructions has posed challenges in designing virtual machine managers.

2.6.1 Hardware level virtualization

- Hardware level virtualization is a virtualization technique that provides an abstract execution environment in terms of computer hardware on top of which a guest operating system can be run.

- In this model, the guest is represented by the operating system, the host by the physical computer hardware, the virtual machine by its emulation and the virtual machine manager by the hypervisor.

- The hypervisor is generally a program or a combination of software and hardware that allows the abstraction of the underlying physical hardware.

- Hardware level virtualization is also called system virtualization, since it provides ISA to virtual machines, which is the representation of the hardware interface of a system.

- This is to differentiate it from process virtual machines, which expose ABI to virtual machines.

- Hypervisors is a fundamental element of hardware virtualization is the hypervisor, or virtual machine manager (VMM).

- It recreates a hardware environment in which guest operating systems are installed.

- There are two major types of hypervisor: Type I and Type II. Figure 2.6 shows different type of hypervisors.

    - Type I hypervisors run directly on top of the hardware.
    - Type I hypervisor take the place of the operating systems and interact directly with the ISA interface exposed by the underlying hardware and they emulate this interface in order to allow the management of guest operating systems.

- This type of hypervisor is also called a native virtual machine since it runs natively on hardware.

- Type II hypervisors require the support of an operating system to provide virtualization services.
- This means that they are programs managed by the operating system, which interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems.
- This type of hypervisor is also called a hosted virtual machine since it is hosted within an operating system.

2.6.2 Hardware virtualization techniques

- Hardware virtualization provides an abstract execution environment by Hardware assisted virtualization, Full virtualization, Paravirtualization and Partial virtualization techniques.

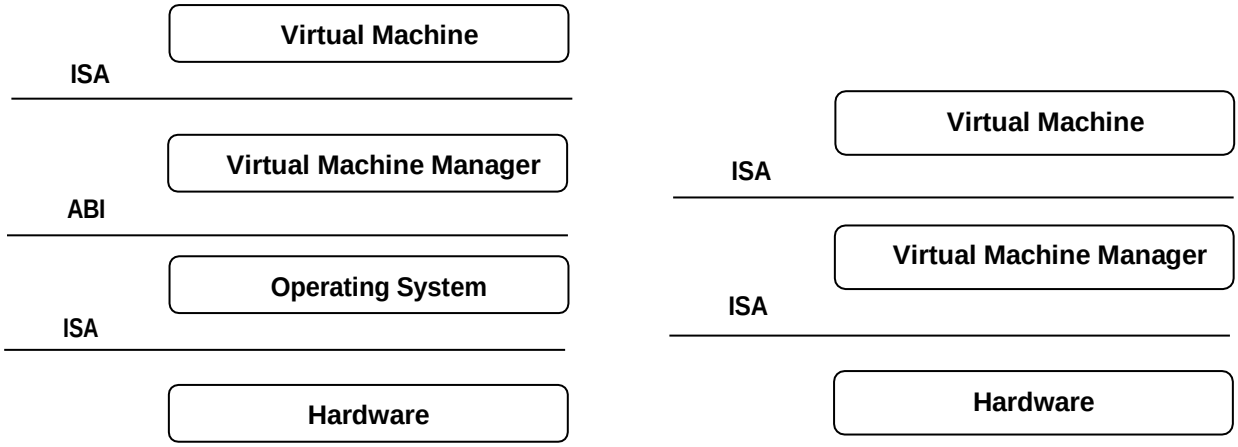


Figure 2.6 Hosted virtual machine and native virtual machine

2.6.2.1 Hardware assisted virtualization

- Hardware assisted virtualization refers to a scenario in which the hardware provides architectural support for building a virtual machine manager able to run a guest operating system in complete isolation.

- This technique was originally introduced in the IBM System/370.

- At present, examples of hardware assisted virtualization are the extensions to the x86 architecture introduced with Intel-VT (formerly known as Vanderpool) and AMD-V (formerly known as Pacifica).

- These extensions, which differ between the two vendors, are meant to reduce the performance penalties experienced by emulating x86 hardware with hypervisors.

- Before the introduction of hardware assisted virtualization, software emulation of x86 hardware was significantly costly from the performance point of view.

- The reason for this is that by design the x86 architecture did not meet the formal requirements introduced by Popek and Goldberg and early products were using binary translation to trap some sensitive instructions and provide an emulated version.

- Products such as VMware Virtual Platform, introduced in 1999 by VMware, which pioneered the field of x86 virtualization, were based on this technique.

- After 2006, Intel and AMD introduced processor extensions and a wide range of virtualization solutions took advantage of them: Kernel-based Virtual Machine (KVM), VirtualBox, Xen, VMware, Hyper-V, Sun xVM, Parallels, and others.

2.6.2.2 Full virtualization

- Full virtualization refers to the ability to run a program, most likely an operating system, directly on top of a virtual machine and without any modification, as though it were run on the raw hardware.

- To make this possible, virtual machine managers are required to provide a complete emulation of the entire underlying hardware.

- The principal advantage of full virtualization is complete isolation, which leads to enhanced security, ease of emulation of different architectures and coexistence of different systems on the same platform.

- Whereas it is a desired goal for many virtualization solutions, full virtualization poses important concerns related to performance and technical implementation.

- A key challenge is the interception of privileged instructions such as I/O instructions: Since they change the state of the resources exposed by the host, they have to be contained within the virtual machine manager.

- A simple solution to achieve full virtualization is to provide a virtual environment for all the instructions, thus posing some limits on performance.

- A successful and efficient implementation of full virtualization is obtained with a combination of hardware and software, not allowing potentially harmful instructions to be executed directly on the host.

2.6.2.3 Paravirtualization

- Paravirtualization is a not transparent virtualization solution that allows implementing thin virtual machine managers.

- Paravirtualization techniques expose a software interface to the virtual machine that is slightly modified from the host and, as a consequence, guests need to be modified.

- The aim of paravirtualization is to provide the capability to demand the execution of performance critical operations directly on the host, thus preventing performance losses that would otherwise be experienced in managed execution.

- This allows a simpler implementation of virtual machine managers that have to simply transfer the execution of these operations, which were hard to virtualize, directly to the host.

- To take advantage of such an opportunity, guest operating systems need to be modified and explicitly ported by remapping the performance critical operations through the virtual machine software interface.

- This is possible when the source code of the operating system is available, and this is the reason that paravirtualization was mostly explored in the opensource and academic environment.

- This technique has been successfully used by Xen for providing virtualization solutions for Linux-based operating systems specifically ported to run on Xen hypervisors.

- Operating systems that cannot be ported can still take advantage of para virtualization by using ad hoc device drivers that remap the execution of critical instructions to the paravirtualization APIs exposed by the hypervisor.

- Xen provides this solution for running Windows based operating systems on x86 architectures.

- Other solutions using paravirtualization include VMWare, Parallels, and some solutions for embedded and real-time environments such as TRANGO, Wind River, and XtratuM.

2.6.2.4 Partial virtualization

- Partial virtualization provides a partial emulation of the underlying hardware, thus not allowing the complete execution of the guest operating system in complete isolation.

- Partial virtualization allows many applications to run transparently, but not all the features of the operating system can be supported as happens with full virtualization.

- An example of partial virtualization is address space virtualization used in time sharing systems; this allows multiple applications and users to run concurrently in a separate memory space, but they still share the same hardware resources (disk, processor, and network).

- Historically, partial virtualization has been an important milestone for achieving full virtualization, and it was implemented on the experimental IBM M44/44X.

- Address space virtualization is a common feature of contemporary operating systems.

2.6.3 Operating system level virtualization

- Operating system level virtualization offers the opportunity to create different and separated execution environments for applications that are managed concurrently.

- Differently from hardware virtualization, there is no virtual machine manager or hypervisor and the virtualization is done within a single operating system where the OS kernel allows for multiple isolated user space instances.

- The kernel is also responsible for sharing the system resources among instances and for limiting the impact of instances on each other.

- A user space instance in general contains a proper view of the file system which is completely isolated and separate IP addresses, software configurations and access to devices.

- Operating systems supporting this type of virtualization are general purpose, timeshared operating systems with the capability to provide stronger namespace and resource isolation.

- This virtualization technique can be considered an evolution of the chroot mechanism in Unix systems.

- The chroot operation changes the file system root directory for a process and its children to a specific directory.

- As a result, the process and its children cannot have access to other portions of the file system than those accessible under the new root directory.

- Because Unix systems also expose devices as parts of the file system, by using this method it is possible to completely isolate a set of processes.

- Following the same principle, operating system level virtualization aims to provide separated and multiple execution containers for running applications.

- This technique is an efficient solution for server consolidation scenarios in which multiple application servers share the same technology: operating system, application server framework, and other components.

- Examples of operating system-level virtualizations are FreeBSD Jails, IBM Logical Partition (LPAR), SolarisZones and Containers, Parallels Virtuozzo Containers, OpenVZ, iCore Virtual Accounts, Free Virtual Private Server (FreeVPS), and others.

2.6.4 Programming language-level virtualization

- Programming language level virtualization is mostly used to achieve ease of deployment of applications, managed execution, portability across different platforms and operating systems.

- It consists of a virtual machine executing the byte code of a program which is the result of the compilation process.

- Compilers implemented and used this technology to produce a binary format representing the machine code for an abstract architecture.

- The characteristics of this architecture vary from implementation to implementation.

- Generally these virtual machines constitute a simplification of the underlying hardware instruction set and provide some high level instructions that map some of the features of the languages compiled for them.

- At runtime, the byte code can be either interpreted or compiled on the fly against the underlying hardware instruction set.

- Programming language level virtualization has a long trail in computer science history and originally was used in 1966 for the implementation of Basic Combined Programming Language (BCPL), a language for writing compilers and one of the ancestors of the C programming language.

- Other important examples of the use of this technology have been the UCSD Pascal and Smalltalk.

- Virtual machine programming languages become popular again with Sun’s introduction of the Java platform in 1996.

- The Java virtual machine was originally designed for the execution of programs written in the Java language, but other languages such as Python, Pascal, Groovy and Ruby were made available.

- The ability to support multiple programming languages has been one of the key elements of the Common Language Infrastructure (CLI) which is the specification behind .NET Framework.

2.6.5 Application level virtualization

- Application level virtualization is a technique allowing applications to be run in runtime environments that do not natively support all the features required by such applications.

- In this scenario, applications are not installed in the expected runtime environment but are run as though they were.

- In general, these techniques are mostly concerned with partial file systems, libraries, and operating system component emulation.

- Such emulation is performed by a thin layer called a program or an operating system component that is in charge of executing the application.

- Emulation can also be used to execute program binaries compiled for different hardware architectures.

- In this case, one of the following strategies can be implemented:

- Interpretation: In this technique every source instruction is interpreted by an emulator for executing native ISA instructions, leading to poor performance. Interpretation has a minimal startup cost but a huge overhead, since each instruction is emulated.

- Binary translation: In this technique every source instruction is converted to native instructions with equivalent functions. After a block of instructions is translated, it is cached and reused.

- Application virtualization is a good solution in the case of missing libraries in the host operating system.

- In this case a replacement library can be linked with the application or library calls can be remapped to existing functions available in the host system.

- Another advantage is that in this case the virtual machine manager is much lighter since it provides a partial emulation of the runtime environment compared to hardware virtualization.

- Compared to programming level virtualization, which works across all the applications developed for that virtual machine, application level virtualization works for a specific environment.

- It supports all the applications that run on top of a specific environment.

- One of the most popular solutions implementing application virtualization is Wine, which is a software application allowing Unix-like operating systems to execute programs written for the Microsoft Windows platform.

- Wine features a software application acting as a container for the guest application and a set of libraries, called Winelib, that developers can use to compile applications to be ported on Unix systems.

- Wine takes its inspiration from a similar product from Sun, Windows Application Binary Interface (WABI) which implements the Win 16 API specifications on Solaris.

- A similar solution for the Mac OS X environment is CrossOver, which allows running Windows applications directly on the Mac OS X operating system.

- VMware ThinApp is another product in this area, allows capturing the setup of an installed application and packaging it into an executable image isolated from the hosting operating system.

2.6.6 Other types of virtualization

- Other than execution virtualization, other types of virtualization provide an abstract environment to interact with.

- These mainly cover storage, networking, and client/server interaction.

2.6.6.1 Storage virtualization

- Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation.

- Using this technique, users do not have to be worried about the specific location of their data, which can be identified using a logical path.

- Storage virtualization allows us to harness a wide range of storage facilities and represent them under a single logical file system.

- There are different techniques for storage virtualization, one of the most popular being network based virtualization by means of storage area networks (SANs).

- SANs use a network accessible device through a large bandwidth connection to provide storage facilities.

2.6.6.2 Network virtualization

- Network virtualization combines hardware appliances and specific software for the creation and management of a virtual network.

- Network virtualization can aggregate different physical networks into a single logical network (external network virtualization) or provide network like functionality to an operating system partition (internal network virtualization).

- The result of external network virtualization is generally a virtual LAN (VLAN).

- A VLAN is an aggregation of hosts that communicate with each other as though they were located under the same broadcasting domain.

- Internal network virtualization is generally applied together with hardware and operating system-level virtualization, in which the guests obtain a virtual network interface to communicate with.

- There are several options for implementing internal network virtualization:

    - The guest can share the same network interface of the host and use Network Address Translation (NAT) to access the network;
    - The virtual machine manager can emulate, and install on the host, an additional network device, together with the driver.
    - The guest can have a private network only with the guest.

2.6.6.3 Desktop virtualization

- Desktop virtualization abstracts the desktop environment available on a personal computer in order to provide access to it using a client/server approach.

- Desktop virtualization provides the same outcome of hardware virtualization but serves a different purpose.

- Similarly to hardware virtualization, desktop virtualization makes accessible a different system as though it were natively installed on the host but this system is remotely stored on a different host and accessed through a network connection.

- Moreover, desktop virtualization addresses the problem of making the same desktop environment accessible from everywhere.

- Although the term desktop virtualization strictly refers to the ability to remotely access a desktop environment, generally the desktop environment is stored in a remote server or a data center that provides a high availability infrastructure and ensures the accessibility and persistence of the data.

- In this scenario, an infrastructure supporting hardware virtualization is fundamental to provide access to multiple desktop environments hosted on the same server.

- A specific desktop environment is stored in a virtual machine image that is loaded and started on demand when a client connects to the desktop environment.

- This is a typical cloud computing scenario in which the user leverages the virtual infrastructure for performing the daily tasks on his computer.

- The advantages of desktop virtualization are high availability, persistence, accessibility, and ease of management.

- The basic services for remotely accessing a desktop environment are implemented in software components such as Windows Remote Services, VNC, and X Server.

- Infrastructures for desktop virtualization based on cloud computing solutions include Sun Virtual Desktop Infrastructure (VDI), Parallels Virtual Desktop Infrastructure (VDI), Citrix XenDesktop, and others.

2.6.6.4 Application server virtualization

- Application server virtualization abstracts a collection of application servers that provide the same services as a single virtual application server by using load balancing strategies and providing a high availability infrastructure for the services hosted in the application server.

- This is a particular form of virtualization and serves the same purpose of storage virtualization by providing a better quality of service rather than emulating a different environment.

2.7 Implementation Levels of Virtualization

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.

- The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.

- Hardware resources (CPU, memory, I/O devices) or software resources (operating system and software libraries) can be virtualized in various functional layers.

- 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.

- With sufficient storage, any computer platform can be installed in another host computer, even if they use processors with different instruction sets and run with distinct operating systems on the same hardware.

2.7.1 Levels of virtualization implementation

- A traditional computer runs with a host operating system specially tailored for its hardware architecture, as shown in Figure 2.7(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 2.7(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, as we will discuss shortly.

- 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.

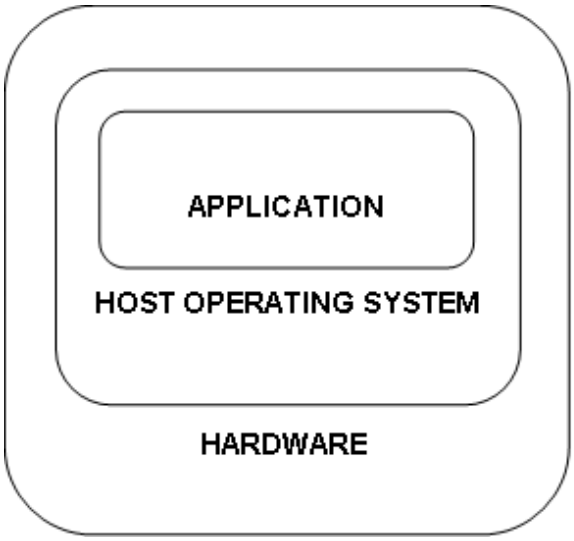


Figure 2.7 (a) Traditional Computer

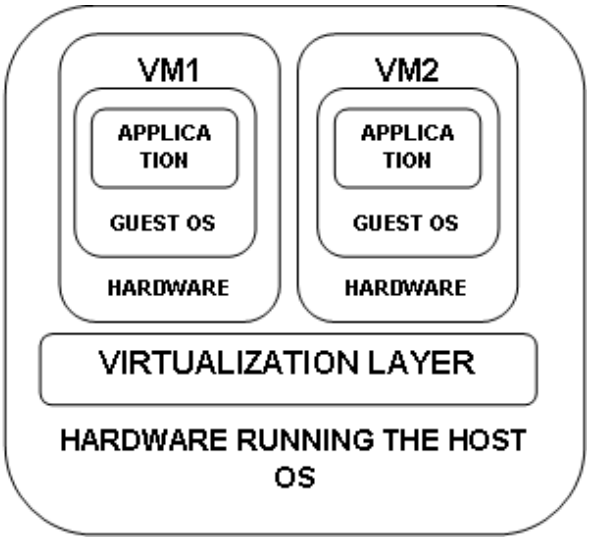


Figure (b) After Virtualization

2.7.2 Instruction set architecture level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.

- 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. 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.

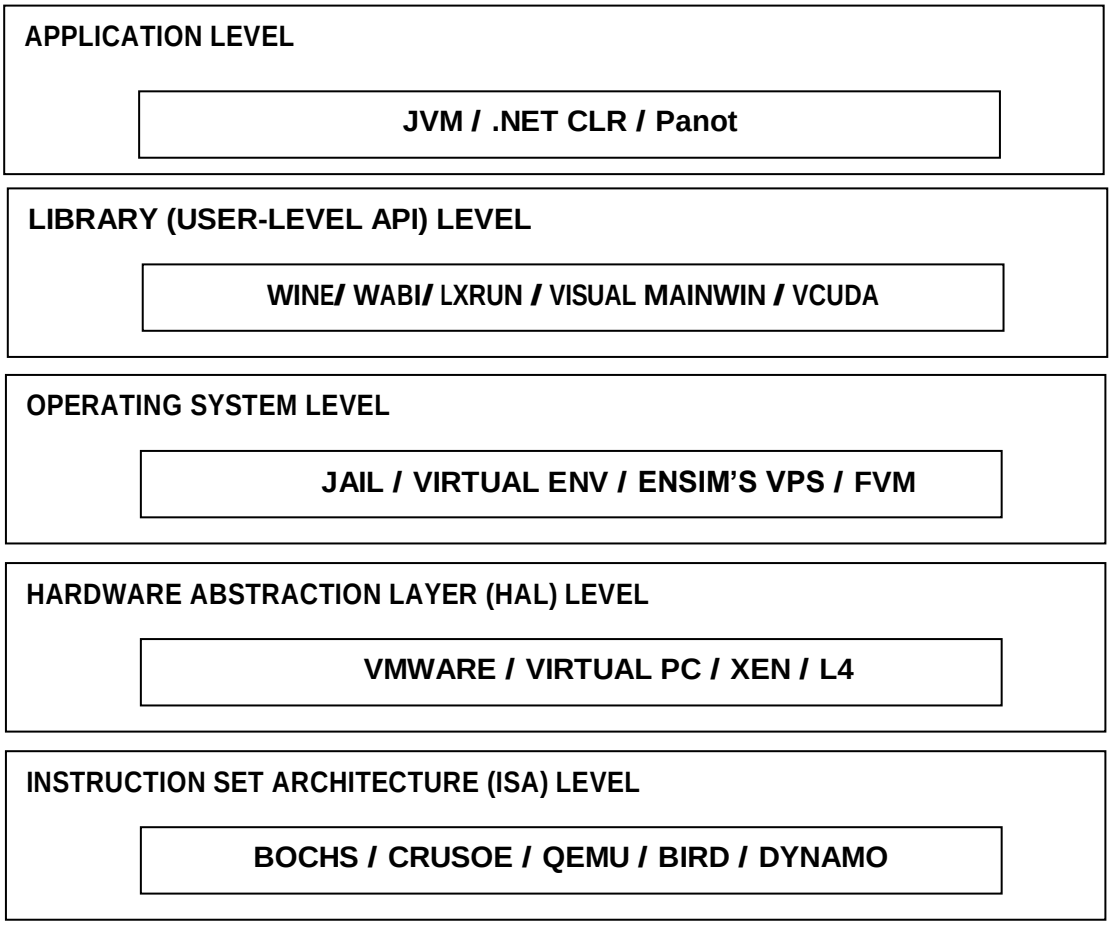


Figure 2.8 Virtualization ranging from hardware to applications in five abstraction levels

2.7.3 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 bymultiple users concurrently.

- The idea was implemented in the IBM VM/370 in the 1960s. Xen hypervisor has been applied to virtualize x86-based machines to run Linux or other guest OS applications.

2.7.4 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.

- 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.

- Compared to hardware-level virtualization, the benefits of OS extensions are twofold: VMs at the operating system level have minimal startup/shutdown costs, low resource requirements, and high scalability

- 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:

- All OS-level VMs on the same physical machine share a single operating system kernel
    - 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.
- Virtualization Support for the Linux Platform
    - OpenVZ is an OS-level tool designed to support Linux platforms to create virtual environments for running VMs under different guest Operating Systems.
    - OpenVZ is an open source container-based virtualization solution built on Linux. To support virtualization and isolation of various subsystems, limited resource management, and check pointing, OpenVZ modifies the Linux kernel.

2.7.5 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. Another example is the vCUDA which allows applications executing within VMs to leverage GPU hardware acceleration.
- Library level virtualization is also known as user-level Application Binary Interface (ABI) or API emulation.
- This type of virtualization can create execution environments for running alien programs on a platform rather than creating a VM to run the entire operating system.

- API call interception and remapping are the key functions performed. The WABI offers middleware to convert Windows system calls to Solaris system calls.

- Lxrun is really a system call emulator that enables Linux applications written for x86 hosts to run on UNIX systems.

- Similarly, Wine offers library support for virtualizing x86 processors to run Windows applications on UNIX hosts.

- Visual MainWin offers a compiler support system to develop Windows applications using Visual Studio to run on some UNIX hosts.

- The vCUDA for Virtualization of General-Purpose GPUs. CUDA is a programming model and library for general-purpose GPUs. It leverages the high performance of GPUs to run compute-intensive applications on host operating systems. However, it is difficult to run CUDA applications on hardware-level VMs directly.

- vCUDA virtualizes the CUDA library and can be installed on guest OSes. When CUDA applications run on a guest OS and issue a call to the CUDA API, vCUDA intercepts the call and redirects it to the CUDA API running on the host OS.

2.7.6 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.

- An example is the LANDesk application virtualization platform which deploys software applications as self contained, executable files in an isolated environment without requiring installation, system modifications or elevated security privileges.

2.7.7 Relative merits of different approaches

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
Instruction Set Architecture	Very Low	Very High	Moderate	Moderate
Hardware-level virtualization	Very High	Moderate	Very High	High
OS-level virtualization	Very High	Low	Moderate	Low
Library support level	Moderate	Low	Low	Low
User application level	Low	Low	Very High	Very High

Table 2.1 Relative Merits of Virtualization at Various Levels

2.8 Virtualization Structures, Tools and Mechanisms

- In general, there are three typical classes of VM architecture.
- 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.

2.8.1 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 micro kernel architecture like the Microsoft Hyper-V.
- It can assume 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.

2.8.2 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. Figure 2.9 shows architecture of Xen hypervisor.

- 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.

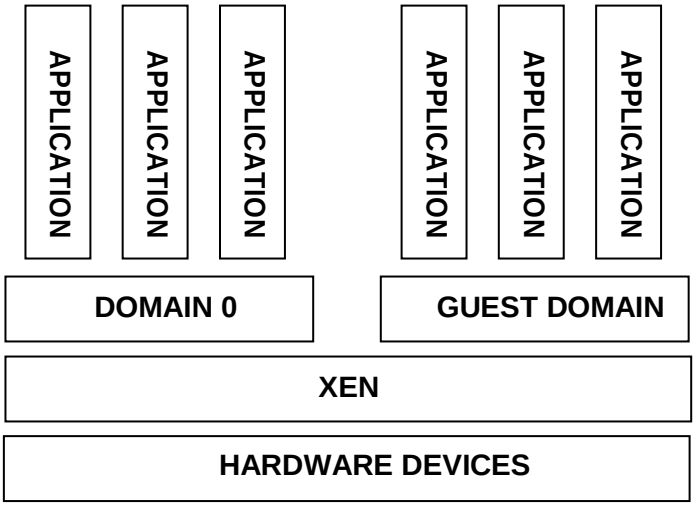


Figure 2.9 Xen domain 0 for control and I/O & guest domain for user applications.

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

- For example, Xen is based on Linux and its security level is C2. Its management VM is named Domain 0 which has the privilege to manage other VMs implemented on the same host.

- If Domain 0 is compromised, the hacker can control the entire system. So, in the VM system, security policies are needed to improve the security of Domain 0.

- Domain 0, behaving as a VMM, allows users to create, copy, save, read, modify, share, migrate and roll back VMs as easily as manipulating a file, which flexibly provides tremendous benefits for users.

2.8.3 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, non virtualizable 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 software layer is built between the host OS and guest OS.

- 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.

- 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.

- Full virtualization combines binary translation and direct execution.

- 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.

- Host based architecture has some distinct advantages, as enumerated next.

    - First, the user can install this VM architecture without modifying the host OS.
    - Second, the host-based approach appeals to many host machine configurations.

2.8.4 Paravirtualization with compiler support

- When x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS.

- According to the x86 ring definitions, the virtualization layer should also be installed at Ring 0. Different instructions at Ring 0 may cause some problems.

- Although paravirtualization 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 paravirtualized OSes is high, because they may require deep OS kernel modifications.
    - Finally, the performance advantage of paravirtualization varies greatly due to workload variations.

- Compared with full virtualization, paravirtualization is relatively easy and more practical. The main problem in full virtualization is its low performance in binary translation.

- KVM is a Linux paravirtualization system. It is a part of the Linux version 2.6.20 kernel.

- In KVM, Memory management and scheduling activities are carried out by the existing Linux kernel.

- The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine.

- KVM is a hardware assisted and paravirtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants.

- Unlike the full virtualization architecture which intercepts and emulates privileged and sensitive instructions at runtime, paravirtualization handles these instructions at compile time.

- The guest OS kernel is modified to replace the privileged and sensitive instructions with hypercalls to the hypervisor or VMM. Xen assumes such paravirtualization architecture.

- The guest OS running in a guest domain may run at Ring 1 instead of at Ring 0. This implies that the guest OS may not be able to execute some privileged and sensitive instructions. The privileged instructions are implemented by hypercalls to the hypervisor.

2.9 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.

- For the x86 architecture, Intel and AMD have proprietary technologies for hardware assisted virtualization.

2.9.1 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.

- 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.

- At the time of this writing, many hardware virtualization products were available.

- The VMware Workstation is a VM software suite for x86 and x86-64 computers.

- This software suite allows users to set up multiple x86 and x86-64 virtual computers and to use one or more of these VMs simultaneously with the host operating system.

- The VMware Workstation assumes the host-based virtualization.

- Xen is a hypervisor for use in IA-32, x86-64, Itanium and PowerPC 970 hosts.

- One or more guest OS can run on top of the hypervisor.

- KVM is a Linux kernel virtualization infrastructure.

- KVM can support hardware assisted virtualization and paravirtualization by using the Intel VT-x or AMD-v and VirtIO framework, respectively.

- The VirtIO framework includes a paravirtual Ethernet card, a disk I/O controller and a balloon device for adjusting guest memory usage and a VGA graphics interface using VMware drivers.

2.9.2 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.

- The unprivileged instructions of VMs run directly on the host machine for higher efficiency.

- The critical instructions are divided into three categories: privileged instructions, control sensitive 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.

- 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.

- The privileged instructions including control and behavior sensitive instructions of a VM are executed; they are trapped in the VMM.

- RISC CPU architectures can be naturally virtualized because all control and behavior sensitive instructions are privileged instructions.

- The x86 CPU architectures are not primarily designed to support virtualization.

2.9.2.1 Hardware-assisted CPU virtualization

- This technique attempts to simplify virtualization because full or paravirtualization is complicated.

- Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors.

- Therefore, operating systems can still run at Ring 0 and hypervisor can run at Ring 1.

- All the privileged and sensitive instructions are trapped in the hypervisor automatically.

- This technique removes the difficulty of implementing binary translation of full virtualization.

- It also lets the operating system run in VMs without modification.

2.9.3 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.

- 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.

- 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.

- 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.

- The AMD Barcelona processor has featured hardware assisted memory virtualization since 2007.

- It provides hardware assistance to the two stage address translation in a virtual execution environment by using a technology called nested paging.

2.9.4 I/O virtualization

- I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.

- There are three ways to implement I/O virtualization: full device emulation, paravirtualization, and direct I/O.

- Full device emulation is the first approach for I/O virtualization. Generally, this approach emulates well known and 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.

- A single hardware device can be shared by multiple VMs that run concurrently. However, software emulation runs much slower than the hardware it emulates.

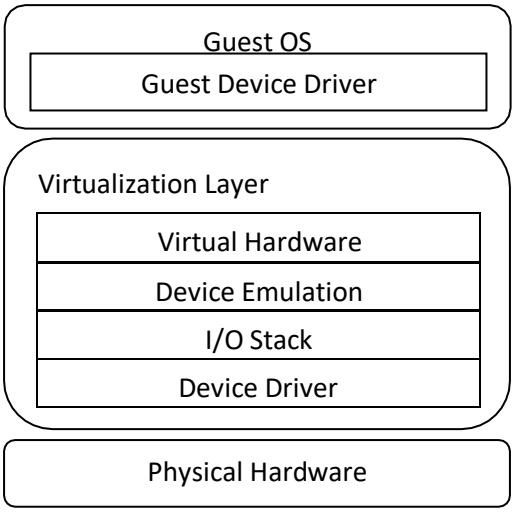


Figure 2.10 Device emulation for I/O Virtualization

- The paravirtualization 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.

- 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.

- For example, when a physical device is reclaimed (required by workload migration) for later reassignment, it may have been set to an arbitrary state (e.g., DMA to some arbitrary memory locations) that can function incorrectly or even crash the whole system.

- Since software based I/O virtualization requires a very high overhead of device emulation, hardware-assisted I/O virtualization is critical.

- Intel VT-d supports the remapping of I/O DMA transfers and device generated interrupts. The architecture of VT-d provides the flexibility to support multiple usage models that may run unmodified, special-purpose, or “virtualization-aware” guest OSes.

- Another way to help I/O virtualization is via self virtualized I/O (SV-IO).

- The key idea of SV-IO is to harness the rich resources of a multicore processor. All tasks associated with virtualizing an I/O device are encapsulated in SV-IO.

- It provides virtual devices and an associated access API to VMs and a management API to the VMM.

- SV-IO defines one virtual interface (VIF) for every kind of virtualized I/O device, such as virtual network interfaces, virtual block devices (disk), virtual camera devices,

2.9.4.1 Virtualization in multi-core processors

- Virtualizing a multi-core processor is relatively more complicated than virtualizing a uni-core processor.

- Muti-core virtualization has raised some new challenges to computer architects, compiler constructors, system designers, and application programmers.

- There are mainly two difficulties: Application programs must be parallelized to use all cores fully, and software must explicitly assign tasks to the cores, which is a very complex problem.

    - The first challenge, new programming models, languages, and libraries are needed to make parallel programming easier.
    - The second challenge has spawned research involving scheduling algorithms and resource management policies.

- Dynamic heterogeneity is emerging to mix the fat CPU core and thin GPU cores on the same chip, which further complicates the multi core or many core resource management.

- The dynamic heterogeneity of hardware infrastructure mainly comes from less reliable transistors and increased complexity in using the transistors.

2.9.4.2 Physical versus virtual processor cores

- A multicore virtualization method to allow hardware designers to get an abstraction of the low-level details of the processor cores.

- This technique alleviates the burden and inefficiency of managing hardware resources by software.

- It is located under the ISA and remains unmodified by the operating system or VMM (hypervisor).

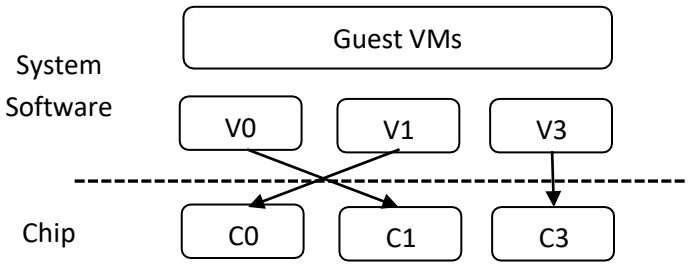


Figure 2.11 Multicore Virtualization method

- Figure 2.11 illustrates the technique of software visible VCPU moving from one core to another and temporarily suspending execution of a VCPU when there are no appropriate cores on which it can run.

2.9.4.3 Virtual hierarchy

- The emerging many core chip multiprocessors (CMPs) provide a new computing landscape.

- Instead of supporting time sharing jobs on one or a few cores, we can use the abundant cores in a space sharing, where single threaded or multithreaded jobs are simultaneously assigned to separate groups of cores for long time intervals.

- To optimize for space shared workloads, they propose using virtual hierarchies to overlay a coherence and caching hierarchy onto a physical processor.

- A virtual hierarchy is a cache hierarchy that can adapt to fit the workload or mix of workloads.

- The hierarchy's first level locates data blocks close to the cores needing them for faster access, establishes a shared-cache domain and establishes a point of coherence for faster communication.

- When a miss leaves a tile, it first attempts to locate the block (or sharers) within the first level. The first level can also provide isolation between independent workloads. A miss at the L1 cache can invoke the L2 access.

- Space sharing is applied to assign three workloads to three clusters of virtual cores:

    - Namely VM0 and VM3 for database workload, VM1 and VM2 for web server workload and VM4–VM7 for middleware workload.

- Each VM operates in a isolated fashion at the first level. This will minimize both miss access time and performance interference with other workloads or VMs.

- The shared resources of cache capacity, inter-connect links, and miss handling are mostly isolated between VMs. The second level maintains a globally shared memory.

- This facilitates dynamically repartitioning resources without costly cache flushes. A virtual hierarchy adapts to space-shared workloads like multiprogramming and server consolidation.

2.10 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.

- 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.

- In addition, application developers do not care about some infrastructure issues such as scalability and fault tolerance. Application developers focus on service logic.

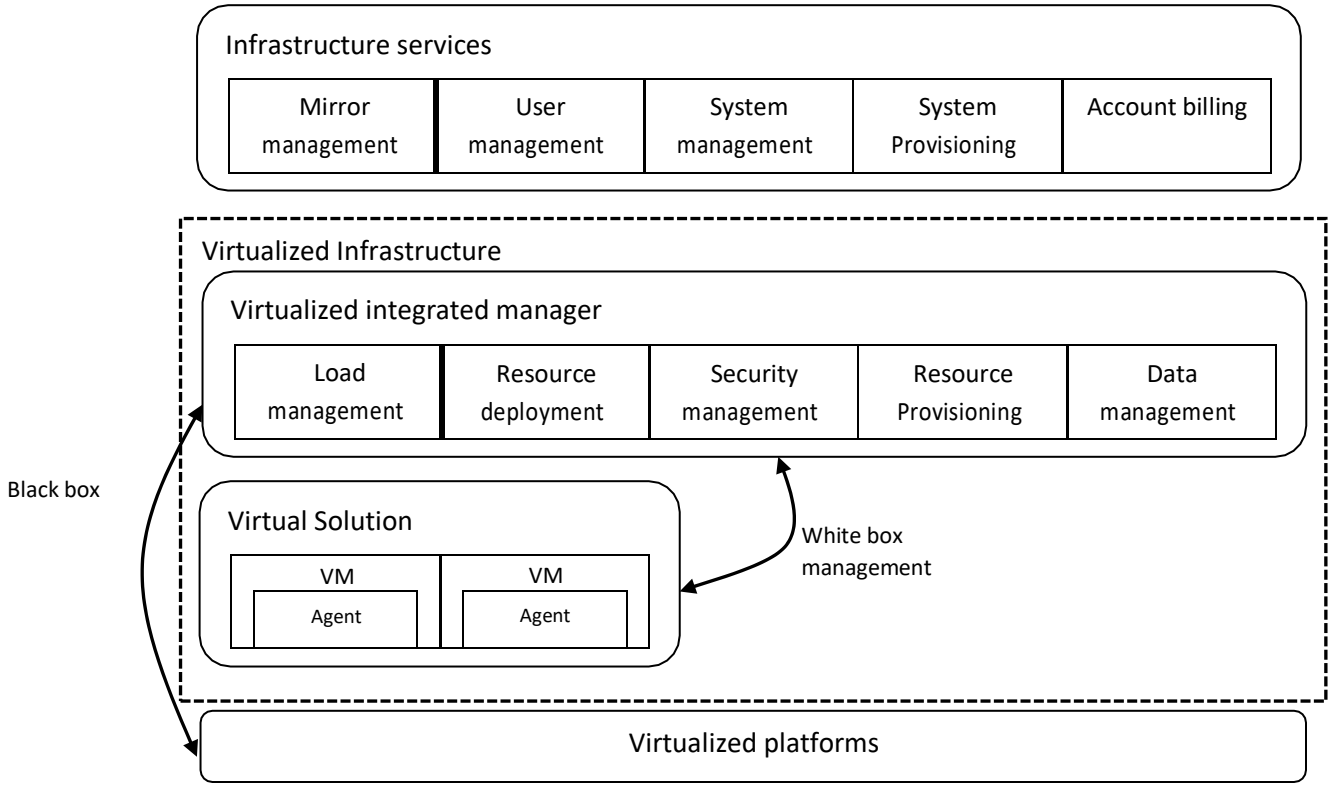


Figure 2.12 Virtualized servers, storage, and network for cloud platform construction

- 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 soware as the running environment for legacy software such as old operating systems and unusual applications.

2.10.1 Hardware Virtualization

- 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.

- VMs provide flexible runtime services to free users from worrying about the system environment.

- Using VMs in a cloud computing platform ensures extreme flexibility for users. As the computing resources are shared by many users, a method is required to maximize the user's privileges and still keep them separated safely.

- Traditional sharing of cluster resources depends on the user and group mechanism on a system.

    - Such sharing is not flexible.
    - Users cannot customize the system for their special purposes.
    - Operating systems cannot be changed.
    - The separation is not complete.

- An environment that meets one user's requirements often cannot satisfy another user. Virtualization allows us to have full privileges while keeping them separate.

- Users have full access to their own VMs, which are completely separate from other user's VMs.

- Multiple VMs can be mounted on the same physical server. Different VMs may run with different OSes.

- The virtualized resources form a resource pool.

- The virtualization is carried out by special servers dedicated to generating the virtualized resource pool.

- The virtualized infrastructure (black box in the middle) is built with many virtualizing integration managers.

- These managers handle loads, resources, security, data, and provisioning functions. Figure 2.13 shows two VM platforms.

- Each platform carries out a virtual solution to a user job. All cloud services are managed in the boxes at the top.

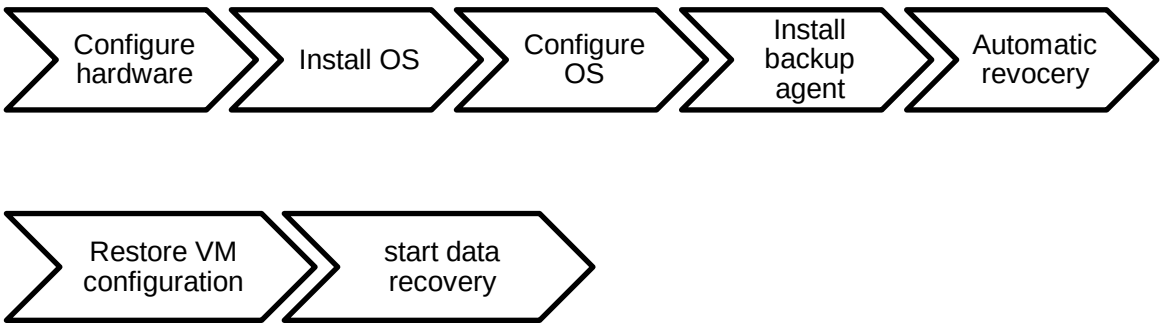


Figure 2.13 Conventional disaster recover scheme versus live migration of VMs

2.10.2 Virtualization Support in Public Clouds

- 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.
- The XenEnterprise tool applies only to Xen-based servers.

2.10.3 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:
    - System administrators consolidate workloads of underutilized servers in fewer servers
    - VMs have the ability to run legacy code without interfering with other APIs
    - VMs can be used to improve security through creation of sandboxes for running applications with questionable reliability

- Virtualized cloud platforms can apply performance isolation, letting providers offer some guarantees and better QoS to customer applications.

2.10.4 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.
- As shown in the top timeline of Figure 2.13, 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.
- Virtualization aids in fast disaster recovery by VM encapsulation.
- The cloning of VMs offers an effective solution.
- The idea is to make a clone VM on a remote server for every running VM on a local server.
- Among the entire clone VMs, only one needs to be active.
- The remote VM should be in a suspended mode.

- A cloud control center should be able to activate this clone VM in case of failure of the original VM, taking a snapshot of the VM to enable live migration in a minimal amount of time.

- The migrated VM can run on a shared Internet connection. Only updated data and modified states are sent to the suspended VM to update its state.

- The Recovery Property Objective (RPO) and Recovery Time Objective (RTO) are affected by the number of snapshots taken.

- Security of the VMs should be enforced during live migration of VMs.

TWO MARK QUESTIONS

1. List the four major characteristics to identify a service.

    - Boundaries are explicit.
    - Services are autonomous.
    - Services share schema and contracts,
    - Services compatibility is determined based on policy.

2. Define SOA.

    - Service Oriented architecture is an architectural style supporting service orientation.
    - It organizes a software system into a collection of interacting services.
    - SOA encompasses a set of design principles that structure system development and provide means for integrating components into a coherent and decentralized system.

3. List the two major roles in SOA.

    - The service provider and the service consumer.
    - The service provider is the maintainer of the service and the organization that makes available one or more services for others to use.
    - The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.

4. Characterize SOA platforms within an enterprise context.

    - Standardized service contract
    - Loose coupling
    - Abstraction
    - Reusability
    - Autonomy
    - Lack of state
    - Discoverability

5. Define Web services.

- Web services are the prominent technology for implementing SOA systems and applications.
- The concept behind a Web service is very simple.
- Using as a basis the object oriented abstraction, a Web service exposes a set of operations that can be invoked by leveraging Internet based protocols.

6. List the aspects that make Web services the technology of choice for SOA.

- First, they allow for interoperability across different platforms and programming languages.
- Second, they are based on well-known and vendor-independent standards such as HTTP, SOAP, XML, and WSDL.
- Third, they provide an intuitive and simple way to connect heterogeneous software systems
- Finally, they provide the features required by enterprise business applications to be used in an industrial environment.

7. What is the purpose of WSDL and UDDI?

- The service description document, expressed by means of Web Service Definition Language (WSDL), can be either uploaded to a global registry or attached as a metadata to the service itself.
- Service consumers can look up and discover services in global catalogs using Universal Description Discovery and Integration (UDDI) or, most likely, directly retrieve the service metadata by interrogating the Web service first.

8. What is SOAP?

- Simple Object Access Protocol (SOAP), an XML-based language for exchanging structured information in a platform-independent manner, constitutes the protocol used for Web service method invocation.

9. Write short note on RESTful systems.

- Representational State Transfer (REST) provides a model for designing network-based software systems utilizing the client/ server model and leverages the facilities provided by HTTP for IPC without additional burden.

10. What is Publish-and-subscribe message model?

- Publish-and-subscribe message model introduces a different message passing strategy, one that is based on notification among components.
- There are two major roles: The publisher and the subscriber.
- It is very suitable for implementing systems based on the one-to-many communication model

11. List the merits of Virtualization.

- Virtualization technology is one of the fundamental components of cloud computing, especially in regard to infrastructure-based services.
- Virtualization allows the creation of a secure, customizable, and isolated execution environment for running applications, even if they are untrusted, without affecting other users' applications.

12. List characteristics of virtualized environments.

- Increased security
    - Sharing
    - Aggregation
    - Emulation

    - Isolation
- Performance tuning.
- Portability

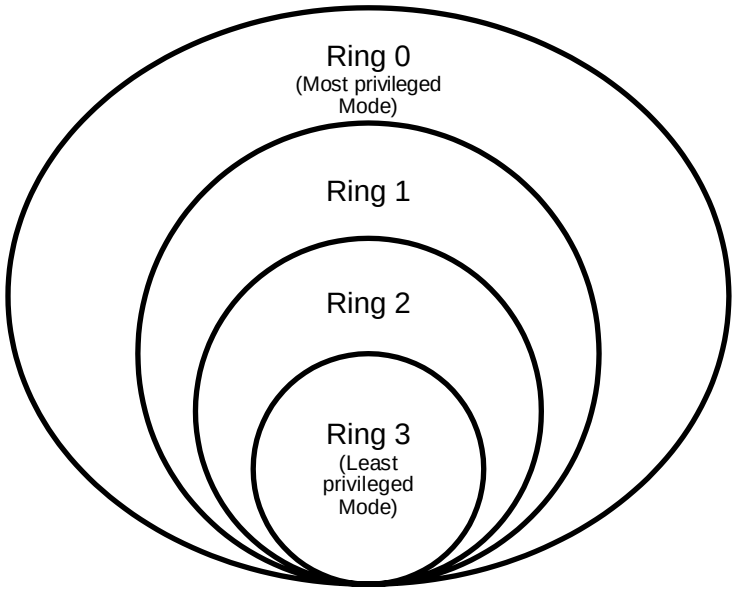
13. Categorize execution virtualization techniques.

- Process level techniques are implemented on top of an existing operating system, which has full control of the hardware.
- System level techniques are implemented directly on hardware and do not require or require a minimum of support from an existing operating system.

14. Differentiate between privileged and non privileged instructions.

- Non privileged instructions are those instructions that can be used without interfering with other tasks because they do not access shared resources.
- Privileged instructions are those that are executed under specific restrictions and are mostly used for sensitive operations, which expose (behavior-sensitive) or modify (control-sensitive) the privileged state.

15. Illustrate ring based security.



16. What is Hardware-level virtualization?

- Hardware-level virtualization is a virtualization technique that provides an abstract execution environment in terms of computer hardware on top of which a guest operating system can be run.

17. Define hypervisor.

- The hypervisor is generally a program or a combination of software and hardware that allows the abstraction of the underlying physical hardware.
- Hypervisors is a fundamental element of hardware virtualization is the hypervisor, or virtual machine manager (VMM).

18. List the types of hypervisor.

- Type I hypervisors run directly on top of the hardware.
- Type II hypervisors require the support of an operating system to provide virtualization services.

19. What is hardware assisted virtualization?

- This term refers to a scenario in which the hardware provides architectural support for building a virtual machine manager able to run a guest operating system in complete isolation.
- This technique was originally introduced in the IBM System/370.

20. Compare Full virtualization and Paravirtualization

- Full virtualization refers to the ability to run a program, most likely an operating system, directly on top of a virtual machine and without any modification, as though it were run on the raw hardware.
- Paravirtualization is a not-transparent virtualization solution that allows implementing thin virtual machine managers.

- Paravirtualization techniques expose a software interface to the virtual machine that is slightly modified from the host and, as a consequence, guests need to be modified.

21. How to virtualization implemented in partial virtualization?

- Partial virtualization provides a partial emulation of the underlying hardware, thus not allowing the complete execution of the guest operating system in complete isolation.
- Partial virtualization allows many applications to run transparently, but not all the features of the operating system can be supported, as happens with full virtualization.

22. What is Operating system-level?

- Operating system-level virtualization offers the opportunity to create different and separated execution environments for applications that are managed concurrently.
- Differently from hardware virtualization, there is no virtual machine manager or hypervisor, and the virtualization is done within a single operating system, where the OS kernel allows for multiple isolated user space instances.

23. What is storage virtualization?

- Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation.
- Using this technique, users do not have to be worried about the specific location of their data, which can be identified using a logical path.

24. Define Desktop virtualization.

- Desktop virtualization abstracts the desktop environment available on a personal computer in order to provide access to it using a client/server approach.
- Desktop virtualization provides the same outcome of hardware virtualization but serves a different purpose.

25. List the merits of Virtualization at various implementation levels.

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
Instruction Set Architecture	Very Low	Very High	Moderate	Moderate
Hardware-level virtualization	Very High	Moderate	Very High	High
OS-level virtualization	Very High	Low	Moderate	Low
Library support level	Moderate	Low	Low	Low
User application level	Low	Low	Very High	Very High

26. Differentiate between micro-kernel and monolithic hypervisor.

- A micro-kernel hypervisor includes only the basic and unchanging functions (such as physical memory management and processor scheduling).
- A monolithic hypervisor implements all the aforementioned functions, including those of the device drivers.

UNIT III CLOUD ARCHITECTURE, SERVICES AND STORAGE

Layered Cloud Architecture Design –NIST Cloud Computing Reference Architecture –Public, Private and Hybrid Clouds -laaS –PaaS –SaaS –Architectural Design Challenges –Cloud Storage –Storage-as-a-Service –Advantages of Cloud Storage –Cloud Storage Providers –S3.

3.1 Layered Cloud Architecture Design

- 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.

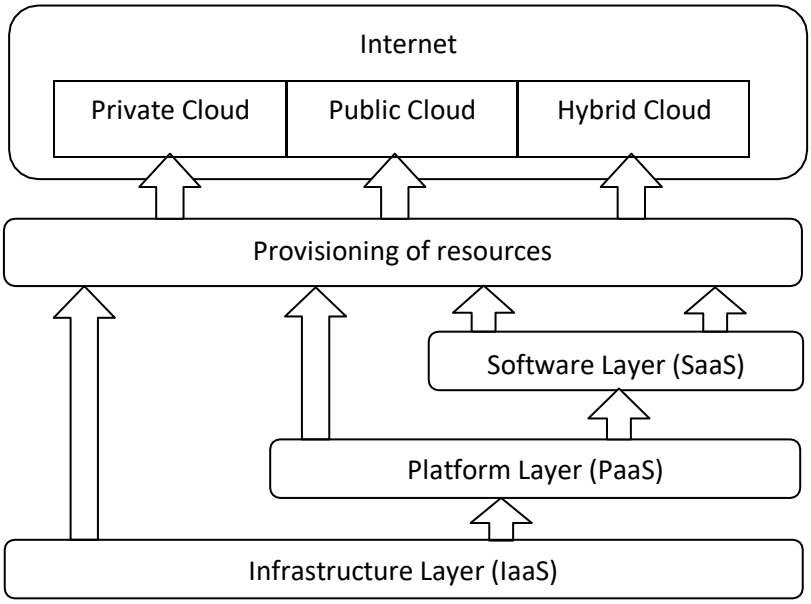


Figure 3.1 Layered architectural development

- 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.

- The application layer is formed with a collection of all needed software modules for SaaS applications.

- Service applications in this layer include daily office management work such as information retrieval, document processing and calendar and authentication services.

- The application layer is also heavily used by enterprises in business marketing and sales, consumer relationship management (CRM), financial transactions and supply chain management.

- From the provider’s perspective, the services at various layers demand different amounts of functionality support and resource management by providers.

- In general, SaaS demands the most work from the provider, PaaS is in the middle, and IaaS demands the least.

- For example, Amazon EC2 provides not only virtualized CPU resources to users but also management of these provisioned resources.

- Services at the application layer demand more work from providers.

- The best example of this is the Salesforce.com CRM service in which the provider supplies not only the hardware at the bottom layer and the software at the top layer but also the platform and software tools for user application development and monitoring.

- In 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.

- Market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand.

- This cloud is basically built with the following entities:

    - 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.
    - The request examiner ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources.
    - The Pricing mechanism decides how service requests are charged. For instance, requests can be charged based on submission time (peak/off-peak), pricing rates (fixed/changing), or availability of resources (supply/demand).
    - The VM Monitor mechanism keeps track of the availability of VMs and their resource entitlements.
    - The Accounting mechanism maintains the actual usage of resources by requests so that the final cost can be computed and charged to users.
    - In addition, the maintained historical usage information can be utilized by the Service Request Examiner and Admission Control mechanism to improve resource allocation decisions.
    - 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.

3.2 NIST Cloud Computing Reference Architecture

- NIST stands for National Institute of Standards and Technology

- The goal is to achieve effective and secure cloud computing to reduce cost and improve services

- NIST composed for six major workgroups specific to cloud computing

- Cloud computing target business use cases work group
- Cloud computing Reference architecture and Taxonomy work group
- Cloud computing standards roadmap work group
- Cloud computing SAJACC (Standards Acceleration to Jumpstart Adoption of Cloud Computing) work group
- Cloud Computing security work group

- Objectives of NIST Cloud Computing reference architecture
  - Illustrate and understand the various level of services
  - To provide technical reference
  - Categorize and compare services of cloud computing
  - Analysis of security, interoperatability and portability

- In general, NIST generates report for future reference which includes survey, analysis of existing cloud computing reference model, vendors and federal agencies.

- The conceptual reference architecture shown in figure 3.2 involves five actors. Each actor as entity participates in cloud computing

- Cloud consumer: A person or an organization that maintains a business relationship with and uses a services from cloud providers

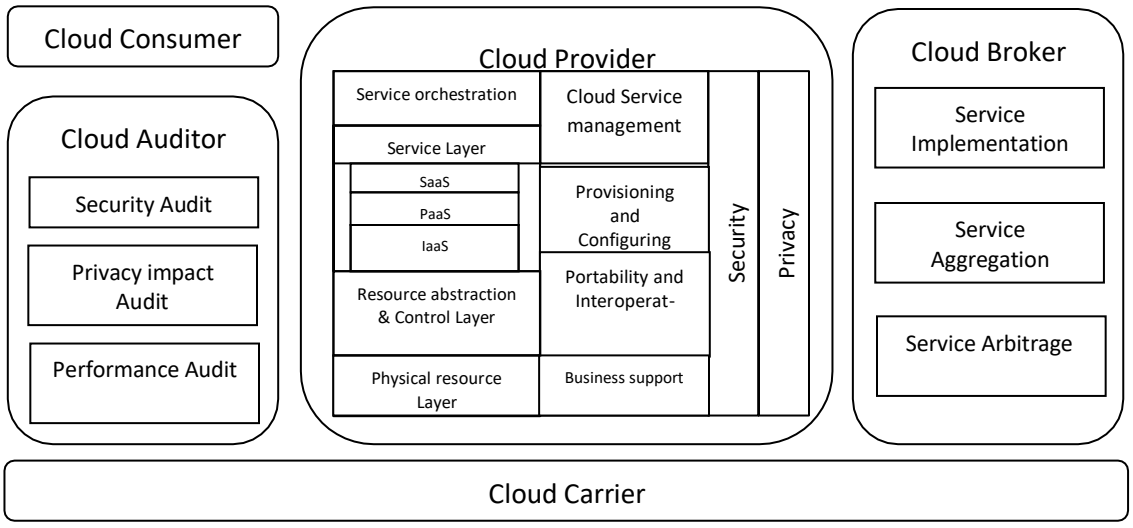


Figure 3.2 Conceptual reference model

- Cloud provider: A person, organization or entity responsible for making a service available to interested parties

- Cloud auditor: A party that conduct independent assessment of cloud services, information system operation, performance and security of cloud implementation

- Cloud broker: An entity that manages the performance and delivery of cloud services and negotiates relationship between cloud provider and consumer.

- Cloud carrier: An intermediary that provides connectivity and transport of cloud services from cloud providers to consumers.

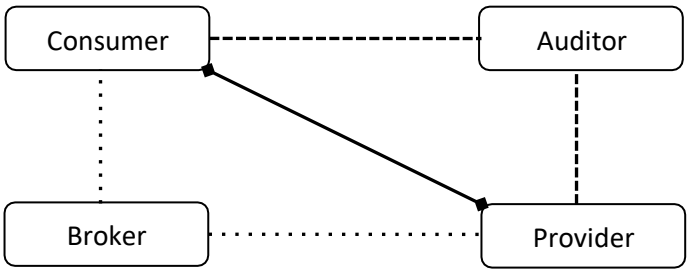


Figure 3.3 Interaction between actors

- Figure 3.3 illustrates the common interaction exist in between cloud consumer and provider where as the broker used to provide service to consumer and auditor collects the audit information.

- The interaction between the actors may lead to different use case scenario.

- Figure 3.4 shows one kind of scenario in which the Cloud consumer may request service from a cloud broker instead of contacting service provider directly. In this case, a cloud broker can create a new service by combining multiple services.

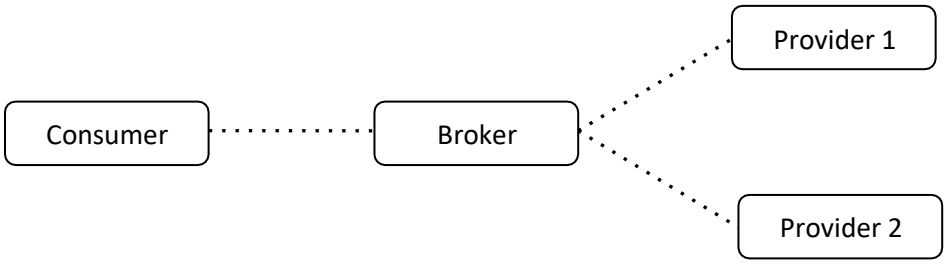


Figure 3.4 Service from Cloud Broker

- Figure 3.5 illustrates the usage of different kind of Service Level Agreement (SLA) between consumer, provider and carrier.

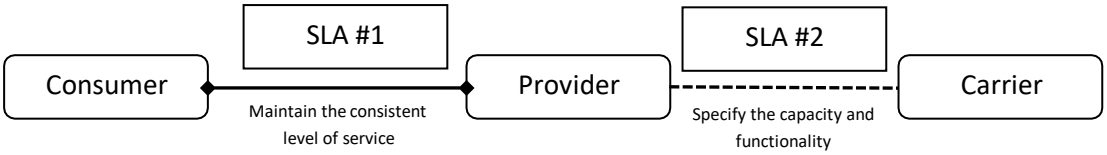


Figure 3.5 Multiple SLA between actors

- Figure 3.6 shows the scenario where the Cloud auditor conducts independent assessment of operation and security of the cloud service implementation.

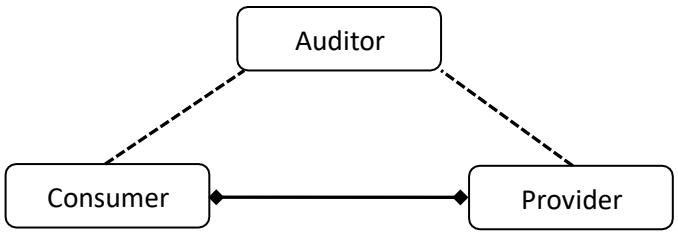


Figure 3.6 Independent assessments by cloud auditor

- Cloud consumer is a principal stake holder for the cloud computing service and requires service level agreements to specify the performance requirements fulfilled by a cloud provider.

- The service level agreement covers Quality of Service and Security aspects.

- Consumers have limited rights to access the software applications.

- There are three kinds of cloud consumers: SaaS consumers, PaaS Consumers and IaaS consumers.

- SaaS consumers are members directly access the software application. For example, document management, content management, social networks, financial billing and so on.

- PaaS consumers are used to deploy, test, develop and manage applications hosted in cloud environment. Database application deployment, development and testing is an example for these kind of consumer.

- IaaS Consumer can access the virtual computer, storage and network infrastructure. For example, usage of Amazon EC2 instance to deploy the web application.

- On the other hand, Cloud Providers have complete rights to access software applications.

- In Software as a Service model, cloud provider is allowed to configure, maintain and update the operations of software application.

- Management process is done by Integrated Development environment and Software Development Kit in Platform as a Service model.

- Infrastructure as a Service model covers Operating System and Networks.

- Normally, the service layer defines the interfaces for cloud consumers to access the computing services.

- Resource abstraction and control layer contains the system components that cloud provider use to provide and mange access to the physical computing resources through software abstraction.

- Resource abstraction covers virtual machine management and virtual storage management.

- Control layer focus on resource allocation, access control and usage monitoring.

- Physical resource layer includes physical computing resources such as CPU, Memory, Router, Switch, Firewalls and Hard Disk Drive.

- Service orchestration describes the automated arrangement, coordination and management of complex computing system.

- In cloud service management, business support entails the set of business related services dealing with consumer and supporting services which includes content management, contract management, inventory management, accounting service, reporting service and rating service.

- Provisioning of equipments, wiring and transmission is mandatory to setup a new service that provides a specific application to cloud consumer. Those details are described in Provisioning and Configuring management.

- Portability enforces the ability to work in more than one computing environment without major task. Similarly, Interoperatability means the ability of the system work with other system.

- Security factor is applicable to enterprise and Government. It may include privacy.

- Privacy is one applies to a cloud consumer's rights to safe guard his information from other consumers are parties.

- The main aim of Security and Privacy in cloud service management is to protect the system from vulnerable customers.

- Cloud auditor performs independent assessments among the services and cloud broker act as intermediate module.

- Service intermediation enhances a given service by improving some specific capability and providing value added services to cloud consumers,

- Service aggregation provides data integration. Cloud broker combines and integrate multiple service into one or more new services.

- Due to Service arbitrage, cloud broker has a flexibility to choose services from multiple providers.

- Cloud carrier is an intermediary that provides connectivity and transport of cloud service between cloud consumer and cloud provider.

- It provides access to cloud consumer with the help of network, telecommunication and other access devices where as distribution is done with transport agent,

- Transport agent is the business organization that provides physical transport of storage media.

3.3 Cloud Deployment Model

- As identified in the NIST cloud computing definition, a cloud infrastructure may be operated in one of the following deployment models: public cloud, private cloud, community cloud, or hybrid cloud.

- The differences are based on how exclusive the computing resources are made to a Cloud Consumer.

3.3.1 Public Cloud

- A public cloud is one in which the cloud infrastructure and computing resources are made available to the general public over a public network.

- A public cloud is owned by an organization selling cloud services, and serves a diverse pool of clients.

- Figure 4.7 presents a simple view of a public cloud and its customers.

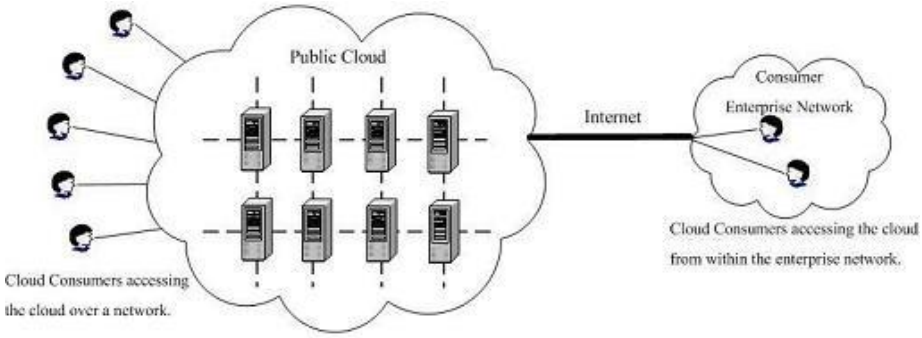


Figure 3.7 Public Cloud

3.3.1.1 Benefits of choosing a Public Cloud

- One of the main benefits that come with using public cloud services is near unlimited scalability.

- The resources are pretty much offered based on demand. So any changes in activity level can be handled very easily.

- This in turn brings with it cost effectiveness.

- Public cloud allows pooling of a large number of resources, users are benefiting from the savings of large scale operations.

- There are many services like Google Drive which are offered for free.

- Finally, the vast network of servers involved in public cloud services means that it can benefit from greater reliability.

- Even if one data center was to fail entirely, the network simply redistributes the load among the remaining enters making it highly unlikely that the public cloud would ever fail.

- In summary, the benefits of the public cloud are:

    - Easy scalability
    - Cost effectiveness
    - Increased reliability

3.3.1.2 Disadvantages of choosing a Public Cloud

- There are of course downsides to using public cloud services.

- At the top of the list is the fact that the security of data held within a public cloud is a cause for concern.

- It is often seen as an advantage that the public cloud has no geographical restrictions making access easy from everywhere, but on the flip side this could mean that the server is in a different country which is governed by an entirely different set of security and/or privacy regulations.

- This could mean that your data is not all that secure making it unwise to use public cloud services for sensitive data.

3.3.2 Private Cloud

- A private cloud gives a single Cloud Consumer's organization the exclusive access to and usage of the infrastructure and computational resources.

- It may be managed either by the Cloud Consumer organization or by a third party, and may be hosted on the organization's premises (i.e. on-site private clouds) or outsourced to a hosting company (i.e. outsourced private clouds).

- Figure 3.8 presents an on-site private cloud and an outsourced private cloud, respectively.

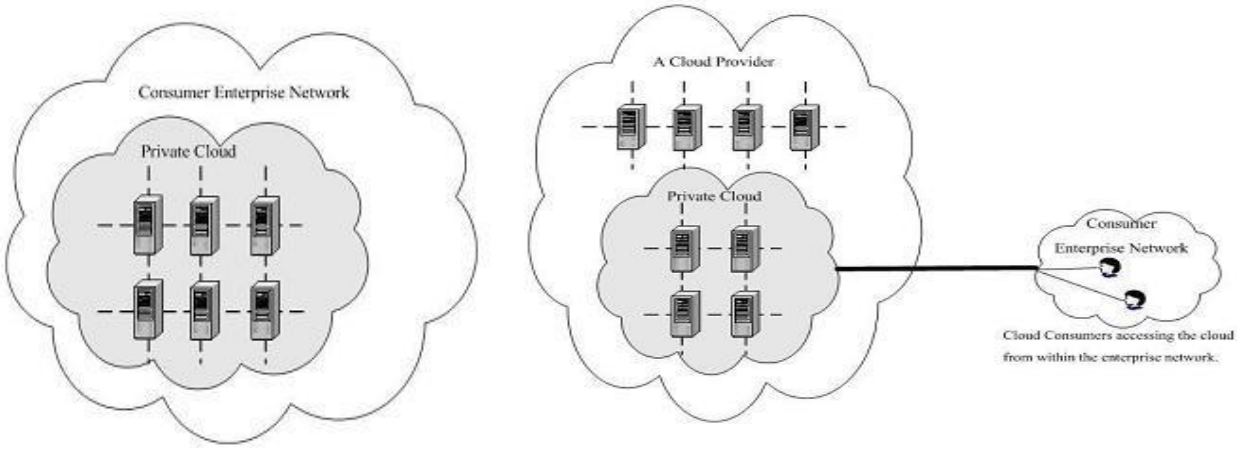


Figure 3.8 (a) On-site Private Cloud (b) Out-sourced Private Cloud

3.3.2.1 Benefits of choosing a Private Cloud

- The main benefit of choosing a private cloud is the greater level of security offered making it ideal for business users who need to store and/or process sensitive data.

- A good example is a company dealing with financial information such as bank or lender who is required by law to use secure internal storage to store consumer information.

- With a private cloud this can be achieved while still allowing the organization to benefit from cloud computing.

- Private cloud services also offer some other benefits for business users including more control over the server allowing it to be tailored to your own preferences and in house styles.

- While this can remove some of the scalability options, private cloud providers often offer what is known as cloud bursting which is when non sensitive data is switched to a public cloud to free up private cloud space in the event of a significant spike in demand until such times as the private cloud can be expanded.

- In summary, the main benefits of the private cloud are:

    - Improved security
    - Greater control over the server
    - Flexibility in the form of Cloud Bursting

3.3.2.2 Disadvantages of choosing a Private Cloud

- The downsides of private cloud services include a higher initial outlay, although in the long term many business owners find that this balances out and actual becomes more cost effective than public cloud use.

- It is also more difficult to access the data held in a private cloud from remote locations due to the increased security measures.

3.3.3 Community Cloud

- A community cloud serves a group of Cloud Consumers which have shared concerns such as mission objectives, security, privacy and compliance policy, rather than serving a single organization as does a private cloud.

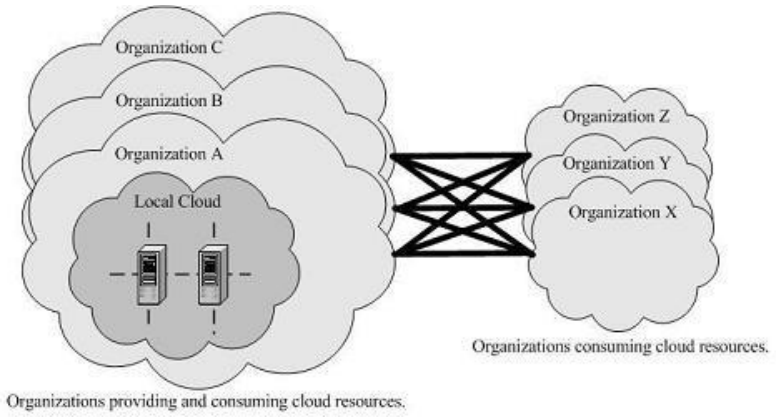
- Similar to private clouds, a community cloud may be managed by the organizations or by a third party and may be implemented on customer premise (i.e. on-site community cloud) or outsourced to a hosting company (i.e. outsourced community cloud).

- Figure 3.9 (a) depicts an on-site community cloud comprised of a number of participant organizations.

- A cloud consumer can access the local cloud resources, and also the resources of other participating organizations through the connections between the associated organizations.

- Figure 3.9 (b) shows an outsourced community cloud, where the server side is outsourced to a hosting company.

- In this case, an outsourced community cloud builds its infrastructure off premise, and serves a set of organizations that request and consume cloud services.



(a) On-site Community Cloud

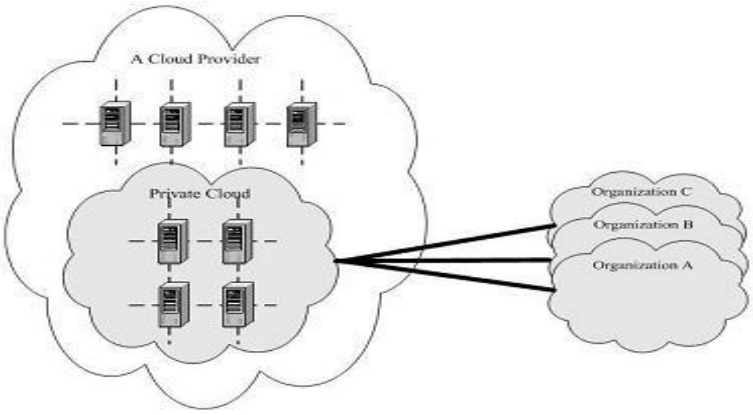


Figure 3.9 (b) Outsourced Community Cloud

3.3.3.1 Benefits of Choosing a Community Cloud

- Ability to easily share and collaborate
- Lower cost

3.3.3.2 Disadvantages of Choosing a Community Cloud

- Not the right choice for every organization
- Slow adoption to date

3.3.4 Hybrid Cloud

- A hybrid cloud is a composition of two or more clouds (on-site private, on-site community, off-site private, off-site community or public) that remain as distinct entities but are bound together by standardized or proprietary technology that enables data and application portability.

- Figure 3.10 illustrates a simple view of a hybrid cloud that could be built with a set of clouds in the five deployment model variants.

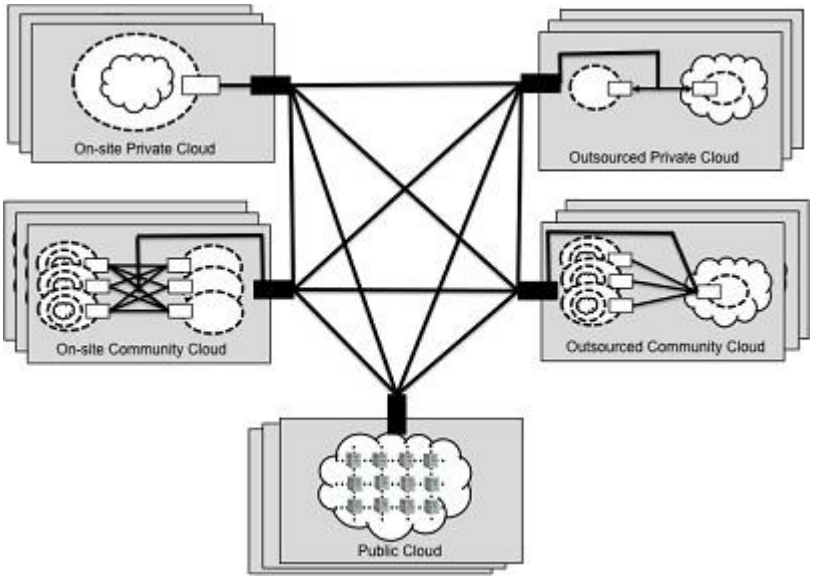


Figure 3.10 Hybrid Cloud

3.4 Cloud Service Model

- The development of cloud computing introduces the concept of everything as a Service (XaaS). This is one of the most important elements of cloud computing

- Cloud services from different providers can be combined to provide a completely integrated solution covering all the computing stack of a system.

- IaaS providers can offer the bare metal in terms of virtual machines where PaaS solutions are deployed.

- When there is no need for a PaaS layer, it is possible to directly customize the virtual infrastructure with the software stack needed to run applications.

- This is the case of virtual Web farms: a distributed system composed of Web servers, database servers and load balancers on top of which prepackaged software is installed to run Web applications.

- Other solutions provide prepackaged system images that already contain the software stack required for the most common uses: Web servers, database servers or LAMP stacks.

- Besides the basic virtual machine management capabilities, additional services can be provided, generally including the following:

    - SLA resource based allocation
    - Workload management
    - Support for infrastructure design through advanced Web interfaces
    - Integrate third party IaaS solutions

- Figure 3.11 provides an overall view of the components forming an Infrastructure as a Service solution.

- It is possible to distinguish three principal layers:

    - Physical infrastructure
    - Software management infrastructure
    - User interface

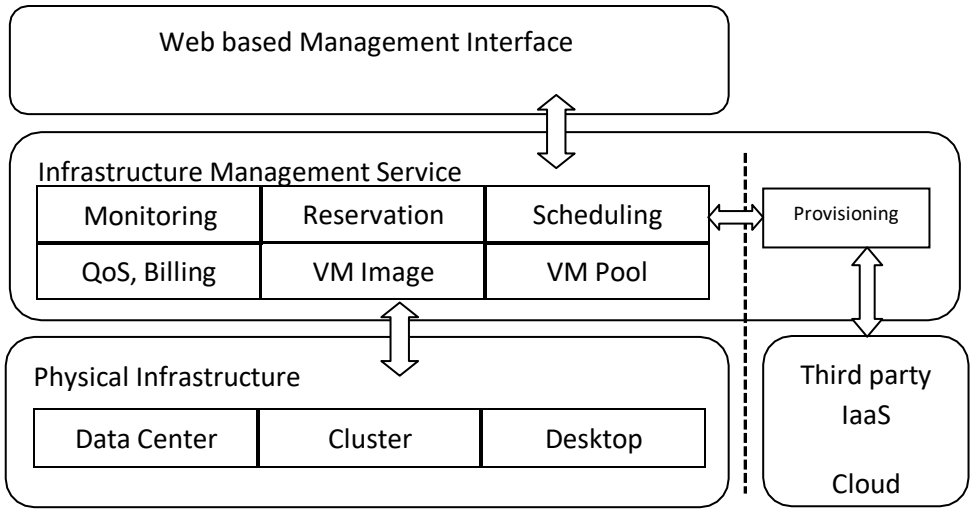


Figure 3.11 IaaS reference implementation

- At the top layer the user interface provides access to the services exposed by the software management infrastructure.

- Such an interface is generally based on Web 2.0 technologies: Web services, RESTful APIs and mash ups.

- Web services and RESTful APIs allow programs to interact with the service without human intervention, thus providing complete integration within a software system.

- The core features of an IaaS solution are implemented in the infrastructure management software layer.

- In particular, management of the virtual machines is the most important function performed by this layer.

- A central role is played by the scheduler, which is in charge of allocating the execution of virtual machine instances.

- The scheduler interacts with the other components such as

    - Pricing and billing component
    - Monitoring component
    - Reservation component
    - QoS/SLA management component
    - VM repository component
    - VM pool manager component
    - Provisioning component

- The bottom layer is composed of the physical infrastructure, on top of which the management layer operates.

- From an architectural point of view, the physical layer also includes the virtual resources that are rented from external IaaS providers.

- In the case of complete IaaS solutions, all three levels are offered as service.

- This is generally the case with public clouds vendors such as Amazon, GoGrid, Joyent, Rightscale, Terremark, Rackspace, ElasticHosts, and Flexiscale, which own large datacenters and give access to their computing infrastructures using an IaaS approach.

3.4.1 IaaS

- Infrastructure or Hardware as a Service (IaaS/HaaS) solutions are the most popular and developed market segment of cloud computing.

- They deliver customizable infrastructure on demand.

- The available options within the IaaS offering umbrella range from single servers to entire infrastructures, including network devices, load balancers, database servers and Web servers.

- The main technology used to deliver and implement these solutions is hardware virtualization: one or more virtual machines opportunely configured and interconnected define the distributed system on top of which applications are installed and deployed.

- Virtual machines also constitute the atomic components that are deployed and priced according to the specific features of the virtual hardware: memory, number of processors and disk storage.

- IaaS/HaaS solutions bring all the benefits of hardware virtualization: workload partitioning, application isolation, sandboxing and hardware tuning.

- From the perspective of the service provider, IaaS/HaaS allows better exploiting the IT infrastructure and provides a more secure environment where executing third party applications.

- From the perspective of the customer, it reduces the administration and maintenance cost as well as the capital costs allocated to purchase hardware.

- At the same time, users can take advantage of the full customization offered by virtualization to deploy their infrastructure in the cloud.

3.4.2 PaaS

- Platform as a Service (PaaS) solutions provide a development and deployment platform for running applications in the cloud.

- They constitute the middleware on top of which applications are built.

- A general overview of the features characterizing the PaaS approach is given in Figure 3.12.

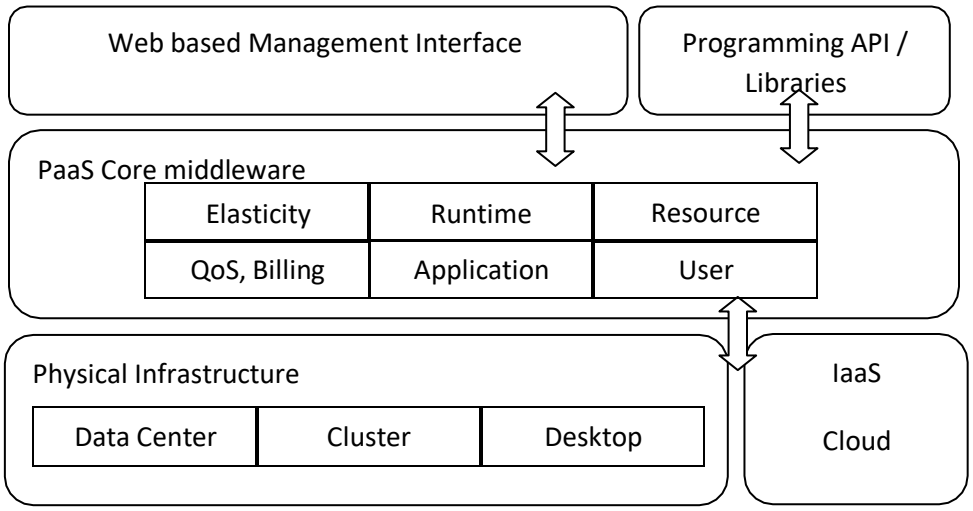


Figure 3.12 PaaS reference implementation

- Application management is the core functionality of the middleware.

- PaaS implementations provide applications with a runtime environment and do not expose any service for managing the underlying infrastructure.

- They automate the process of deploying applications to the infrastructure, configuring application components, provisioning and configuring supporting technologies such as load balancers and databases and managing system change based on policies set by the user.

- The core middleware is in charge of managing the resources and scaling applications on demand or automatically, according to the commitments made with users.

- From a user point of view, the core middleware exposes interfaces that allow programming and deploying applications on the cloud.

- Some implementations provide a completely Web based interface hosted in the cloud and offering a variety of services.

- It is possible to find integrated developed environments based on 4GL and visual programming concepts or rapid prototyping environments where applications are built by assembling mash ups and user defined components and successively customized.

- Other implementations of the PaaS model provide a complete object model for representing an application and provide a programming language-based approach.

- Developers generally have the full power of programming languages such as Java, .NET, Python and Ruby with some restrictions to provide better scalability and security.

- PaaS solutions can offer middleware for developing applications together with the infrastructure or simply provide users with the software that is installed on the user premises.

- In the first case, the PaaS provider also owns large datacenters where applications are executed

- In the second case, referred to in this book as Pure PaaS, the middleware constitutes the core value of the offering.

- PaaS implementation classified into three wide categories:

    - PaaS-I
    - PaaS-II
    - PaaS-III

- The first category identifies PaaS implementations that completely follow the cloud computing style for application development and deployment.

    - They offer an integrated development environment hosted within the Web browser where applications are designed, developed, composed, and deployed.
    - This is the case of Force.com and Longjump. Both deliver as platforms the combination of middleware and infrastructure.

- In the second class focused on providing a scalable infrastructure for Web application, mostly websites.

    - In this case, developers generally use the provider's APIs, which are built on top of industrial runtimes, to develop applications.
    - Google AppEngine is the most popular product in this category.
    - It provides a scalable runtime based on the Java and Python programming languages, which have been modified for providing a secure runtime environment and enriched with additional APIs and components to support scalability.

- AppScale, an open source implementation of Google AppEngine, provides interfacecompatible middleware that has to be installed on a physical infrastructure.

- The third category consists of all those solutions that provide a cloud programming platform for any kind of application, not only Web applications.

  - Among these, the most popular is Microsoft Windows Azure, which provides a comprehensive framework for building service oriented cloud applications on top of the .NET technology, hosted on Microsoft's datacenters.
  - Other solutions in the same category, such as Manjrasoft Aneka, Apprenda SaaSGrid, Appistry Cloud IQ Platform, DataSynapse, and GigaSpaces DataGrid, provide only middleware with different services.

- Some essential characteristics that identify a PaaS solution:

  - Runtime framework: This framework represents the software stack of the PaaS model and the most intuitive aspect that comes to people's minds when they refer to PaaS solutions.
  - Abstraction: PaaS solutions are distinguished by the higher level of abstraction that they provide.
  - Automation: PaaS environments automate the process of deploying applications to the infrastructure, scaling them by provisioning additional resources when needed.
  - Cloud services: PaaS offerings provide developers and architects with services and APIs, helping them to simplify the creation and delivery of elastic and highly available cloud application.

3.4.3 SaaS

- Software as a Service (SaaS) is a software delivery model that provides access to applications through the Internet as a Web based service.

- It provides a means to free users from complex hardware and software management by offloading such tasks to third parties, which build applications accessible to multiple users through a Web browser.

- On the provider side, the specific details and features of each customer's application are maintained in the infrastructure and made available on demand.

- The SaaS model is appealing for applications serving a wide range of users and that can be adapted to specific needs with little further customization.

- This requirement characterizes SaaS as a one-to-many software delivery model, whereby an application is shared across multiple users.

- This is the case of CRM and ERP applications that constitute common needs for almost all enterprises, from small to medium-sized and large business.

- Every enterprise will have the same requirements for the basic features concerning CRM and ERP and different needs can be satisfied with further customization.

- SaaS applications are naturally multitenant.

- Multitenancy, which is a feature of SaaS compared to traditional packaged software, allows providers to centralize and sustain the effort of managing large hardware infrastructures, maintaining as well as upgrading applications transparently to the users and optimizing resources by sharing the costs among the large user base.

- On the customer side, such costs constitute a minimal fraction of the usage fee paid for the software.

- The analysis carried out by Software Information and Industry Association (SIIA) was mainly oriented to cover application service providers (ASPs) and all their variations,

which capture the concept of software applications consumed as a service in a broader sense.

- ASPs already had some of the core characteristics of SaaS:
    - The product sold to customer is application access
    - The application is centrally managed
    - The service delivered is one-to-many
    - The service delivered is an integrated solution delivered on the contract, which means provided as promised.
- ASPs provided access to packaged software solutions that addressed the needs of a variety of customers.
- Initially this approach was affordable for service providers, but it later became inconvenient when the cost of customizations and specializations increased.
- The SaaS approach introduces a more flexible way of delivering application services that are fully customizable by the user by integrating new services, injecting their own components and designing the application and information workflows.
- Initially the SaaS model was of interest only for lead users and early adopters.
- The benefits delivered at that stage were the following:
    - Software cost reduction and total cost of ownership (TCO) were paramount
    - Service level improvements
    - Rapid implementation
    - Standalone and configurable applications
    - Rudimentary application and data integration
    - Subscription and pay as you go (PAYG) pricing

- With the advent of cloud computing there has been an increasing acceptance of SaaS as a viable software delivery model.

- This lead to transition into SaaS 2.0, which does not introduce a new technology but transforms the way in which SaaS is used.

- In particular, SaaS 2.0 is focused on providing a more robust infrastructure and application platforms driven by SLAs.

- SaaS 2.0 will focus on the rapid achievement of business objectives.

- Software as a Service based applications can serve different needs. CRM, ERP, and social networking applications are definitely the most popular ones.

- SalesForce.com is probably the most successful and popular example of a CRM service.

- It provides a wide range of services for applications: customer relationship and human resource management, enterprise resource planning, and many other features.

- SalesForce.com builds on top of the Force.com platform, which provides a fully featured environment for building applications.

- In particular, through AppExchange customers can publish, search and integrate new services and features into their existing applications.

- This makes SalesForce.com applications completely extensible and customizable.

- Similar solutions are offered by NetSuite and RightNow.

- NetSuite is an integrated software business suite featuring financials, CRM, inventory, and ecommerce functionalities integrated all together.

- RightNow is customer experience centered SaaS application that integrates together different features, from chat to Web communities, to support the common activity of an enterprise

- Another important class of popular SaaS applications comprises social networking applications such as Facebook and professional networking sites such as LinkedIn.

- Other than providing the basic features of networking, they allow incorporating and extending their capabilities by integrating third-party applications.

- Office automation applications are also an important representative for SaaS applications:

    - Google Documents and Zoho Office are examples of Web based applications that aim to address all user needs for documents, spreadsheets and presentation management.
    - These applications offer a Web based interface for creating, managing, and modifying documents that can be easily shared among users and made accessible from anywhere.

3.5 Architectural Design Challenges

3.5.1 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.

- In addition to mitigating data lock-in concerns, standardization of APIs enables a new usage model in which the same software infrastructure can be used in both public and private clouds.

- Such an option could enable surge computing, in which the public cloud is used to capture the extra tasks that cannot be easily run in the data center of a private cloud.

3.5.2 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 middle boxes (e.g., firewalls, packet filters).

- For example, the end user could encrypt data before placing it in a cloud. Many nations have laws requiring SaaS providers to keep customer data and copyrighted material within national boundaries.

- 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.

- On the other hand, Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

3.5.3 Challenge 3: Unpredictable Performance and Bottlenecks

- Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic.

- For example, to run 75 EC2 instances with the STREAM benchmark requires a mean bandwidth of 1,355 MB/second.

- However, for each of the 75 EC2 instances to write 1 GB files to the local disk requires a mean disk write bandwidth of only 55 MB/second.

- This demonstrates the problem of I/O interference between VMs.

- One solution is to improve I/O architectures and operating systems to efficiently virtualize interrupts and I/O channels.

- 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.

3.5.4 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 programmer's 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.

3.5.5 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 or decreases and the 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 is agnostic packaging with certification and integrity of packaged software.

- The package supports virtual appliances to span more than one VM.

- OVF also defines a transport mechanism for VM templates and the format can apply to different virtualization platforms with different levels of virtualization.

- In terms of cloud standardization, the ability for virtual appliances to run on any virtual platform.

- The user is also need to enable VMs to run on heterogeneous hardware platform hypervisors.

- This requires hypervisor-agnostic VMs.

- And also the user need to realize cross platform live migration between x86 Intel and AMD technologies and support legacy hardware for load balancing.

- All these issues are wide open for further research.

3.5.6 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.

3.6 Cloud Storage

- Cloud storage means storing the data with a cloud service provider rather than on a local system.

- The end user can access the data stored on the cloud using an Internet link.

- Cloud storage has a number of advantages over traditional data storage.

- If the users stored some data on a cloud, they can get at it from any location that has Internet access.

- Workers do not need to use the same computer to access data nor do they have to carry around physical storage devices.

- Also, if any organization has branch offices, they can all access the data from the cloud provider.

- There are hundreds of different cloud storage systems, and some are very specific in what they do.

- Some are niche-oriented and store just email or digital pictures, while others store any type of data. Some providers are small, while others are huge and fill an entire warehouse.

- At the most rudimentary level, a cloud storage system just needs one data server connected to the Internet.

- A subscriber copies files to the server over the Internet, which then records the data. When a client wants to retrieve the data, the client accesses the data server with a web based interface and the server then either sends the files back to the client or allows the client to access and manipulate the data itself.

- More typically, however, cloud storage systems utilize dozens or hundreds of data servers.

- Because servers require maintenance or repair, it is necessary to store the saved data on multiple machines, providing redundancy.

- Without that redundancy, cloud storage systems could not assure clients that they could access their information at any given time.

3.6.1 Storage-as-a-Service

- The term Storage as a Service (another Software as a Service, or SaaS, acronym) means that a third-party provider rents space on their storage to end users who lack the budget or capital budget to pay for it on their own.

- Figure 3.13 illustrates the storage as a service where the data stored in cloud storage.

- It is also ideal when technical personnel are not available or have inadequate knowledge to implement and maintain that storage infrastructure.

- Storage service providers are nothing new, but given the complexity of current backup, replication, and disaster recovery needs, the service has become popular, especially among small and medium sized businesses.

- The biggest advantage to SaaS is cost savings.

- Storage is rented from the provider using a cost-per-gigabyte-stored or cost-per-data-transferred model.

- The end user does not have to pay for infrastructure. They simply pay for how much they transfer and save on the provider's servers.

Figure 3.13 Storage as a Service

- A customer uses client software to specify the backup set and then transfers data across a WAN.

- Examples:

    - Google Docs allows users to upload documents, spreadsheets, and presentations to Google's data servers. Those files can then be edited using a Google application.
    - Web email providers like Gmail, Hotmail, and Yahoo! Mail store email messages on their own servers. Users can access their email from computers and other devices connected to the Internet.
    - Flickr and Picasa host millions of digital photographs. Users can create their own online photo albums.

    - YouTube hosts millions of user uploaded video files.
    - Hostmonster and GoDaddy store files and data for many client web sites.
    - Facebook and MySpace are social networking sites and allow members to post pictures and other content. That content is stored on the company's servers.
    - MediaMax and Strongspace offer storage space for any kind of digital data.

- To secure data, most systems use a combination of the listed techniques:

- Encryption: A complex algorithm is used to encode information. To decode the encrypted files, a user needs the encryption key.

- Authentication processes: This requires a user to create a name and password.

- Authorization practices: The client lists the people who are authorized to access information stored on the cloud system. Many corporations have multiple levels of authorization.

- The other concern is reliability.

- If a cloud storage system is unreliable, it becomes a liability. No one wants to save data on an unstable system, nor would they trust a company that is financially unstable.

- Most cloud storage providers try to address the reliability concern through redundancy, but the possibility still exists that the system could crash and leave clients with no way to access their saved data.

3.6.2 Advantages of Cloud Storage

- Cloud storage is becoming an increasingly attractive solution for organizations.

- Cloud storage providers balance server loads and move data among various datacenters, ensuring that information is stored close and thereby available quickly while using the data.

- Storing data on the cloud is advantageous, because it allows the user to protect the data in case there's a disaster.

- Having the data stored off-site can be the difference between closing the door for good or being down for a few days or weeks.

- Which storage vendor to go with can be a complex issue, and how the end user technology interacts with the cloud can be complex.

- For instance, some products are agent based and the application automatically transfers information to the cloud via FTP.

- But others employ a web front end and the user has to select local files on their computer to transmit.

- Amazon S3 is the best known storage solution, but other vendors might be better for large enterprises.

- For instance, those who offer service level agreements and direct access to customer support are critical for a business moving storage to a service provider

3.6.3 Cloud Storage Providers

- There are hundreds of cloud store providers every day.

- This is simply a listing of what some of the big players in the game have to offer and anyone can use it as a starting guide to determine if their services match user's needs.

- Amazon and Nirvanix are the current industry top dogs, but many others are in the field, including some well known names.

- Google offers cloud storage solution called GDrive.

- EMC is readying a storage solution and IBM already has a number of cloud storage options called Blue Cloud.

3.6.4 S3

- The well known cloud storage service is Amazon's Simple Storage Service (S3), which is launched in 2006.

- Amazon S3 is designed to make web scale computing easier for developers.

- Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the Web.

- It gives any developer access to the same highly scalable data storage infrastructure that Amazon uses to run its own global network of web sites.

- The service aims to maximize benefits of scale and to pass those benefits on to developers.

- Amazon S3 is intentionally built with a minimal feature set that includes the following functionality:

    - Write, read, and delete objects containing from 1 byte to 5 gigabytes of data each. The number of objects that can be stored is unlimited.
    - Each object is stored and retrieved via a unique developer assigned key.
    - Objects can be made private or public and rights can be assigned to specific users.
    - Uses standards based REST and SOAP interfaces designed to work with any Internet development toolkit.

- Design Requirements Amazon built S3 to fulfill the following design requirements:

    - Scalable: Amazon S3 can scale in terms of storage, request rate and users to support an unlimited number of web-scale applications.
    - Reliable: Store data durably with 99.99 percent availability. Amazon says it does not allow any downtime.
    - Fast: Amazon S3 was designed to be fast enough to support high-performance applications. Server-side latency must be insignificant relative to Internet latency.
    - Inexpensive: Amazon S3 is built from inexpensive commodity hardware components.
    - Simple: Building highly scalable, reliable, fast and inexpensive storage is difficult.

- Design Principles Amazon used the following principles of distributed system design to meet Amazon S3 requirements:

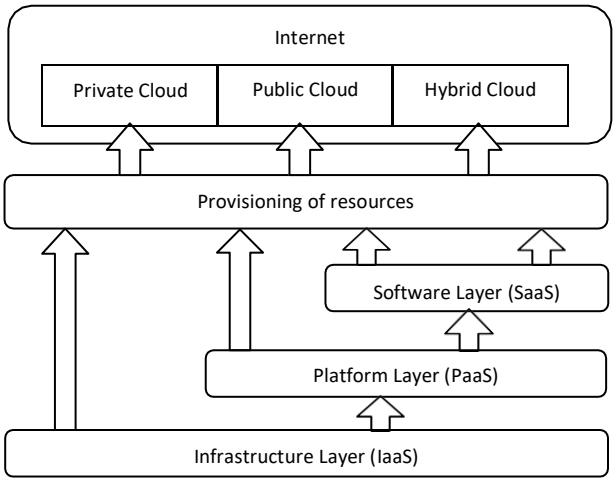
    - Decentralization: It uses fully decentralized techniques to remove scaling bottlenecks and single points of failure.
    - Autonomy: The system is designed such that individual components can make decisions based on local information.
    - Local responsibility: Each individual component is responsible for achieving its consistency. This is never the burden of its peers.
    - Controlled concurrency: Operations are designed such that no or limited concurrency control is required.
    - Failure toleration: The system considers the failure of components to be a normal mode of operation and continues operation with no or minimal interruption.
    - Controlled parallelism: Abstractions used in the system are of such granularity that parallelism can be used to improve performance and robustness of recovery or the introduction of new nodes.
    - Symmetry: Nodes in the system are identical in terms of functionality, and require no or minimal node specific configuration to function.
    - Simplicity: The system should be made as simple as possible, but no simpler.

- Amazon keeps its lips pretty tight about how S3 works, but according to Amazon, S3's design aims to provide scalability, high availability, and low latency at commodity costs.

- S3 stores arbitrary objects at up to 5GB in size, and each is accompanied by up to 2KB of metadata.
- Objects are organized by buckets.
- Each bucket is owned by an AWS account and the buckets are identified by a unique user assigned key.
- Buckets and objects are created, listed and retrieved using either a REST or SOAP interface.
- Objects can also be retrieved using the HTTP GET interface or via BitTorrent.
- An access control list restricts who can access the data in each bucket.
- Bucket names and keys are formulated so that they can be accessed using HTTP.
- Requests are authorized using an access control list associated with each bucket and object, for instance: http://s3.amazonaws.com/samplebucket/samplekey
- The Amazon AWS Authentication tools allow the bucket owner to create an authenticated URL with a set amount of time that the URL will be valid.
- Bucket items can also be accessed via a BitTorrent feed, enabling S3 to act as a seed for the client.
- Buckets can also be set up to save HTTP log information to another bucket.
- This information can be used for later data mining.

TWO MARK QUESTIONS

1. Illustrate architecture of a cloud is developed using three layers.



2. What is 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.
   - Market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand.

3. List the entities involved in the cloud platform.

   - Users or brokers and Request examiner
   - Pricing mechanism and VM Monitor mechanism
   - Accounting mechanism
   - Service Request Examiner and Admission Control mechanism
   - Dispatcher mechanism
   - Service Request Monitor mechanism

4. List the objectives of NIST Cloud Computing reference architecture

    - Illustrate and understand the various level of services
    - To provide technical reference
    - Categorize and compare services of cloud computing
    - Analysis of security, interoperatability and portability

5. Mention the major actors involved in NIST reference model.
    - Cloud consumer
    - Cloud provider
    - Cloud auditor
    - Cloud broker
    - Cloud carrier

6. Define service orchestration.

    - Service orchestration describes the automated arrangement, coordination and management of complex computing system.

7. Differentiate between Public cloud and Private Cloud.

    - A public cloud is one in which the cloud infrastructure and computing resources are made available to the general public over a public network.
    - A public cloud is owned by an organization selling cloud services, and serves a diverse pool of clients.
    - A private cloud gives a single Cloud Consumer's organization the exclusive access to and usage of the infrastructure and computational resources.
    - It may be managed either by the Cloud Consumer organization or by a third party, and may be hosted on the organization's premises (i.e. on-site private clouds) or outsourced to a hosting company (i.e. outsourced private clouds).

8. Tabulate the merits and demerits of Choosing a Community Cloud.

Merits	Demerits
• Ability to easily share and collaborate • Lower cost	• Not the right choice for every organization • Slow adoption to date

9. What is IaaS or HaaS?

- Infrastructure or Hardware-as-a-Service (IaaS/HaaS) solutions are the most popular and developed market segment of cloud computing.
- They deliver customizable infrastructure on demand.

10. What is PaaS?

- Platform-as-a-Service (PaaS) solutions provide a development and deployment platform for running applications in the cloud.
- They constitute the middleware on top of which applications are built.

11. Classify Pass Implementation

- PaaS implementation classified into three wide categories:
- PaaS-I, PaaS-II, and PaaS-III.

12. What is SaaS?

- Software-as-a-Service (SaaS) is a software delivery model that provides access to applications through the Internet as a Web-based service.
- It provides a means to free users from complex hardware and software management by offloading such tasks to third parties, which build applications accessible to multiple users through a Web browser.

13. What is SaaS 2.0?

- SaaS 2.0 is not a new technology but transforms the way in which SaaS is used.

- In particular, SaaS 2.0 is focused on providing a more robust infrastructure and application platforms driven by SLAs.
- SaaS 2.0 will focus on the rapid achievement of business objectives.

14. List the six architectural design challenges in cloud.

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

15. What is cloud storage?

- Cloud storage means storing the data with a cloud service provider rather than on a local system. The end user can access the data stored on the cloud using an Internet link.
- Cloud storage has a number of advantages over traditional data storage.
- If the users stored some data on a cloud, they can get at it from any location that has Internet access.

16. What is Storage-as-a-Service?

- The term Storage as a Service means that a third-party provider rents space on their storage to end users who lack the budget or capital budget to pay for it on their own.
- It is also ideal when technical personnel are not available or have inadequate knowledge to implement and maintain that storage infrastructure.

17. List the real time examples for cloud storage.

- Google Docs allows users to upload documents, spreadsheets, and presentations to Google’s data servers.

- Web email providers like Gmail, Hotmail, and Yahoo! Mail store email messages on their own servers.
- Flickr and Picasa host millions of digital photographs. YouTube hosts millions of user-uploaded video files.
- Hostmonster and GoDaddy store files and data for many client web sites.
- Facebook and MySpace are social networking sites and allow members to post pictures and other content.
- MediaMax and Strongspace offer storage space for any kind of digital data.

18. How to secure data in cloud storage?

- Most systems use a combination of following techniques:
    - Encryption
    - Authentication processes
    - Authorization practices

19. List the advantages of cloud storage.

- Storing data on the cloud is advantageous, because it allows you to protect your data in case there's a disaster.
- Having your data stored off-site can be the difference between closing your door for good or being down for a few days or weeks.
- Which storage vendor to go with can be a complex issue, and how the end user technology interacts with the cloud can be complex.

20. What is S3?

- The best-known cloud storage service is Amazon's Simple Storage Service (S3), which launched in 2006.
- Amazon S3 is designed to make web-scale computing easier for developers.
- Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the Web.
- It gives any developer access to the same highly scalable data storage infrastructure that Amazon uses to run its own global network of web sites.

21. What are the design requirements considers by Amazon to build S3?

- Scalable
- Reliable
- Fast
- Inexpensive
- Simple

22. What are the design principles considers by Amazon to meet S3 requirements?

- Decentralization
- Autonomy
- Local responsibility
- Controlled concurrency
- Failure toleration
- Controlled parallelism
- Symmetry
- Simplicity

23. How the data stored in S3?

- S3 stores arbitrary objects at up to 5GB in size, and each is accompanied by up to 2KB of metadata.
- Objects are organized by buckets.
- Each bucket is owned by an AWS account and the buckets are identified by a unique, user-assigned key.
- Buckets and objects are created, listed, and retrieved using either a REST-style or SOAP interface.

UNIT IV RESOURCE MANAGEMENT AND SECURITY IN CLOUD

Inter Cloud Resource Management –Resource Provisioning and Resource Provisioning Methods –Global Exchange of Cloud Resources –Security Overview –Cloud Security Challenges –Software-as-a-Service Security –Security Governance –Virtual Machine Security – IAM –Security Standards.

4.1 Inter Cloud Resource Management

Cloud application (SaaS)
Cloud Software environment (PaaS)
Cloud software infrastructure (IaaS, DaaS, CaaS)
Collocation cloud services (LaaS)
Network cloud services (NaaS)
Hardware or Virtualization cloud Services (HaaS)

Figure 4.1 A stack of six layers of cloud services

- Figure 4.1 shows six layers of cloud services, ranging from hardware, network, and collocation to infrastructure, platform and software applications.
- The cloud platform provides PaaS, which sits on top of the IaaS infrastructure. The top layer offers SaaS.
- 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 it 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 as well as network resources.

- Some authors say this layer provides Security as a Service (SaaS).

- 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.

- The three cloud models as viewed by different players.

- From the software vendor perspective, application performance on a given cloud platform is most important.

- From the provider perspective, cloud infrastructure performance is the primary concern.

- From the end users perspective, the quality of services, including security, is the most important.

- CRM offered the first SaaS on the cloud successfully.

- The approach is to widen market coverage by investigating customer behaviors and revealing opportunities by statistical analysis.

- SaaS tools also apply to distributed collaboration, financial and human resources management. These cloud services have been growing rapidly in recent years.

- PaaS is provided by Google, Salesforce.com, Facebook, and so on.

- IaaS is provided by Amazon, Windows Azure, RackRack, and so on.

- Based on the observations of some typical cloud computing instances, such as Google, Microsoft, and Yahoo!, the overall software stack structure of cloud computing software can be viewed as layers.

- Each layer has its own purpose and provides the interface for the upper layers just as the traditional soware stack does. However, the lower layers are not completely transparent to the upper layers.

- The platform for running cloud computing services can be either physical servers or virtual servers.

- By using VMs, the platform can be flexible; It means the running services are not bound to specific hardware platforms.

- The software layer on top of the platform is the layer for storing massive amounts of data.

- This layer acts like the file system in a traditional single machine. Other layers running on top of the file system are the layers for executing cloud computing applications.

- The next layers are the components in the software stack.

4.1.1 Runtime Support Services

- As in a cluster environment, there are also some runtime supporting services in the cloud computing environment.

- Cluster monitoring is used to collect the runtime status of the entire cluster.

- The scheduler queues the tasks submitted to the whole cluster and assigns the tasks to the processing nodes according to node availability.

- The distributed scheduler for the cloud application has special characteristics that can support cloud applications, such as scheduling the programs written in MapReduce style.

- The runtime support system keeps the cloud cluster working properly with high efficiency.

- Runtime support is software needed in browser initiated applications applied by thousands of cloud customers.

- The SaaS model provides the software applications as a service, rather than lifting users purchase the software.

- As a result, 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 customer data is stored in the cloud that is either vendor proprietary or a publicly hosted cloud supporting PaaS and IaaS.

4.2 Resource Provisioning

- 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.

- Over provisioning 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.

- Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures.

- Resource provisioning schemes also demand fast discovery of services and data in cloud computing infrastructures.

- In a virtualized cluster of servers, this demands efficient installation of VMs, live VM migration and fast recovery from failures.

- To deploy VMs, users treat them as physical hosts with customized operating systems for specific applications.

- For example, Amazon's EC2 uses Xen as the virtual machine monitor (VMM). The same VMM is used in IBM's Blue Cloud.

- In the EC2 platform, some predefined VM templates are also provided. Users can choose different kinds of VMs from the templates.

- IBM's Blue Cloud does not provide any VM templates.

4.3 Resource Provisioning Methods

- Figure 4.2 shows three cases of static cloud resource provisioning policies.

- In case (a), over provisioning with the peak load causes heavy resource waste (shaded area).

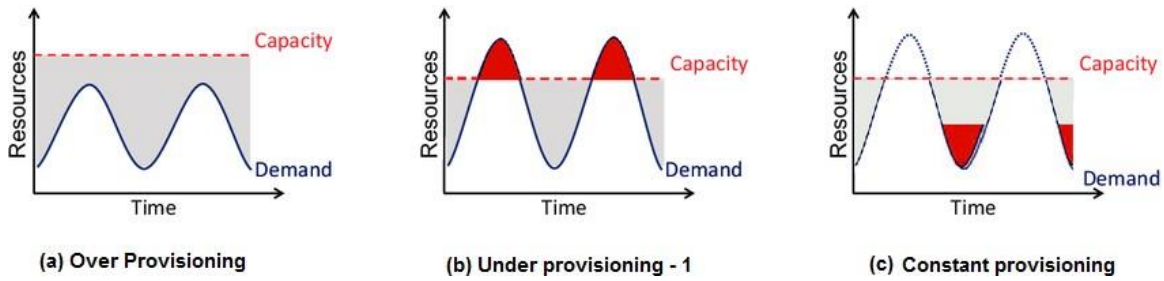


Figure 4.2 Three cases of resource provisioning

- In case (b), under provisioning (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.

- The user may give up the service by canceling the demand, resulting in reduced revenue for the provider.

- Both the user and provider may be losers in resource provisioning without elasticity.

- The demand-driven method provides static resources and has been used in grid computing for many years.

- The event-driven method is based on predicted workload by time.

- The popularity-driven method is based on Internet traffic monitored.

4.3.1 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

- In general, when a resource has surpassed a threshold for a certain amount of time, the scheme increases that resource based on demand.

- When a resource is below a threshold for a certain amount of time, that resource could be decreased accordingly.

- Amazon implements such an auto-scale feature in its EC2 platform. This method is easy to implement.

- The scheme does not work out right if the workload changes abruptly.

4.3.2 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.

- This scheme anticipates peak traffic before it happens.

- The method results in a minimal loss of QoS, if the event is predicted correctly.

- Otherwise, wasted resources are even greater due to events that do not follow a fixed pattern.

4.3.3 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.

4.4 Global Exchange of Cloud Resources

- In order to support a large number of application service consumers from around the world, cloud infrastructure providers (i.e., IaaS providers) have established data centers in multiple geographical locations to provide redundancy and ensure reliability in case of site failures.
- For example, Amazon has data centers in the United States (e.g., one on the East Coast and another on the West Coast) and Europe.
- However, currently Amazon expects its cloud customers (i.e., SaaS providers) to express a preference regarding where they want their application services to be hosted.
- Amazon does not provide seamless/automatic mechanisms for scaling its hosted services across multiple geographically distributed data centers.
- This approach has many shortcomings.
    - First, it is difficult for cloud customers to determine in advance the best location for hosting their services as they may not know the origin of consumers of their services.

- Second, SaaS providers may not be able to meet the QoS expectations of their service consumers originating from multiple geographical locations.

- Figure 4.3 shows the high-level components of the Melbourne group's proposed Inter Cloud architecture.

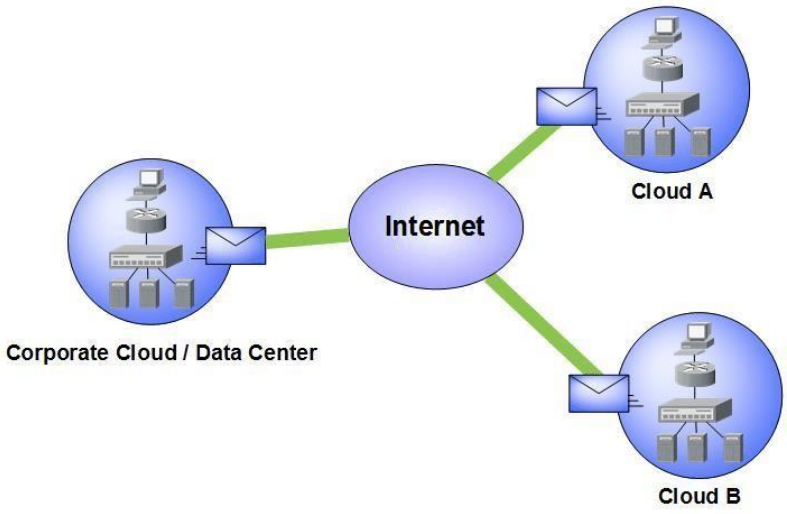


Figure 4.3 Inter cloud architecture

- In addition, no single cloud infrastructure provider will be able to establish its data centers at all possible locations throughout the world.

- As a result, cloud application service (SaaS) providers will have difficulty in meeting QoS expectations for all their consumers.

- The Cloudbus Project at the University of Melbourne has proposed InterCloud architecture supporting brokering and exchange of cloud resources for scaling applications across multiple clouds.

- By realizing InterCloud architectural principles in mechanisms in their offering,

    - Cloud providers will be able to dynamically expand or resize their provisioning capability based on sudden spikes in workload demands by leasing available computational and storage capabilities from other cloud service providers.

    - Operate as part of a market driven resource leasing federation, where application service providers such as Salesforce.com host their services based on negotiated SLA contracts driven by competitive market prices.
    - Deliver on-demand, reliable, cost-effective, and QoS-aware services based on virtualization technologies while ensuring high QoS standards and minimizing service costs.

- They need to be able to utilize market-based utility models as the basis for provisioning of virtualized software services and federated hardware infrastructure among users with heterogeneous applications.

- They consist of client brokering and coordinator services that support utility-driven federation of clouds:
    - Application scheduling
    - Resource allocation
    - Migration of workloads

- The architecture cohesively couples the administratively and topologically distributed storage and compute capabilities of clouds as part of a single resource leasing abstraction.

- The Cloud Exchange (CEx) acts as a market maker for bringing together service producers and consumers.

- It aggregates the infrastructure demands from application brokers and evaluates them against the available supply currently published by the cloud coordinators.

- It supports trading of cloud services based on competitive economic models such as commodity markets and auctions.

- An SLA specifies the details of the service to be provided in terms of metrics agreed upon by all parties, and incentives and penalties for meeting and violating the expectations, respectively.

- The availability of a banking system within the market ensures that financial transactions pertaining to SLAs between participants are carried out in a secure and dependable environment.

4.5 Security Overview

- Cloud service providers must learn from the managed service provider (MSP) model and ensure that their customer's applications and data are secure if they hope to retain their customer base and competitiveness.

- Today, enterprises are looking toward cloud computing horizons to expand their on-premises infrastructure, but most cannot afford the risk of compromising the security of their applications and data.

- For example, IDC recently conducted a survey1 (Figure 4.4) of 244 IT executives/CIOs and their line-of-business (LOB) colleagues to gauge their opinions and understand their companies' use of IT cloud services.

- Security ranked first as the greatest challenge or issue of cloud computing.

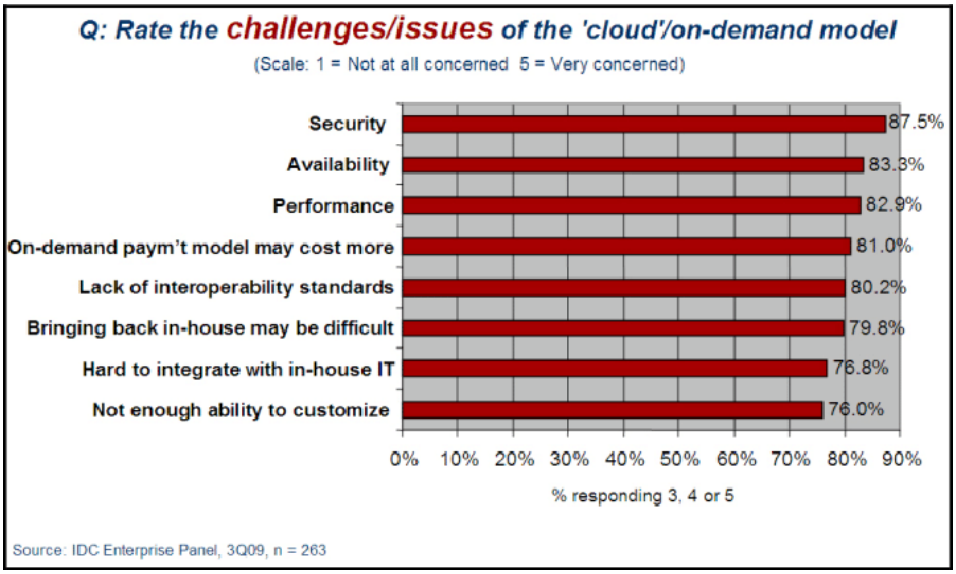


Figure 4.4 Results of IDC survey

- Moving critical applications and sensitive data to public and shared cloud environments is of great concern for those corporations that are moving beyond their data center's network perimeter defense.

- To alleviate these concerns, a cloud solution provider must ensure that customers will continue to have the same security and privacy controls over their applications and services.

- In addition, solution provider give evidence to customers that their organization and customers are secure and they can meet their service level agreements, and that they can prove compliance to auditors.

4.6 Cloud Security Challenges

- Although virtualization and cloud computing can help companies accomplish more by breaking the physical bonds between an IT infrastructure and its users, heightened security threats must be overcome in order to benefit fully from this new computing paradigm.

- Enterprise security is only as good as the least reliable partner, department and vendor.

- With the cloud model, the cloud consumer's loss control over physical security.

- In a public cloud, the consumers are sharing computing resources with other companies.

- In a shared pool outside the enterprise, users do not have any knowledge or control of where the resources run.

- Storage services provided by one cloud vendor may be incompatible with another vendor's services should you decide to move from one to the other.

- Ensuring the integrity of the data really means that it changes only in response to authorized transactions.

- The immature use of mash up technology (combinations of web services), which is fundamental to cloud applications, is inevitably going to cause unwitting security vulnerabilities in those applications.

- Since access to logs is required for Payment Card Industry Data Security Standard (PCI DSS) compliance and may be requested by auditors and regulators, security managers need to make sure to negotiate access to the provider's logs as part of any service agreement.

- Cloud applications undergo constant feature additions and users must keep up to date with application improvements to be sure they are protected.

- The speed at which applications will change in the cloud will affect both the SDLC and security.

- Security needs to move to the data level, so that enterprises can be sure their data is protected wherever it goes.

- Sensitive data is the domain of the enterprise, not the cloud computing provider.

- One of the key challenges in cloud computing is data level security.

- Most compliance standards do not envision compliance in a world of cloud computing.

- There is a huge body of standards that apply for IT security and compliance, governing most business interactions that will, over time, have to be translated to the cloud.

- SaaS makes the process of compliance more complicated, since it may be difficult for a customer to discern where its data resides on a network controlled by its SaaS provider, or a partner of that provider, which raises all sorts of compliance issues of data privacy, segregation, and security.

- Security managers will need to pay particular attention to systems that contain critical data such as corporate financial information or source code during the transition to server virtualization in production environments.

- Outsourcing means losing significant control over data, and while this is not a good idea from a security perspective, the business ease and financial savings will continue to increase the usage of these services.

- Security managers will need to work with their company's legal staff to ensure that appropriate contract terms are in place to protect corporate data and provide for acceptable service level agreements.

- Cloud based services will result in many mobile IT users accessing business data and services without traversing the corporate network.

- This will increase the need for enterprises to place security controls between mobile users and cloud based services.

- Although traditional data center security still applies in the cloud environment, physical segregation and hardware based security cannot protect against attacks between virtual machines on the same server.

- Administrative access is through the Internet rather than the controlled and restricted direct or on-premises connection that is adhered to in the traditional data center model.

- This increases risk and exposure and will require stringent monitoring for changes in system control and access control restriction.

- Proving the security state of a system and identifying the location of an insecure virtual machine will be challenging.

- The co-location of multiple virtual machines increases the attack surface and risk of virtual machine to virtual machine compromise.

- Localized virtual machines and physical servers use the same operating systems as well as enterprise and web applications in a cloud server environment, increasing the threat of an attacker or malware exploiting vulnerabilities in these systems and applications remotely.

- Virtual machines are vulnerable as they move between the private cloud and the public cloud.

- A fully or partially shared cloud environment is expected to have a greater attack surface and therefore can be considered to be at greater risk than a dedicated resources environment.

- Operating system and application files are on a shared physical infrastructure in a virtualized cloud environment and require system, file, and activity monitoring to provide confidence and auditable proof to enterprise customers that their resources have not been compromised or tampered with.

- In the cloud computing environment, the enterprise subscribes to cloud computing resources, and the responsibility for patching is the subscriber’s rather than the cloud computing vendors.

- The need for patch maintenance vigilance is imperative.

- Data is fluid in cloud computing and may reside in on-premises physical servers, on-premises virtual machines, or off-premises virtual machines running on cloud computing resources and this will require some rethinking on the part of auditors and practitioners alike.

- To establish zones of trust in the cloud, the virtual machines must be self-defending, effectively moving the perimeter to the virtual machine itself.

- Enterprise perimeter security (i.e., firewalls, demilitarized zones [DMZs], network segmentation, intrusion detection and prevention systems [IDS/IPS], monitoring tools, and the associated security policies) only controls the data that resides and transits behind the perimeter.

- In the cloud computing world, the cloud computing provider is in charge of customer data security and privacy.

4.7 Software-as-a-Service Security

- Cloud computing models of the future will likely combine the use of SaaS (and other XaaS's as appropriate), utility computing and Web 2.0 collaboration technologies to leverage the Internet to satisfy their customer needs.

- New business models being developed as a result of the move to cloud computing are creating not only new technologies and business operational processes but also new security requirements and challenges as described previously.

- As the most recent evolutionary step in the cloud service model (Figure 4.5), SaaS will likely remain the dominant cloud service model for the predictable future and the area where the most critical need for security practices and oversight will reside.

- The technology analyst and consulting firm Gartner lists seven security issues which one should discuss with a cloud computing vendor.

- Privileged user access inquires about who has specialized access to data and about the hiring and management of such administrators.

- Regulatory compliance makes sure that the vendor is willing to undergo external audits and/or security certifications.

- Data location does the provider allow for any control over the location of data.

- Data segregation makes encryption is available at all stages and that these encryption schemes were designed and tested by experienced professionals.

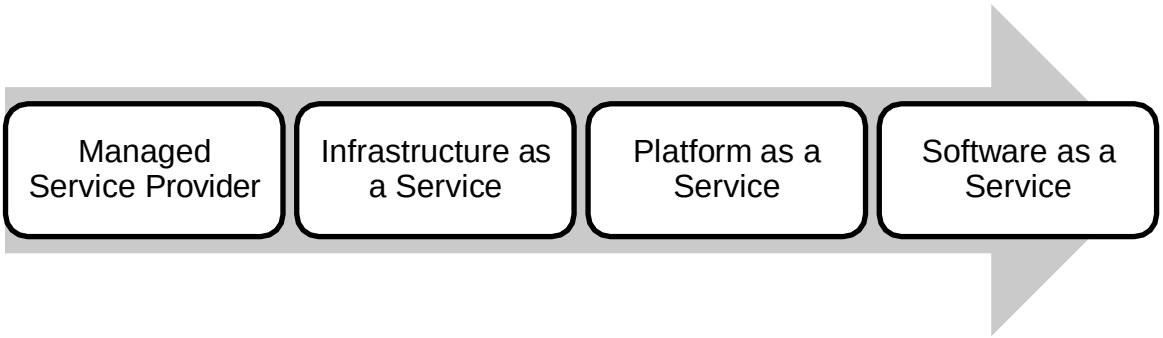


Figure 4.5 Evolution of cloud services

- Recovery is the way to find out what will happen to data in the case of a disaster. And also it covers the way to perform complete restoration.

- Investigative support does the vendor have the ability to investigate any inappropriate or illegal activity.

- Long-term viability focus on data if the company goes out of business and format and process behind the returned data.

- To address the security issues listed above, SaaS providers will need to incorporate and enhance security practices used by the managed service providers and develop new ones as the cloud computing environment evolves.

4.8 Security Governance

- A security steering committee should be developed whose objective is to focus on providing guidance about security initiatives and alignment with business and IT strategies.

- A charter for the security team is typically one of the first deliverables from the steering committee.

- This charter must clearly define the roles and responsibilities of the security team and other groups involved in performing information security functions.

- Lack of a formalized strategy can lead to an unsustainable operating model and security level as it evolves.

- In addition, lack of attention to security governance can result in key needs of the business not being met, including but not limited to, risk management, security monitoring, application security, and sales support.

- Lack of proper governance and management of duties can also result in potential security risks being left unaddressed and opportunities to improve the business being missed because the security team is not focused on the key security functions and activities that are critical to the business.

4.9 Virtual Machine Security

- In the cloud environment, physical servers are consolidated to multiple virtual machine instances on virtualized servers.

- Not only can data center security teams replicate typical security controls for the data center at large to secure the virtual machines, they can also advise their customers on how to prepare these machines for migration to a cloud environment when appropriate.

- Firewalls, intrusion detection and prevention, integrity monitoring and log inspection can all be deployed as software on virtual machines to increase protection as well as maintain compliance integrity of servers and applications as virtual resources move from on-premises to public cloud environments.

- By deploying this traditional line of defense to the virtual machine itself, the user can enable critical applications and data to be moved to the cloud securely.

- To facilitate the centralized management of a server firewall policy, the security software loaded onto a virtual machine should include a bidirectional stateful firewall that enables virtual machine isolation and location awareness, thereby enabling a tightened policy and the flexibility to move the virtual machine from on-premises to cloud resources.

- Integrity monitoring and log inspection software must be applied at the virtual machine level.

- This approach to virtual machine security, which connects the machine back to the mother ship, has some advantages in that the security software can be put into a single software agent that provides for consistent control and management throughout the cloud while integrating seamlessly back into existing security infrastructure investments, providing economies of scale, deployment, and cost savings for both the service provider and the enterprise.

4.10 IAM

- Identity and access management is a critical function for every organization and a fundamental expectation of SaaS customers is that the principle of least privilege is granted to their data.

- The principle of least privilege states that only the minimum access necessary to perform an operation should be granted, and that access should be granted only for the minimum amount of time necessary.

- However, business and IT groups will need and expect access to systems and applications.

- The advent of cloud services and services on demand is changing the identity management landscape.

- Most of the current identity management solutions are focused on the enterprise and typically are architected to work in a very controlled, static environment.

- User-centric identity management solutions such as federated identity management make some assumptions about the parties involved and their related services.

- In the cloud environment, where services are offered on demand and they can continuously evolve, aspects of current models such as trust assumptions, privacy implications, and operational aspects of authentication and authorization, will be challenged.

- Meeting these challenges will require a balancing act for SaaS providers as they evaluate new models and management processes for IAM to provide end-to-end trust and identity throughout the cloud and the enterprise.

- Another issue will be finding the right balance between usability and security. If a good balance is not achieved, both business and IT groups may be affected by barriers to completing their support and maintenance activities efficiently.

4.11 Security Standards

- Security standards define the processes, procedures, and practices necessary for implementing a security program.

- These standards also apply to cloud related IT activities and include specific steps that should be taken to ensure a secure environment is maintained that provides privacy and security of confidential information in a cloud environment.

- Security standards are based on a set of key principles intended to protect this type of trusted environment.

- Messaging standards, especially for security in the cloud, must also include nearly all the same considerations as any other IT security endeavor.

- Security (SAML OAuth, OpenID, SSL/TLS) A basic philosophy of security is to have layers of defense, a concept known as defense in depth.

- This means having overlapping systems designed to provide security even if one system fails. An example is a firewall working in conjunction with an intrusion-detection system (IDS).

- Defense in depth provides security because there is no single point of failure and no single entry vector at which an attack can occur.

- For this reason, a choice between implementing network security in the middle part of a network (i.e., in the cloud) or at the endpoints is a false dichotomy.

- No single security system is a solution by itself, so it is far better to secure all systems.

- This type of layered security is precisely what we are seeing develop in cloud computing.

- Traditionally, security was implemented at the endpoints, where the user controlled access.

- An organization had no choice except to put firewalls, IDSs, and antivirus software inside its own network.

- Today, with the advent of managed security services offered by cloud providers, additional security can be provided inside the cloud.

4.11.1 Security Assertion Markup Language (SAML)

- SAML is an XML-based standard for communicating authentication, authorization and attribute information among online partners.

- It allows businesses to securely send assertions between partner organizations regarding the identity and entitlements of a principal.

- The Organization for the Advancement of Structured Information Standards (OASIS) Security Services Technical Committee is in charge of defining, enhancing and maintaining the SAML specifications.

- SAML is built on a number of existing standards, namely, SOAP, HTTP and XML. SAML relies on HTTP as its communications protocol and specifies the use of SOAP (currently, version 1.1).

- Most SAML transactions are expressed in a standardized form of XML.

- SAML assertions and protocols are specified using XML schema.

- Both SAML 1.1 and SAML 2.0 use digital signatures (based on the XML Signature standard) for authentication and message integrity.

- XML encryption is supported in SAML 2.0, though SAML 1.1 does not have encryption capabilities.

- SAML defines XML based assertions and protocols, bindings and profiles.

- The term SAML Core refers to the general syntax and semantics of SAML assertions as well as the protocol used to request and transmit those assertions from one system entity to another.

- SAML protocol refers to what is transmitted, not how it is transmitted.

- A SAML binding determines how SAML requests and responses map to standard messaging protocols. An important (synchronous) binding is the SAML SOAP binding.

- SAML standardizes queries for, and responses that contain, user authentication, entitlements and attribute information in an XML format.

- This format can then be used to request security information about a principal from a SAML authority.

- A SAML authority, sometimes called the asserting party. It is a platform or application that can relay security information.

- The relying party (or assertion consumer or requesting party) is a partner site that receives the security information.

- The exchanged information deals with a subject’s authentication status, access authorization, and attribute information.

- A subject is an entity in a particular domain.

- A person identified by an email address is a subject, as might be a printer.

- SAML assertions are usually transferred from identity providers to service providers.

- Assertions contain statements that service providers use to make access control decisions.

- Three types of statements are provided by SAML:

    - Authentication statements
    - Attribute statements
    - Authorization decision statements

- SAML assertions contain a packet of security information in this form:

<saml: Asssertion A>

<Authentication>

…

</Authentication>

<Attribute>

…

</Attribute>

<Authentication>

…

</Authentication>

</saml: Asssertion A>

- The assertion shown above is interpreted as follows:
  Assertion A, issued at time T by issuer I, regarding subject S, provided conditions C are valid.

- Authentication statements assert to a service provider that the principal did indeed authenticate with an identity provider at a particular time using a particular method of authentication.

- Other information about the authenticated principal (called the authentication context) may be disclosed in an authentication statement.

- An attribute statement asserts that a subject is associated with certain attributes.

- An attribute is simply a name-value pair.

- An authorization decision statement asserts that a subject is permitted to perform action A on resource R given evidence E.

- A SAML protocol describes how certain SAML elements (including assertions) are packaged within SAML request and response elements

- Generally, a SAML protocol is a simple request–response protocol.

- The most important type of SAML protocol request is a query.

- A service provider makes a query directly to an identity provider over a secure back channel. For this reason, query messages are typically bound to SOAP.

- Corresponding to the three types of statements, there are three types of SAML queries:
    - Authentication query
    - Attribute query
    - Authorization decision query.

- Of these, the attribute query is perhaps most important. The result of an attribute query is a SAML response containing an assertion, which itself contains an attribute statement.

4.11.2 Open Authentication (OAuth)

- OAuth is an open protocol, initiated by Blaine Cook and Chris Messina, to allow secure API authorization in a simple, standardized method for various types of web applications.

- Cook and Messina had concluded that there were no open standards for API access delegation.

- The OAuth discussion group was created in April 2007, for the small group of implementers to write the draft proposal for an open protocol.

- DeWitt Clinton of Google learned of the OAuth project and expressed interest in supporting the effort.

- In July 2007, the team drafted an initial specification and it was released in October of the same year.

- OAuth is a method for publishing and interacting with protected data.

- For developers, OAuth provides users access to their data while protecting account credentials.

- OAuth allows users to grant access to their information, which is shared by the service provider and consumers without sharing all of their identity.

- The Core designation is used to stress that this is the baseline, and other extensions and protocols can build on it.

- By design, OAuth Core 1.0 does not provide many desired features (e.g., automated discovery of endpoints, language support, support for XML-RPC and SOAP, standard definition of resource access, OpenID integration, signing algorithms, etc.).

- This intentional lack of feature support is viewed by the authors as a significant benefit.

- The Core deals with fundamental aspects of the protocol, namely, to establish a mechanism for exchanging a user name and password for a token with defined rights and to provide tools to protect the token.

- It is important to understand that security and privacy are not guaranteed by the protocol.

- In fact, OAuth by itself provides no privacy at all and depends on other protocols such as SSL to accomplish that.

- OAuth can be implemented in a secure manner.

- In fact, the specification includes substantial security considerations that must be taken into account when working with sensitive data.

- With Oauth, sites use tokens coupled with shared secrets to access resources.

- Secrets, just like passwords, must be protected.

4.11.3 OpenID

- OpenID is an open, decentralized standard for user authentication and access control that allows users to log onto many services using the same digital identity.

- It is a single-sign-on (SSO) method of access control. As such, it replaces the common log-in process (i.e., a log-in name and a password) by allowing users to log in once and gain access to resources across participating systems.

- The original OpenID authentication protocol was developed in May 2005 by Brad Fitzpatrick, creator of the popular community web site LiveJournal.

- In late June 2005, discussions began between OpenID developers and other developers from an enterprise software company named NetMesh.

- These discussions led to further collaboration on interoperability between OpenID and NetMesh's similar Light-Weight Identity (LID) protocol.

- The direct result of the collaboration was the Yadis discovery protocol, which was announced on October 24, 2005.

- The Yadis specification provides a general-purpose identifier for a person and any other entity, which can be used with a variety of services.

- It provides syntax for a resource description document identifying services available using that identifier and an interpretation of the elements of that document.

- Yadis discovery protocol is used for obtaining a resource description document, given that identifier.

- Together these enable coexistence and interoperability of a rich variety of services using a single identifier.

- The identifier uses a standard syntax and a well established namespace and requires no additional namespace administration infrastructure.

- An OpenID is in the form of a unique URL and is authenticated by the entity hosting the OpenID URL.

- The OpenID protocol does not rely on a central authority to authenticate a user's identity.

- Neither the OpenID protocol nor any web sites requiring identification can mandate that a specific type of authentication be used; nonstandard forms of authentication such as smart cards, biometrics, or ordinary passwords are allowed.

- A typical scenario for using OpenID might be something like this:

    - A user visits a web site that displays an OpenID log in form
    - Unlike a typical log in form, which has fields for user name and password, the OpenID log in form has only one field for the OpenID identifier (which is an OpenID URL).
    - This form is connected to an implementation of an OpenID client library.
    - A user will have previously registered an OpenID identifier with an OpenID identity provider.
    - The user types this OpenID identifier into the OpenID log-in form.
    - The relying party then requests the web page located at that URL and reads an HTML link tag to discover the identity provider service URL.

- With OpenID 2.0, the client discovers the identity provider service URL by requesting the XRDS document (also called the Yadis document) with the content type application/xrds+xml, which may be available at the target URL but is always available for a target XRI.

- There are two modes by which the relying party can communicate with the identity provider: checkid_immediate and checkid_setup.

- In checkid_immediate, the relying party requests that the provider not interact with the user. All communication is relayed through the user's browser without explicitly notifying the user.

- In checkid_setup, the user communicates with the provider server directly using the same web browser as is used to access the relying party site.

- OpenID does not provide its own authentication methods, but if an identity provider uses strong authentication, OpenID can be used for secure transactions.

- SSL/TLS Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographically secure protocols designed to provide security and data integrity for communications over TCP/IP.

- TLS and SSL encrypt the segments of network connections at the transport layer.

- Several versions of the protocols are in general use in web browsers, email, instant messaging and Voice-over-IP (VoIP).

- TLS is an IETF standard protocol which was last updated in RFC 5246.

- The TLS protocol allows client/server applications to communicate across a network in a way specifically designed to prevent eavesdropping, tampering, and message forgery.

- TLS provides endpoint authentication and data confidentiality by using cryptography.

- TLS authentication is one way in which the server is authenticated, because the client already knows the server's identity. In this case, the client remains unauthenticated.

- TLS also supports a more secure bilateral connection mode whereby both ends of the connection can be assured that they are communicating with whom they believe they are connected.

- This is known as mutual authentication.

- Mutual authentication requires the TLS client side to also maintain a certificate.

- TLS involves three basic phases:

    - Peer negotiation for algorithm support
    - Key exchange and authentication
    - Symmetric cipher encryption and message authentication

TWO MARK QUESTIONS

1. List the runtime supporting services in the cloud computing environment.

    - Cluster monitoring is used to collect the runtime status of the entire cluster.
    - The scheduler queues the tasks submitted to the whole cluster and assigns the tasks to the processing nodes according to node availability.
    - The distributed scheduler for the cloud application has special characteristics that can support cloud applications, such as scheduling the programs written in MapReduce style.

2. Why inter cloud resource management requires runtime support system?

    - The runtime support system keeps the cloud cluster working properly with high efficiency.
    - Runtime support is software needed in browser-initiated applications applied by thousands of cloud customers.

3. Differentiate between over provisioning and under provisioning.

    - Under provisioning of resources will lead to broken SLAs and penalties.
    - Over provisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider.
    - over provisioning with the peak load causes heavy resource waste (shaded area).
    - under provisioning (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.

4. List the various resource provisioning methods.

    - demand-driven resource provisioning
    - Event-Driven Resource Provisioning
    - Popularity-Driven Resource Provisioning
    - Global Exchange of Cloud Resources

5. What is 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

6. Write short notes on cloud security.

    - Cloud service providers must learn from the managed service provider (MSP) model and ensure that their customer's applications and data are secure if they hope to retain their customer base and competitiveness.
    - Security ranked first as the greatest challenge or issue of cloud computing.

7. List the challenges in cloud security.

- Enterprise security is only as good as the least reliable partner, department, or vendor.
- With the cloud model, users lose control over physical security.
- In a public cloud, the users are sharing computing resources with other companies.
- In a shared pool outside the enterprise, users don't have any knowledge or control of where the resources run.
- Storage services provided by one cloud vendor may be incompatible with another vendor's services should you decide to move from one to the other.
- Ensuring the integrity of the data really means that it changes only in response to authorized transactions.

8. List the seven security issues with respect to cloud computing vendor.

- Privileged user access
- Regulatory compliance
- Data location
- Data segregation
- Recovery
- Investigative support
- Long-term viability

9. What is the purpose of security governance?

- A security steering committee should be developed whose objective is to focus on providing guidance about security initiatives and alignment with business and IT strategies.
- A charter for the security team is typically one of the first deliverables from the steering committee.

10. How to perform virtual machine security?

- Firewalls, intrusion detection and prevention, integrity monitoring, and log inspection can all be deployed as software on virtual machines to increase protection and maintain compliance integrity of servers and applications as virtual resources move from on-premises to public cloud environments.
- Integrity monitoring and log inspection software must be applied at the virtual machine level.

11. Define IAM.

- Identity and access management is a critical function for every organization, and a fundamental expectation of SaaS customers is that the principle of least privilege is granted to their data.

12. Why cloud requires security standards?

- Security standards define the processes, procedures, and practices necessary for implementing a security program.
- These standards also apply to cloud related IT activities and include specific steps that should be taken to ensure a secure environment is maintained that provides privacy and security of confidential information in a cloud environment.

13. What is SAML?

- Security Assertion Markup Language (SAML) is an XML-based standard for communicating authentication, authorization, and attribute information among online partners.
- It allows businesses to securely send assertions between partner organizations regarding the identity and entitlements of a principal.

14. List the types of statements are provided by SAML.

- Authentication statements
- Attribute statements
- Authorization decision statements

15. Describe about SAML protocol.

- A SAML protocol describes how certain SAML elements (including assertions) are packaged within SAML request and response elements
- SAML protocol is a simple request–response protocol.
- The most important type of SAML protocol request is a query.

16. List the types of SAML queries.

- Authentication query
- Attribute query
- Authorization decision query.

17. What is OAuth?

- OAuth (Open authentication) is an open protocol, initiated by Blaine Cook and Chris Messina, to allow secure API authorization in a simple, standardized method for various types of web applications.
- OAuth is a method for publishing and interacting with protected data.
- OAuth allows users to grant access to their information, which is shared by the service provider and consumers without sharing all of their identity.

18. What is the purpose of OpenID?

- OpenID is an open, decentralized standard for user authentication and access control that allows users to log onto many services using the same digital identity.
- It is a single-sign-on (SSO) method of access control.

- An OpenID is in the form of a unique URL and is authenticated by the entity hosting the OpenID URL.

19. Why cloud environment need SSL/TLS?

- SSL/TLS Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographically secure protocols designed to provide security and data integrity for communications over TCP/IP.
- TLS and SSL encrypt the segments of network connections at the transport layer.

20. What is mutual authentication?

- TLS also supports a more secure bilateral connection mode whereby both ends of the connection can be assured that they are communicating with whom they believe they are connected. This is known as mutual authentication.
- Mutual authentication requires the TLS client side to also maintain a certificate.

UNIT V CLOUD TECHNOLOGIES AND ADVANCEMENTS

Hadoop – MapReduce – Virtual Box -- Google App Engine – Programming Environment for Google App Engine – OpenStack – Federation in the Cloud – Four Levels of Federation – Federated Services and Applications – Future of Federation.

5.1 Hadoop

- 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:

    - MapReduce engine
    - HDFS

- The MapReduce engine is the computation engine running on top of HDFS as its data storage manager.

- 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 fixed-size 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.

- In such systems, the namespace is the area maintaining the metadata and metadata refers to all the information stored by a file system that is needed for overall management of all files.

- For example, NameNode in the metadata stores all information regarding the location of input splits/blocks in all DataNodes.

- Each DataNode, usually one per node in a cluster, manages the storage attached to the node. Each DataNode is responsible for storing and retrieving its file blocks.

- 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.

- One of the main aspects of HDFS is its fault tolerance characteristic. Since Hadoop is designed to be deployed on low-cost hardware by default, a hardware failure in this system is considered to be common rather than an exception.

- Hadoop considers the following issues to fulfill reliability requirements of the file system

    - Block replication: To reliably store data in HDFS, file blocks are replicated in this system. The replication factor is set by the user and is three by default.

- Replica placement: The placement of replicas is another factor to fulfill the desired fault tolerance in HDFS.
- Heartbeat and Block report messages: Heartbeats and Block reports are periodic messages sent to the NameNode by each DataNode in a cluster.
- 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.
    - Keeping large amounts of data sequentially within a block 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

    - 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.
    - 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.
- The steamer then stores the block in the first allocated DataNode.
- Afterward, the block is forwarded to the second DataNode by the first DataNode.
- The process continues until all allocated DataNodes receive a replica of the first block from the previous DataNode.
- Once this replication process is finalized, the same process starts for the second block and continues until all blocks of the file are stored and replicated on the file system.

5.2 MapReduce

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

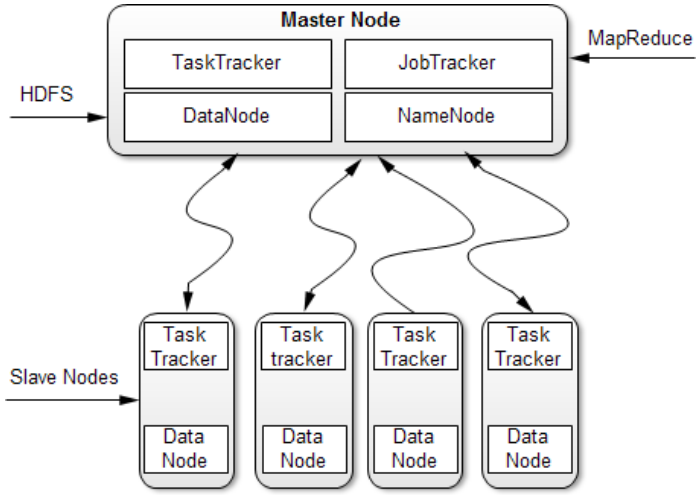


Figure 5.1 HDFS and MapReduce Architecture

- Figure 5.1 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).

- 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.

- For example, a TaskTracker node with N CPUs, each supporting M threads, has M * N simultaneous execution slots.

- It is worth noting that each data block is processed by one map task running on a single slot.

- Therefore, there is a one to one correspondence between map tasks in a TaskTracker and data blocks in the respective DataNode.

Running a Job in Hadoop

- Three components contribute in running a job in this system:
    - User node
    - JobTracker
    - 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 5.2 depicts the data flow of running a MapReduce job in Hadoop.

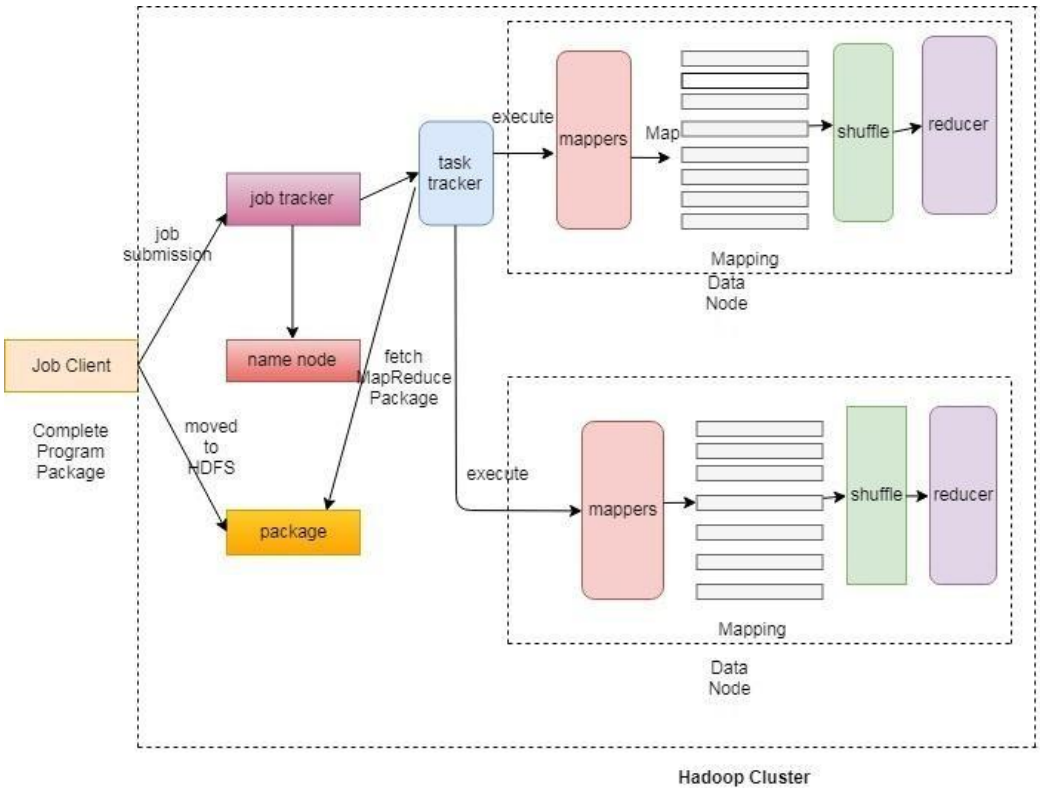


Figure 5.2 Data flow in Hadoop

- 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.
- 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.

- 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.
- 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.

5.3 Virtual Box

- Oracle VM VirtualBox is a cross platform virtualization application.

- For one thing, it installs on the existing Intel or AMD-based computers, whether they are running Windows, Mac OS X, Linux, or Oracle Solaris operating systems (OSes).

- Secondly, it extends the capabilities of existing computer so that it can run multiple OSes, inside multiple virtual machines, at the same time.

- As an example, the end user can run Windows and Linux on your Mac, run Windows Server 2016 on your Linux server, run Linux on your Windows PC, and so on, all alongside the existing applications.

- The user can install and run as many virtual machines.

- The only practical limits are disk space and memory.

- Oracle VM VirtualBox is deceptively simple yet also very powerful.

- It can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments.

- Virtual Box is created by Innotek and it was acquired by Sun Microsystems. In 2010, Virtual Box was acquired by Oracle.

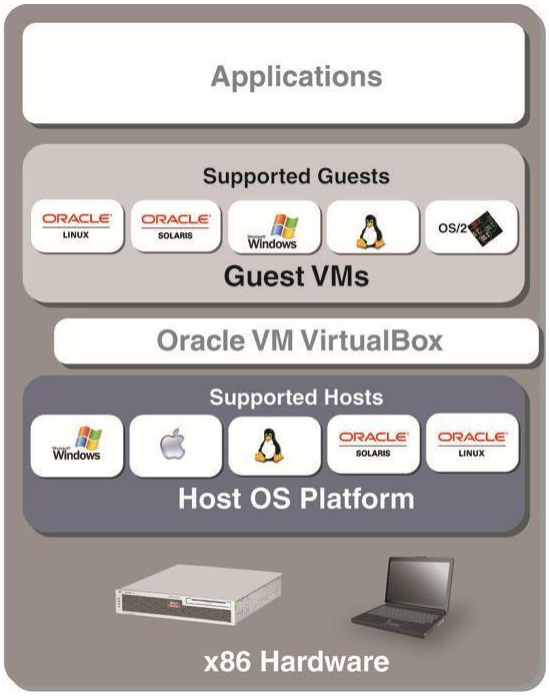


Figure 5.3 architecture of Virtual Box

- Virtual Box supported in Windows, macOS. Linux, Solaris and Open Solaris.

- Figure 5.3 depicts the architecture of Virtual Box

- The user can independently configure each VM and run it under a choice of software-based virtualization or hardware assisted virtualization if the underlying host hardware supports this.

- The host OS and guest OSs and applications can communicate with each other through a number of mechanisms including a common clipboard and a virtualized network facility.

- Guest VMs can also directly communicate with each other if configured to do so.

- The software based virtualization was dropped starting with VirtualBox 6.1. In earlier versions the absence of hardware assisted virtualization, VirtualBox adopts a standard software-based virtualization approach.

- This mode supports 32 bit guest OSs which run in rings 0 and 3 of the Intel ring architecture.

    - The system reconfigures the guest OS code, which would normally run in ring 0, to execute in ring 1 on the host hardware.
    - Because this code contains many privileged instructions which cannot run natively in ring 1, VirtualBox employs a Code Scanning and Analysis Manager (CSAM) to scan the ring 0 code recursively before its first execution to identify problematic instructions and then calls the Patch Manager (PATM) to perform in-situ patching.
    - This replaces the instruction with a jump to a VM-safe equivalent compiled code fragment in hypervisor memory.

    - The guest user mode code, running in ring 3, generally runs directly on the host hardware in ring 3.

- In both cases, VirtualBox uses CSAM and PATM to inspect and patch the offending instructions whenever a fault occurs.

- VirtualBox also contains a dynamic recompiler, based on QEMU to recompile any real mode or protected mode code entirely.

- Hardware assisted virtualization is starting with version 6.1, VirtualBox only supports.

- VirtualBox supports both Intel VT-X and AMD-V hardware assisted virtualization.

- Making use of these facilities, VirtualBox can run each guest VM in its own separate address-space.

- The guest OS ring 0 code runs on the host at ring 0 in VMX non-root mode rather than in ring 1.

- Until then, VirtualBox specifically supported some guests (including 64 bit guests, SMP guests and certain proprietary OSs) only on hosts with hardware-assisted virtualization

- The system emulates hard disks in one of three disk image formats:

    - VDI: This format is the VirtualBox-specific VirtualBox Disk Image and stores data in files bearing a ".vdi" .
    - VMDK: This open format is used by VMware products and stores data in one or more files bearing ".vmdk" filename extensions. A single virtual hard disk may span several files.
    - VHD: This format is used by Windows Virtual PC and Hyper-V and it is the native virtual disk format of the Microsoft Windows operating system. Data in this format are stored in a single file bearing the ".vhd" filename extension.

- A VirtualBox virtual machine can, therefore, use disks previously created in VMware or Microsoft Virtual PC, as well as its own native format.

- VirtualBox can also connect to iSCSI targets and to raw partitions on the host, using either as virtual hard disks.

- VirtualBox has supported Open Virtualization Format (OVF).

- By default, VirtualBox provides graphics support through a custom virtual graphics-card

- For an Ethernet network adapter, VirtualBox virtualizes these Network Interface Cards.

    - AMD PCnet PCI II
    - AMD PCnet-Fast III
    - Intel Pro/1000 MT Desktop
    - Intel Pro/1000 MT Server
    - Intel Pro/1000 T Server
    - Paravirtualized network adapter

- For a sound card, VirtualBox virtualizes Intel HD Audio.

- A USB controller is emulated so that any USB devices attached to the host can be seen in the guest.

- Oracle VM VirtualBox was designed to be modular and flexible.

- When the Oracle VM VirtualBox graphical user interface (GUI) is opened and a VM is started, at least the following three processes are running:

    - VBoxSVC is Oracle VM VirtualBox service process which always runs in the background. This process is started automatically by the first Oracle VM VirtualBox client process and exits a short time after the last client exits.

- The first Oracle VM VirtualBox service can be the GUI, VBoxManage, VBoxHeadless, the web service amongst others.

- The service is responsible for bookkeeping, maintaining the state of all VMs, and for providing communication between Oracle VM VirtualBox components.

- Oracle VM VirtualBox comes with comprehensive support for third-party developers.

- The Main API of Oracle VM VirtualBox exposes the entire feature set of the virtualization engine.

- The Main API is made available to C++ clients through COM on Windows hosts or XPCOM on other hosts. Bridges also exist for SOAP, Java and Python.

5.4 Google App Engine

- 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 platform is based on its search engine expertise.

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

- For example, 200 Google data centers are used at one time for a number of cloud applications.

- Data items are stored in text, images, and video and are replicated to tolerate faults or failures.

- Google's App Engine (GAE) which offers a PaaS platform supporting various cloud and web applications.

- 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.

5.4.1 GAE Architecture

- Figure 5.4 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.

- 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

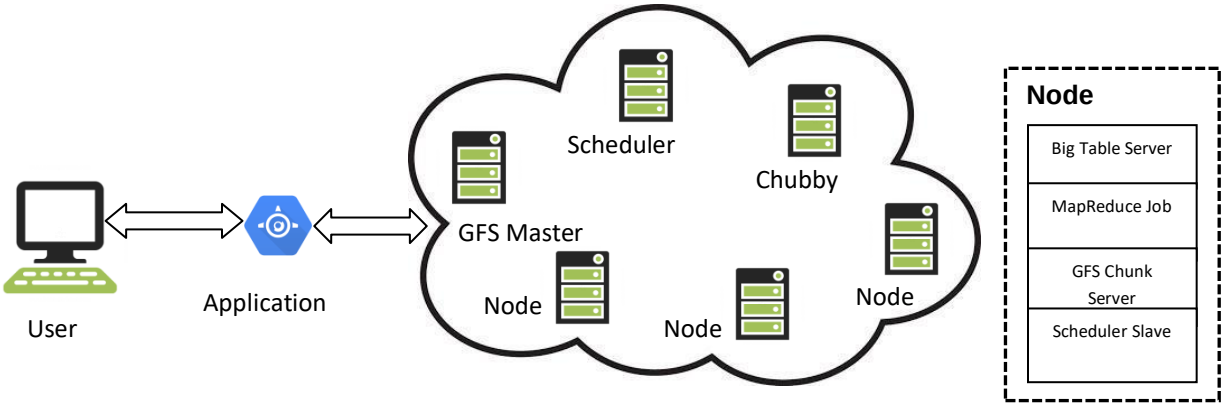


Figure 5.4 Google cloud platform

- Google is one of the larger cloud application providers, although it is 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 semistructural data.

- With these building blocks, Google has built many cloud applications.

- Figure 5.4 shows the overall architecture of the Google cloud infrastructure.

- A typical cluster configuration can run the Google File System, MapReduce jobs and BigTable servers for structure data.

- Extra services such as Chubby for distributed locks can also run in the clusters.

- GAE runs the user program on Google's infrastructure. As it is a platform running third-party programs, application developers now do not need to worry about the maintenance of servers.

- GAE can be thought of as the combination of several software components.

- The frontend is an application framework which is similar to other web application frameworks such as ASP, J2EE and JSP.

- At the time of this writing, GAE supports Python and Java programming environments. The applications can run similar to web application containers.

- The frontend can be used as the dynamic web serving infrastructure which can provide the full support of common technologies.

5.4.2 Functional Modules of GAE

- The GAE platform comprises the following five major components.

- The GAE is not an infrastructure platform, but rather an application development platform for users.

    - The datastore offers object-oriented, distributed, structured data storage services based on BigTable techniques. The datastore secures data management operations.
    - The application runtime environment offers a platform for scalable web programming and execution. It supports two development languages: Python and Java.
    - The software development kit (SDK) is used for local application development. The SDK allows users to execute test runs of local applications and upload application code.
    - The administration console is used for easy management of user application development cycles, instead of for physical resource management.
    - The GAE web service infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources by GAE.

- Google offers essentially free GAE services to all Gmail account owners.

- The user can register for a GAE account or use your Gmail account name to sign up for the service.

- The service is free within a quota.

- If the user exceeds the quota, the page instructs how to pay for the service. Then the user can download the SDK and read the Python or Java guide to get started.

- Note that GAE only accepts Python, Ruby and Java programming languages.

- The platform does not provide any IaaS services, unlike Amazon, which offers IaaS and PaaS.

- This model allows the user to deploy user-built applications on top of the cloud infrastructure that are built using the programming languages and software tools supported by the provider (e.g., Java, Python).

- Azure does this similarly for .NET. The user does not manage the underlying cloud infrastructure.

- The cloud provider facilitates support of application development, testing, and operation support on a well-defined service platform.

5.4.3 GAE Applications

- Best-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. This can eliminate the tedious work of building customized user management components in web applications.

5.5 Programming Environment for Google App Engine

- Several web resources (e.g., http://code.google.com/appengine/) and specific books and articles discuss how to program GAE.

- Figure 5.5 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. Developers can use this, or any other language using a JVM based 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.

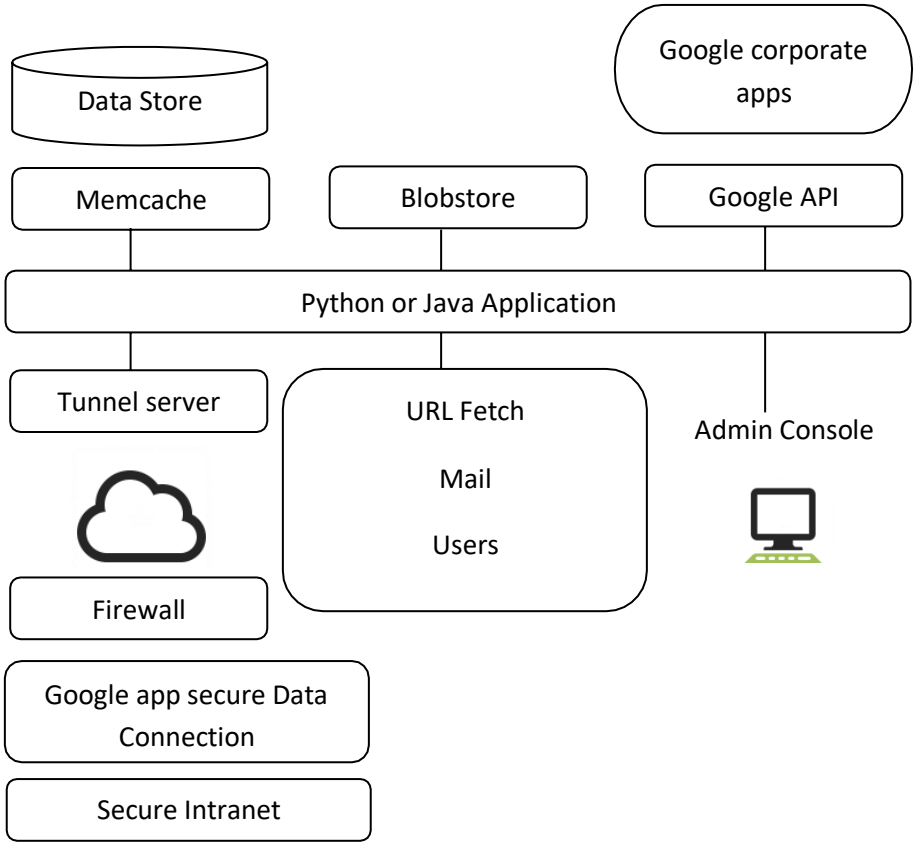


Figure 5.5 Programming Environment of Google AppEngine

- 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.

- An update of an entity occurs in a transaction that is retried a fixed number of times if other processes are trying to update the same entity simultaneously.

- The user application can execute multiple data store operations in a single transaction which either all succeed or all fail together.

- The data store implements transactions across its distributed network using entity groups.

- A transaction manipulates entities within a single group.

- Entities of the same group are stored together for efficient execution of transactions.

- The user GAE application can assign entities to groups when the entities are created.

- 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 high-speed 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.

- A GAE application is configured to consume resources up to certain limits or quotas. With quotas, GAE ensures that your application would not exceed your budget and that other applications running on GAE would not impact the performance of your app.

- In particular, GAE use is free up to certain quotas.

- 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.

- 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.

- 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.

- BigTable was designed to provide a service for storing and retrieving structured and semi structured data.

- BigTable applications include storage of web pages, per-user data, and geographic locations.

- The scale of such data is incredibly large. There will be billions of URLs, and each URL can have many versions, with an average page size of about 20 KB per version.

- The user scale is also huge.

- There are hundreds of millions of users and there will be thousands of queries per second.

- The same scale occurs in the geographic data, which might consume more than 100 TB of disk space.

- It is not possible to solve such a large scale of structured or semi structured data using a commercial database system.

- This is one reason to rebuild the data management system and the resultant system can be applied across many projects for a low incremental cost.

- The other motivation for rebuilding the data management system is performance.

- Low level storage optimizations help increase performance significantly which is much harder to do when running on top of a traditional database layer.

- 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.
    - The application may need to examine data changes over time.

- Thus, BigTable can be viewed as a distributed multilevel map. It provides a fault tolerant and persistent database as in a storage service.

- The BigTable system is scalable, which means the system has thousands of servers, terabytes of in-memory data, peta bytes of disk based data, millions of reads/writes per second and efficient scans.

- BigTable is a self managing system (i.e., servers can be added/removed dynamically and it features automatic load balancing).

- Chubby, Google's Distributed Lock Service Chubby is intended to provide a coarse-grained locking service.

- It can store small files inside Chubby storage which provides a simple namespace as a file system tree.

- The files stored in Chubby are quite small compared to the huge files in GFS.

5.6 OpenStack

- The OpenStack project is an open source cloud computing platform for all types of clouds, which aims to be simple to implement, massively scalable and feature rich.

- Developers and cloud computing technologists from around the world create the OpenStack project.

- OpenStack provides an Infrastructure as a Service (IaaS) solution through a set of interrelated services.

- Each service offers an application programming interface (API) that facilitates this integration.

- Depending on their needs, administrator can install some or all services.

- OpenStack began in 2010 as a joint project of Rackspace Hosting and NASA.

- As of 2012, it is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2013 to promote OpenStack software and its community.

- Now, More than 500 companies have joined the project

- The OpenStack system consists of several key services that are separately installed.

- These services work together depending on your cloud needs and include the Compute, Identity, Networking, Image, Block Storage, Object Storage, Telemetry, Orchestration, and Database services.

- The administrator can install any of these projects separately and configure them standalone or as connected entities.

- Figure 5.6 shows the relationships among the OpenStack services:

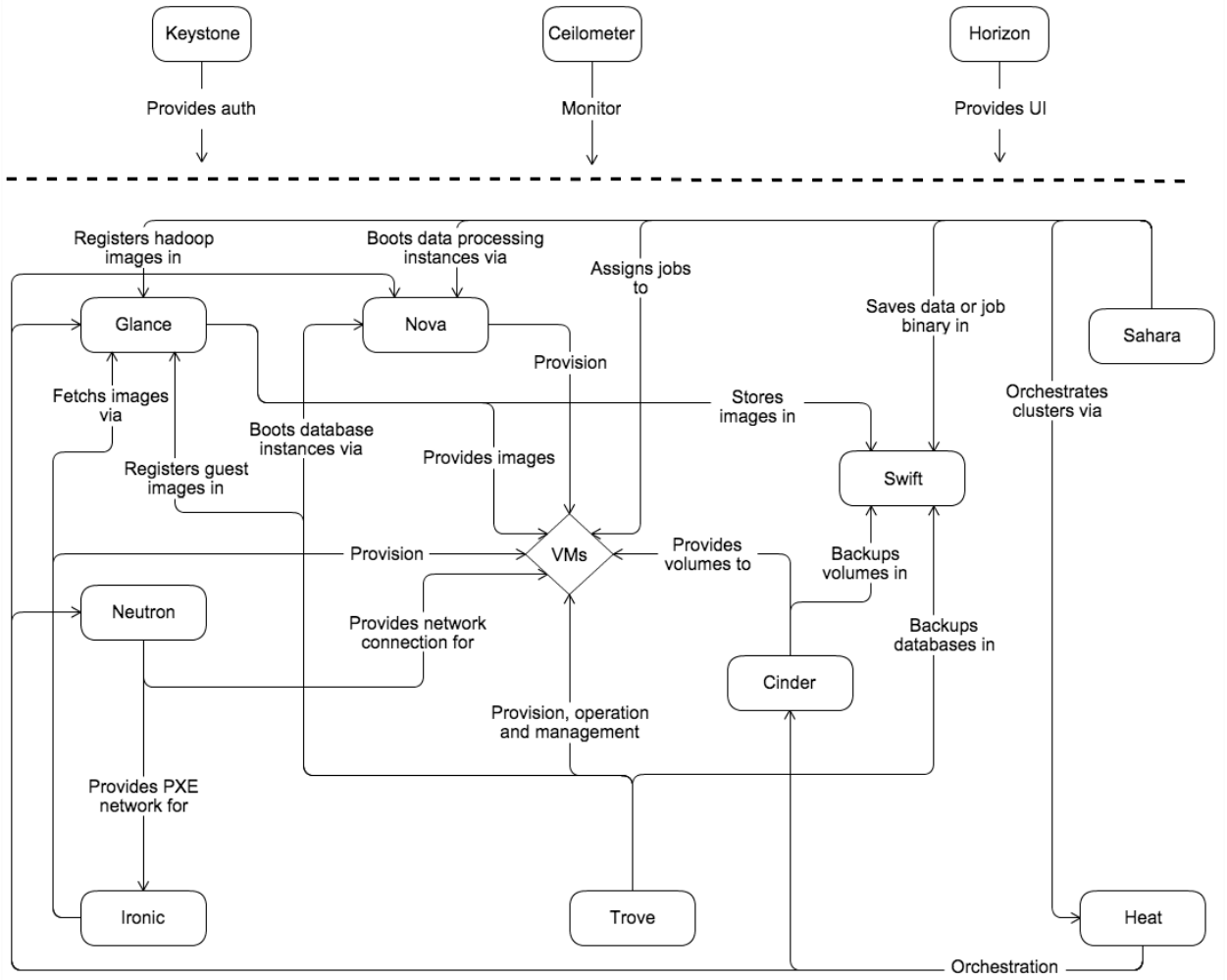


Figure 5.6 Relationship between OpenStack services

- To design, deploy, and configure OpenStack, administrators must understand the logical architecture.

- OpenStack consists of several independent parts, named the OpenStack services. All services authenticate through a common Identity service.

- Individual services interact with each other through public APIs, except where privileged administrator commands are necessary.

- Internally, OpenStack services are composed of several processes.

- All services have at least one API process, which listens for API requests, preprocesses them and passes them on to other parts of the service.

- With the exception of the Identity service, the actual work is done by distinct processes.

- For communication between the processes of one service, an AMQP message broker is used.

- The service's state is stored in a database.

- When deploying and configuring the OpenStack cloud, administrator can choose among several message broker and database solutions, such as RabbitMQ, MySQL, MariaDB, and SQLite.

- Users can access OpenStack via the web-based user interface implemented by the Horizon Dashboard, via command-line clients and by issuing API requests through tools like browser plug-ins or curl.

- For applications, several SDKs are available. Ultimately, all these access methods issue REST API calls to the various OpenStack services.

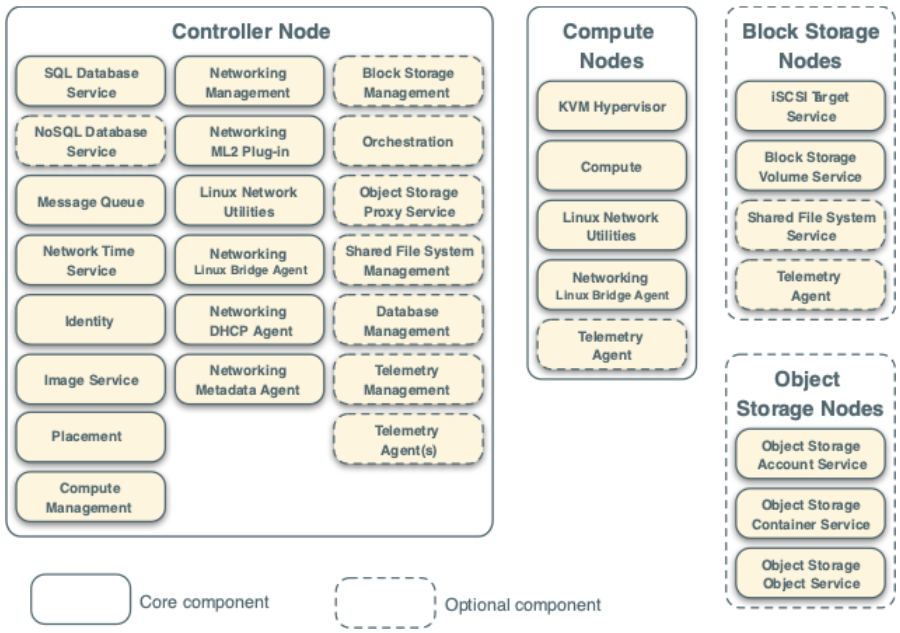


Figure 5.7 Example OpenStack architecture

- The controller node runs the Identity service, Image service, Placement service, management portions of Compute, management portion of Networking, various Networking agents, and the Dashboard.

- It also includes supporting services such as an SQL database, message queue, and NTP.

    - Optionally, the controller node runs portions of the Block Storage, Object Storage, Orchestration, and Telemetry services.

- The controller node requires a minimum of two network interfaces.

- The compute node runs the hypervisor portion of Compute that operates instances. By default, Compute uses the KVM hypervisor.

- The compute node also runs a Networking service agent that connects instances to virtual networks and provides firewalling services to instances via security groups.

- Administrator can deploy more than one compute node. Each node requires a minimum of two network interfaces.

- The optional Block Storage node contains the disks that the Block Storage and Shared File System services provision for instances.

- For simplicity, service traffic between compute nodes and this node uses the management network.

- Production environments should implement a separate storage network to increase performance and security.

- Administrator can deploy more than one block storage node. Each node requires a minimum of one network interface.

- The optional Object Storage node contains the disks that the Object Storage service uses for storing accounts, containers, and objects.

- For simplicity, service traffic between compute nodes and this node uses the management network.

- Production environments should implement a separate storage network to increase performance and security.

- This service requires two nodes. Each node requires a minimum of one network interface. Administrator can deploy more than two object storage nodes.

- The provider networks option deploys the OpenStack Networking service in the simplest way possible with primarily layer 2 (bridging/switching) services and VLAN segmentation of networks.

- Essentially, it bridges virtual networks to physical networks and relies on physical network infrastructure for layer-3 (routing) services.

- Additionally, a DHCP service provides IP address information to instances.

5.7 Federation in the Cloud

- One challenge in creating and managing a globally decentralized cloud computing environment is maintaining consistent connectivity between untrusted components while remaining fault tolerant.

- A key opportunity for the emerging cloud industry will be in defining a federated cloud ecosystem by connecting multiple cloud computing providers using a common standard.

- A notable research project being conducted by Microsoft called the Geneva Framework. This framework focuses on issues involved in cloud federation.

- Geneva has been described as claims based access platform and is said to help simplify access to applications and other systems.

- The concept allows for multiple providers to interact seamlessly with others and it enables developers to incorporate various authentication models that will work with any corporate identity system, including Active Directory,

- LDAPv3 based directories, application specific databases, and new user centric identity models such as LiveID, OpenID, and InfoCard systems.

- It also supports Microsoft's CardSpace and Novell's Digital Me.

- Federation in cloud is implemented by the use of Internet Engineering Task Force (IETF) standard Extensible Messaging and Presence Protocol (XMPP) and inter domain federation using the Jabber Extensible Communications Platform (Jabber XCP).

- Because this protocol is currently used by a wide range of existing services offered by providers as diverse as Google Talk, Live Journal, Earthlink, Facebook, ooVoo, Meebo, Twitter, the U.S. Marines Corps, the Defense Information Systems Agency (DISA), the U.S. Joint Forces Command (USJFCOM), and the National Weather Service.

- Session Initiation Protocol (SIP), which is the foundation of popular enterprise messaging systems such as IBM's Lotus Sametime and Microsoft's Live Communications Server (LCS) and Office Communications Server (OCS).

- Jabber XCP is a highly scalable, extensible, available, and device-agnostic presence solution built on XMPP and supports multiple protocols such as Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) and Instant Messaging and Presence Service (IMPS).

- Jabber XCP is a highly programmable platform, which makes it ideal for adding presence and messaging to existing applications or services and for building next-generation, presence based solutions.

- Over the last few years there has been a controversy brewing in web services architectures.

- Cloud services are being talked up as a fundamental shift in web architecture that promises to move us from interconnected silos to a collaborative network of services whose sum is greater than its parts.

- The problem is that the protocols powering current cloud services, SOAP (Simple Object Access Protocol) and a few other assorted HTTP based protocols, are all one-way information exchanges.

- Therefore cloud services are not real time, would not scale, and often cannot clear the firewall.

- Many believe that those barriers can be overcome by XMPP (also called Jabber) as the protocol that will fuel the Software as a Service (SaaS) models of tomorrow.

- Google, Apple, AOL, IBM, Live journal and Jive have all incorporated this protocol into their cloud based solutions in the last few years.

- Since the beginning of the Internet era, if the user wanted to synchronize services between two servers, the most common solution was to have the client “ping” the host at regular intervals, which is known as polling.

- Polling is how most of us check our email.

- XMPP’s profile has been steadily gaining since its inception as the protocol behind the open source instant messenger (IM) server jabberd in 1998.

- XMPP’s advantages include:

    - It is decentralized, meaning anyone may set up an XMPP server.
    - It is based on open standards.
    - It is mature multiple implementations of clients and servers exist.

- Robust security is supported via Simple Authentication and Security Layer (SASL) and Transport Layer Security (TLS).

- It is flexible and designed to be extended.

- XMPP is a good fit for cloud computing because it allows for easy two way communication

- XMPP eliminates the need for polling and focus on rich publish subscribe functionality

- It in it is XML-based and easily extensible, perfect for both new IM features and custom cloud services

- It is efficient and has been proven to scale to millions of concurrent users on a single service (such as Google's GTalk). And also it has a built-in worldwide federation model.

- Of course, XMPP is not the only pub-sub enabler getting a lot of interest from web application developers.

- An Amazon EC2-backed server can run Jetty and Cometd from Dojo.

- Unlike XMPP, Comet is based on HTTP and in conjunction with the Bayeux Protocol, uses JSON to exchange data.

- Given the current market penetration and extensive use of XMPP and XCP for federation in the cloud and that it is the dominant open protocol in that space.

- The ability to exchange data used for presence, messages, voice, video, files, notifications, etc., with people, devices and applications gain more power when they can be shared across organizations and with other service providers.

- Federation differs from peering, which requires a prior agreement between parties before a server-to-server (S2S) link can be established.

- In the past, peering was more common among traditional telecommunications providers (because of the high cost of transferring voice traffic).

- In the brave new Internet world, federation has become a de facto standard for most email systems because they are federated dynamically through Domain Name System (DNS) settings and server configurations.

5.8 Four Levels of Federation

- Federation is the ability for two XMPP servers in different domains to exchange XML stanzas.

- According to the XEP-0238: XMPP Protocol Flows for Inter-Domain Federation, there are at least four basic types of federation:

- Permissive federation

    - Permissive federation occurs when a server accepts a connection from a peer network server without verifying its identity using DNS lookups or certificate checking.
    - The lack of verification or authentication may lead to domain spoofing (the unauthorized use of a third-party domain name in an email message in order to pretend to be someone else), which opens the door to widespread spam and other abuses. With the release of the open source jabberd 1.2 server in October 2000, which included support for the Server Dialback protocol (fully supported in Jabber XCP), permissive federation met its demise on the XMPP network.

- Verified federation

    - This type of federation occurs when a server accepts a connection from a peer after the identity of the peer has been verified.
    - It uses information obtained via DNS and by means of domain-specific keys exchanged beforehand.
    - The connection is not encrypted, and the use of identity verification effectively prevents domain spoofing.
    - To make this work, federation requires proper DNS setup and that is still subject to DNS poisoning attacks.

    - Verified federation has been the default service policy on the open XMPP since the release of the open-source jabberd 1.2 server.

- Encrypted federation

    - In this mode, a server accepts a connection from a peer if and only if the peer supports Transport Layer Security (TLS) as defined for XMPP in Request for Comments (RFC) 3920.
    - The peer must present a digital certificate.
    - The certificate may be self signed, but this prevents using mutual authentication.
    - If this is the case, both parties proceed to weakly verify identity using Server Dialback.
    - XEP-0220 defines the Server Dialback protocol, which is used between XMPP servers to provide identity verification.
    - Server Dialback uses the DNS as the basis for verifying identity
    - The basic approach is that when a receiving server receives a server-to-server connection request from an originating server, it does not accept the request until it has verified a key with an authoritative server for the domain asserted by the originating server.
    - Although Server Dialback does not provide strong authentication or trusted federation, and although it is subject to DNS poisoning attacks, it has effectively prevented most instances of address spoofing on the XMPP network since its release in 2000.
    - This results in an encrypted connection with weak identity verification.

- Trusted federation

    - In this federation, a server accepts a connection from a peer only under the stipulation that the peer supports TLS and the peer can present a digital certificate issued by a root certification authority (CA) that is trusted by the authenticating server.
    - The list of trusted root CAs may be determined by one or more factors, such as the operating system, XMPP server software or local service policy.

- In trusted federation, the use of digital certificates results not only in a channel encryption but also in strong authentication.
- The use of trusted domain certificates effectively prevents DNS poisoning attacks but makes federation more difficult, since such certificates have traditionally not been easy to obtain.

5.9 Federated Services and Applications

- S2S federation is a good start toward building a real-time communications cloud.
- Clouds typically consist of all the users, devices, services, and applications connected to the network.
- In order to fully leverage the capabilities of this cloud structure, a participant needs the ability to find other entities of interest.
- Such entities might be end users, multiuser chat rooms, real-time content feeds, user directories, data relays, messaging gateways, etc.
- Finding these entities is a process called discovery.
- XMPP uses service discovery (as defined in XEP-0030) to find the aforementioned entities.
- The discovery protocol enables any network participant to query another entity regarding its identity, capabilities and associated entities.
- When a participant connects to the network, it queries the authoritative server for its particular domain about the entities associated with that authoritative server.

- In response to a service discovery query, the authoritative server informs the inquirer about services hosted there and may also detail services that are available but hosted elsewhere.

- XMPP includes a method for maintaining personal lists of other entities, known as roster technology, which enables end users to keep track of various types of entities.

- Usually, these lists are comprised of other entities the users are interested in or interact with regularly.

- Most XMPP deployments include custom directories so that internal users of those services can easily find what they are looking for.

5.10 Future of Federation

- The implementation of federated communications is a precursor to building a seamless cloud that can interact with people, devices, information feeds, documents, application interfaces and other entities.

- The power of a federated, presence enabled communications infrastructure is that it enables software developers and service providers to build and deploy such applications without asking permission from a large, centralized communications operator.

- The process of server-to-server federation for the purpose of inter domain communication has played a large role in the success of XMPP, which relies on a small set of simple but powerful mechanisms for domain checking and security to generate verified, encrypted, and trusted connections between any two deployed servers.

- These mechanisms have provided a stable, secure foundation for growth of the XMPP network and similar real time technologies.

TWO MARK QUESTIONS

1. What is Hadoop?

   - 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.

2. List the fundamental layers of Hadoop core.

   - The Hadoop core is divided into two fundamental layers:
     - MapReduce engine
     - HDFS

3. Describe about HDFS.

   - HDFS is a Hadoop distributed file system inspired by GFS that organizes files and stores their data on a distributed computing system.
   - 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 fixed-size blocks (e.g., 64 MB) and stores them on workers (DataNodes).
   - The mapping of blocks to DataNodes is determined by the NameNode.

4. Is HDFS provides fault tolerant?

   - One of the main aspects of HDFS is its fault tolerance characteristic. Since Hadoop is designed to be deployed on low-cost hardware by default, a hardware failure in this system is considered to be common rather than an exception.

5. List the issues to fulfill reliability requirements of the file system by hadoop.

- Block replication
- Replica placement
- Heartbeat and Block report messages

6. What is the purpose of heartbeat messages?

   - Heartbeat is a periodic message sent to the NameNode by each DataNode in a cluster.

7. List the advantages of HDFS.

   - 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.

8. Define MapReduce.

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

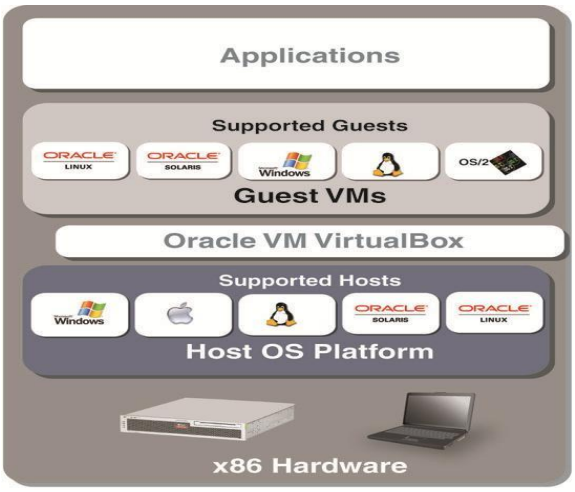
9. List the components contribute in running a job in Hadoop system.

   - a user node
   - a JobTracker
   - TaskTrackers

10. What is the use of VirtualBox?

- Oracle VM VirtualBox is a cross-platform virtualization application.
- For one thing, it installs on the existing Intel or AMD-based computers, whether they are running Windows, Mac OS X, Linux, or Oracle Solaris operating systems (OSes).
- Secondly, it extends the capabilities of existing computer so that it can run multiple OSes, inside multiple virtual machines, at the same time.

11. Illustrate the architecture of VirtualBox.



12. List the three disk image formats used in VirtualBox:

- VDI: This format is the VirtualBox-specific VirtualBox Disk Image and stores data in files bearing a ".vdi".
- VMDK: This open format is used by VMware products and stores data in one or more files bearing ".vmdk" filename extensions.
- VHD: This format is used by Windows Virtual PC and Hyper-V, and is the native virtual disk format of the Microsoft Windows operating system.

13. Describe about GAE.

- Google's App Engine (GAE) which offers a PaaS platform supporting various cloud and web applications.

- 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.

14. Mention the components maintained in a node of Google cloud platform.

- 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.

15. List the functional modules of GAE.

- Datastore
- Application runtime environment
- Software development kit (SDK)
- Administration console
- GAE web service infrastructure

16. List the applications of GAE.

- 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.

17. Mention the goals for design and implementation of the BigTable system.

- 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.
- The application may need to examine data changes over time.

18. Describe about Openstack.

- The OpenStack project is an open source cloud computing platform for all types of clouds, which aims to be simple to implement, massively scalable, and feature rich.
- Developers and cloud computing technologists from around the world create the OpenStack project.
- OpenStack provides an Infrastructure-as-a-Service (IaaS) solution through a set of interrelated services.

19. List the key services of OpenStack.

- The OpenStack system consists of several key services that are separately installed.
- Compute, Identity, Networking, Image, Block Storage, Object Storage, Telemetry, Orchestration and Database services.

20. What is the need of federated cloud ecosystem?

- One challenge in creating and managing a globally decentralized cloud computing environment is maintaining consistent connectivity between untrusted components while remaining fault-tolerant.
- A key opportunity for the emerging cloud industry will be in defining a federated cloud ecosystem by connecting multiple cloud computing providers using a common standard.
- A notable research project being conducted by Microsoft, called the Geneva Framework, focuses on issues involved in cloud federation.

21. List the advantages of Extensible Messaging and Presence Protocol.

- XMPP's is decentralized, meaning anyone may set up an XMPP server. It is based on open standards. It is mature multiple implementations of clients and servers exist.

22. List the levels of Federation.

- Permissive federation
- Verified federation
- Encrypted federation
- Trusted federation

23. What is S2S federation?

- S2S federation is a good start toward building a real-time communications cloud. Clouds typically consist of all the users, devices, services, and applications connected to the network.

24. What is the future of federation?

- The power of a federated, presence enabled communications infrastructure is that it enables software developers and service providers to build and deploy such applications without asking permission from a large, centralized communications operator.
- These mechanisms have provided a stable, secure foundation for growth of the XMPP network and similar real time technologies.

MODEL QUESTION PAPER - I

B.E./B.Tech. DEGREE EXAMINATION

Seventh Semester

Computer Science and Engineering

CS8791 – Cloud Computing

(Regulation 2017)

Time: Three hours Maximum: 100 marks

Answer ALL questions

PART A – (10 X 2 = 20 marks)

1. Define Cloud.
2. List the components of cloud model.
3. Mention the four characteristics to identify the service.
4. Differentiate between Full virtualization and Paravirtualization.
5. What are advantages of cloud storage?
6. What is Hardware as a Service?
7. What is the purpose of runtime support service named cluster monitoring?
8. Compare over provisioning and under provisioning?
9. Illustrate the architecture of VirtualBox.
10. List the merits of XMPP.

PART B – (5 X 16 = 80 marks)

11. (a) Explain about evolution of cloud computing.

Or

(b) (i) Explain about the elements of parallel and distributed computing. (8)

(ii) Explain about elasticity nature of cloud computing and on-demand provisioning. (8)

12. (a) (i) Explain about Service Oriented Architecture. (8)

(ii) Explain about Publish-Subscribe model. (8)

Or

(b) Explain about various implementation levels of virtualization.

13. (a) (i) Explain about layered architectural design of cloud computing. (8)

(ii) Explain about cloud deployment models. (8)

Or

(b) Explain about major architectural design challenges in cloud. (16)

14. (a) (i) Explain about inter cloud resource management with neat diagram. (8)

(ii) Explain about resource provisioning methods. (8)

Or

(b) (i) Explain about Identity Access Management. (8)

(ii) Explain about Virtual Machine Security. (8)

15. (a) Explain about HDFS and MapReduce in Hadoop framework. (16)

Or

(b) (i) Explain about Programming environment for Google AppEngine (8)

(ii) Explain about the levels of federation. (8)

MODEL QUESTION PAPER - II

B.E./B.Tech. DEGREE EXAMINATION

Seventh Semester

Computer Science and Engineering

CS8791 – Cloud Computing

(Regulation 2017)

Time: Three hours Maximum: 100 marks

Answer ALL questions

PART A – (10 X 2 = 20 marks)

1. Differentiate between Parallel and Distributed computing.
2. List the various models for message based communication.
3. Define service oriented architecture.
4. Illustrate ring based security with neat diagram.
5. Compare Public cloud and Private cloud.
6. What are the design requirements considers by Amazon to build S3?
7. What is Event-driven provisioning?
8. Mention the purpose of Security Governance.
9. What is purpose of Task tracker and Job tracker in Hadoop?
10. What is the need for federated cloud ecosystem?

PART B – (5 X 16 = 80 marks)

11. (a) Explain about the principles of Parallel and Distributed Computing. (16)

Or

(b) Explain about characteristics of cloud computing. (16)

12. (a) (i) Explain about RESTful Systems. (8)

(ii) Explain about Web service technologies stack. (8)

Or

(b) (i) Explain about CPU, Memory and I/O device virtualization . (8)

(ii) Explain about virtualization support and disaster recovery strategies.(8)

13. (a) Explain about NIST reference architecture with neat diagram. (16)

Or

(b) (i) Explain about cloud service model. (8)

(ii) Explain about Storage-as-a-Service. (8)

14. (a) (i) Explain about global exchange of cloud resources (8).

(ii) Explain about runtime support services in inter cloud management. (8)

Or

(b) Explain about cloud security and its challenges. Elaborate some standards specific to cloud security. (16)

15. (a) Explain about functional modules and programming environment of Google App Engine. (16)

Or

(b) Explain about OpenStack architecture with neat diagram. (16)

"Books are my favorite friends and I Consider my home library, with many thousand books, to be my greatest wealth. Every new book, based on some new idea inspires me and gives me a new thought to ponder."

— Dr. A.P.J. Abdul Kalam
