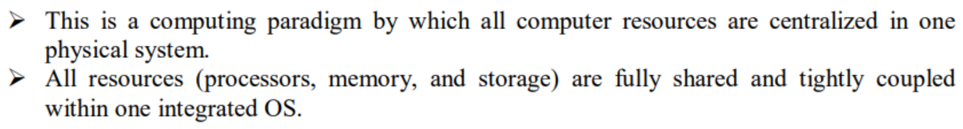
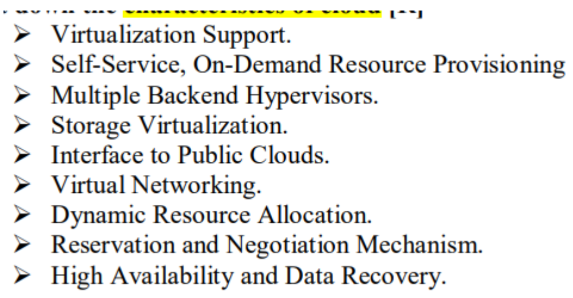
What is meant by centralized computing



- This is a computing paradigm by which all computer resources are centralized in one physical system.
- All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS.

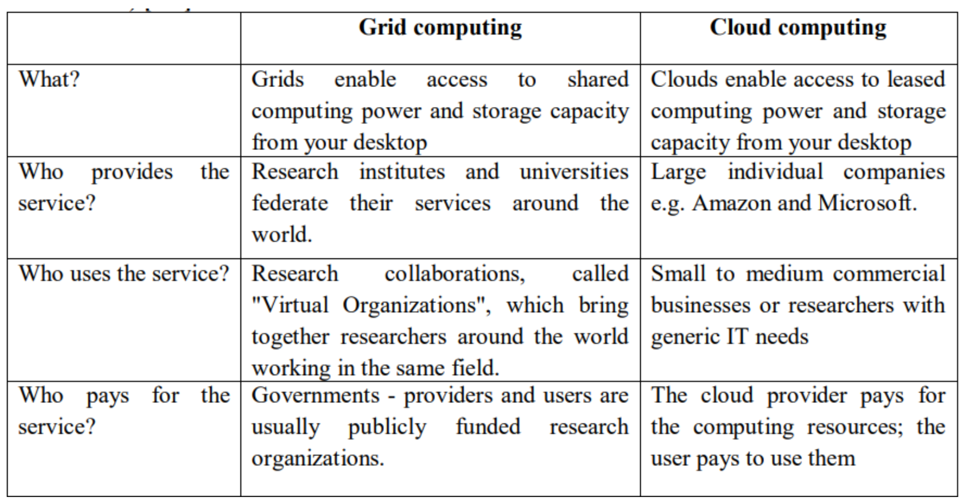
List few drawbacks of Grid Computing

List the main characteristics of cloud computing



- Virtualization Support.
- Self-Service, On-Demand Resource Provisioning
- Multiple Backend Hypervisors.
- Storage Virtualization.
- Interface to Public Clouds.
- Virtual Networking.
- Dynamic Resource Allocation.
- Reservation and Negotiation Mechanism.
- High Availability and Data Recovery.

Compare grid computing and cloud computing



	Grid computing	Cloud computing
What?	Grids enable access to shared computing power and storage capacity from your desktop	Clouds enable access to leased computing power and storage capacity from your desktop
Who provides the service?	Research institutes and universities federate their services around the world.	Large individual companies e.g. Amazon and Microsoft.
Who uses the service?	Research collaborations, called "Virtual Organizations", which bring together researchers around the world working in the same field.	Small to medium commercial businesses or researchers with generic IT needs
Who pays for the service?	Governments - providers and users are usually publicly funded research organizations.	The cloud provider pays for the computing resources; the user pays to use them

Give the sample REST Request-Response for creating a S3 Bucket

Define host-based virtualization

In host-based virtualization, the virtualization layer runs on top of the host OS and

guest OS runs over the virtualization layer. Therefore, host OS is responsible for

managing the hardware and control the instructions executed by guest OS. The host based virtualization doesn’t require to modify the code in host OS but virtualization

software has to rely on the host OS to provide device drivers and other low-level services.

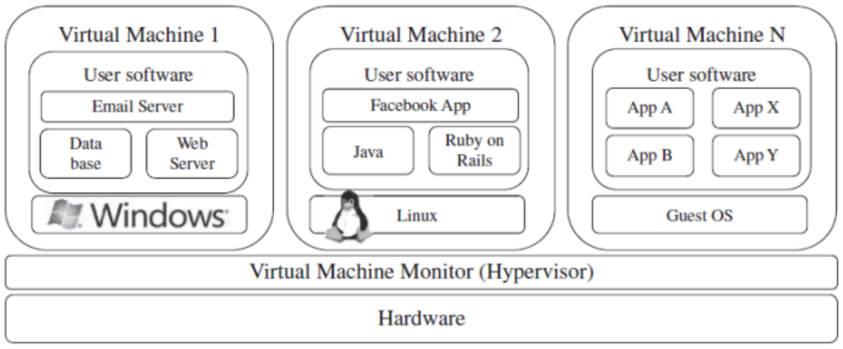
This architecture simplifies the VM design with ease of deployment but gives degraded

performance compared to other hypervisor architectures because of host OS

interventions. The host OS performs four layers of mapping during any IO request by guest OS or VMM which downgrades performance significantly.

What is hypervisor?

A hypervisor, also called a virtual machine manager, is a program that allows multiple operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and other resources all to itself.



What are the fundamental components of SOAP specification?

State the role of cloud auditor .

The cloud auditor performs the task of independently evaluating cloud service controls to provide an honest opinion when requested. Cloud audits are done to validate standards conformance by reviewing the objective evidence. The auditor will examine services provided by the cloud provider for its security controls, privacy, performance, and so on.

Analyze the storage as a service.

Why does one choose public cloud over private cloud? Analyze

Show the interaction between the Actors in the cloud computing

What are the different resource provisioning methods?

a. Demand-Driven Resource Provisioning
b. Event-Driven Resource Provisioning
c. Popularity-Driven Resource Provisioning

Define cloud security governance

What are the six layers of cloud services?

Generalize about the IAM.

The Identity and access management in cloud computing is the security framework composed of policy and governance components used for creation, maintenance and termination of digital identities with controlled access of shared resources. It composed of multiple processes, components, services and standard practices. It focuses on two parts namely Identity management and access management. The directory services are used in IAM for creating a repository for identity management, authentication and access management. The IAM provides many features like user management, authentication management, authorization management, credential and attribute management, compliance management, monitoring and auditing etc.

Differentiate name node with data node in Hadoop file system.

What are the different ways of storing application data in Google App Engine.

- Datastore
- Google Cloud SQL
- Google Cloud Storage

Identify the development technologies currently modules in Hadoop supported by AppEngine

Name the different modules in Hadoop framework

Examine the challenges and risks available in cloud computing.

What is meant Scale-Up Scale-Down?

State the responsibilities of VMM.

Define the advantages of using the cloud storage.

- Accessibility
- Greater collaboration
- Security
- Cost - efficient
- Instant data recovery
- Syncing and updating
- Disaster recovery

Illustrate anything as a service.

Providing services to the client on the basis on meeting their demands at some pay per use cost such as data storage as a service, network as a service, communication as a service etc. It is generally denoted as anything as a service XaaS.

Unit 1

Evolution of cloud computing

Evolution of Cloud Computing

The cloud computing becomes very popular in short span of time along with delivery of prominent and unique benefits which were never before. Therefore, it is important to understand evaluation of cloud computing. In this section we are going to understand the evolution of cloud computing with respect to hardware, internet, protocol, computing and processing technologies.

1.3.1 Evolution of Hardware

a) First-Generation Computers : The first-generation computing hardware called Mark and Colossus, which was used for solving the binary arithmetic. It was developed in 1930 which became foundation for programming languages, computer processing and terminologies. The first generation of computer was evolved with second version in 1943 at Harvard university which was an electromechanical programable computer by mark and colossus. It was developed using vacuum tubes and hardwire circuits where punch cards were used to stored data.

b) Second-Generation Computers : The second-generation computing hardware called ENIAC (Electronic Numerical Integrator and Computer) builded in 1946, which was capable to solve range of computing problems. It was capable to perform one lakh calculations per seconds. It was composed of thermionic valves, transistors and circuits.

c) Third-Generation Computers : The Third-generation computers were produced in 1958 using integrated circuits (IC's). The first mainframe computer by IBM was developed in this era. The IBM 360 has got more processing and storage capabilities due to the integrated circuit. The minicomputer was also developed in this era. At the later stage, intel has released first commercial microprocessor called intel 4004 which has multiple transistors integrated on a single chip to perform processing at faster speed.

d) Fourth-Generation Computers : The fourth-generation computer have introduced microprocessors with single integrated circuits and random-access memory for performing execution of millions of instructions per seconds. In this phase IBM have developed an personal computers in 1981 along with LSI and VLSI Microchips.

1.3.2 Evolution of Internet and Protocols

The evolution of internet has begun in 1930 with the concept of MEMEX for storing books records and communication. In 1957 Soviet Union have launched the first satellite to create Advanced Research Project Agency (ARPA) for US military. The internet was firstly introduced with the creation of ARPANET in 1967 where they have used internet message processors (IMP) at each site for communication. In ARPANET, initially host to host protocols were used for communication which were evolved with application protocols like FTP and SMTP in 1983. ARPANET has introduced a flexible and powerful TCP-IP protocol suit which is used over the internet till today. The internet protocol had initial version IPV4 which again evolved with new generation IPV6 protocol. The first web browser MOSAIC was developed in 1990 by Berners-Lee followed by Netscape browser in 1994.In 1995, Microsoft had developed a Windows 95 operating system with integrated browser called Internet Explorer along with supporting dial-up TCP/IP protocols. The first web server work on hypertext transfer protocol released in 1996 followed by various scriptings supported web servers and web browsers.

1.3.3 Evolution of Computing Technologies

Few decades ago, the popular computing technology for processing a complex and large computational problem was "Cluster computing". It has group of computers were used to solve a larger computational problem as a single unit. It was designed such a way that the computational load used to divide in to similar unit of work and allocated across multiple processors which is balanced across the several machines. In 1990, the cluster computing was evolved with the concept of Grid Computing which was developed by Ian Foster. The grid computing is nothing but the group of interconnected independent computers intended to solve a common computational problem as a single unit. They are usually geographically distributed across different locations like city, country or continents. It was analogous to electric Grids where users were allowed to plug in and use the computing power as like utility service. But the main limitation of grid computing was data residency, as data were located and stored at geographically diverse locations miles away from each other's. So further, grid computing is evolved with the cloud computing where centralized entity like data centers is used to offer different computing services to others which is similar to grid computing model. The cloud computing becomes more popular with the introduction of "Virtualization" technology. The Virtualization is a method of running multiple independent virtual operating systems on a single physical computer. It saves hardware cost due to consolidation of multiple servers along with maximum throughput and optimum resource utilization.

1.3.4 Evolution of Processing Technologies

When computers were initially launched, people used to work with mechanical devices, vacuum tubes, transistors, etc. Then with the advent of Small-Scale Integration (SSI), Medium Scale Integration (MSI), Large Scale Integration (LSI), and Very Large-Scale Integration (VLSI) technology, circuits with very small dimension became more reliable and faster. This development in hardware technology gave new dimension in designing processors and its peripherals. The processing is nothing but the execution of programs, applications or tasks on one or more computers. The two basic approaches of processing are serial and parallel processing. In serial processing, the given problem or task is broken into a discrete series of instructions. These instructions are executed on a single processor sequentially. In Parallel processing, the tasks of programming instructions are executed simultaneously across multiple processors with the objective of running program in a lesser time. The next advancement in parallel processing was multiprogramming. In Multiprogramming system, the multiple programs are submitted at same time for execution where each program is allowed to use the processor for a specific period of allotted time. Here, each program gets an equal amount of time to use the processor in round robin manner in order to execute the instructions. Later, multiprogramming system was evolved with vector processing. The vector processing was developed to increase the processing performance by operating in a multitasking manner. It was specially designed to perform Matrix operations to allow a single instruction to manipulate two arrays of numbers performing arithmetic operations. The vector processing was used in certain applications where the data generated in the form of vectors or matrices. The next advancement to vector processing was the development of symmetric multiprocessing systems (SMP). As multiprogramming and vector processing system has limitation of managing the resources in master slave model, the symmetric multiprocessing systems was designed to address that problem. The SMP systems is intended to achieve sequential consistency where each processor is assigned an equal number of OS tasks. These processors are responsible for managing the workflow of task execution as it passes through the system. Lastly, Massive parallel processing (MPP) is developed with many independent arithmetic units or microprocessors that runs in parallel and are interconnected to act as a single very large computer. Today, the massively parallel processor arrays can be implemented into a single-chip which becomes cost effective due to the integrated circuit technology and it is mostly used in advanced computing applications used in artificial intelligence.

Underlying principles of parallel and cloud computing

Underlying Principles of Parallel and Distributed Computing

In previous section, we have seen the evolution of cloud computing with respect to its hardware, internet, protocol and processing technologies. This section briefly explains about the principals of two essential computing mechanisms which largely used in cloud computing called Parallel and Distributed computing. Computing in computer technology can be defined as the execution of single or multiple programs, applications, tasks or activities, sequentially or parallelly on one or more computers. The two basic approaches of computing are serial and parallel computing.

Serial Computing

In serial computing, the given problem or task is broken into a discrete series of instructions. These instructions are executed on a single processor sequentially as shown in Fig. 1.4.1. The problem in serial computing is that it executes only one instruction at any moment of time. It is mostly used in monolithic applications on single machine which do not have any time constraint.

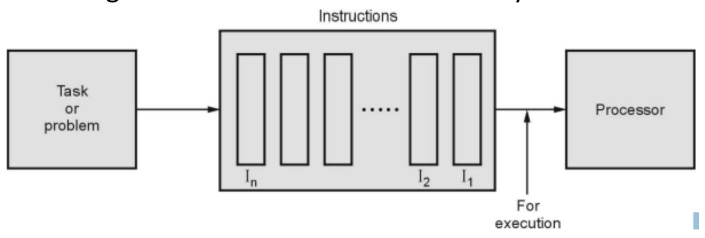


Fig. 1.4.1 : Serial Computing

Parallel Computing

As single processor system is becoming archaic and quaint for doing fast computation as required by real-time applications. So parallel computing is needed to speed up the execution of real-time applications to achieve high performance. The parallel computing makes use of multiple computing resources to solve a complex computational problem in which the problem is broken into discrete parts that can be solved concurrently as shown in Fig. 1.4.2.

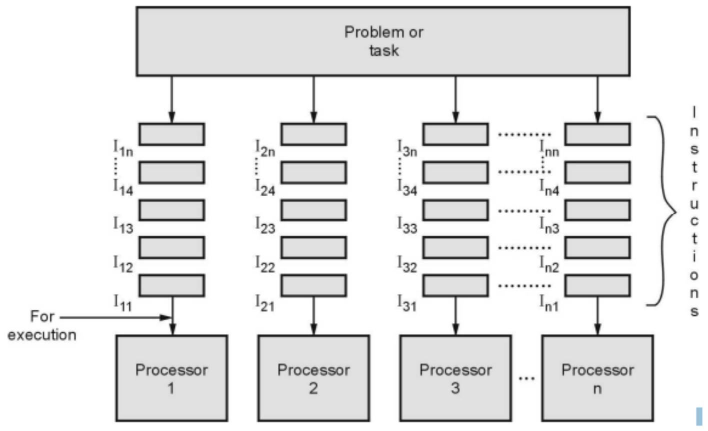


Fig. 1.4.2 : Parallel Computing

Each part is further broken down into a series of instructions which execute simultaneously on different processors using overall control/coordination mechanism. Here, the different processors share the work-load which results in producing the much higher computing power and performance than could not be achieved with traditional single processor system.

The parallel computing often correlated with parallel processing and parallel programming. Processing of multiple tasks and subtasks simultaneously on multiple processors is called parallel processing while parallel programming refers to programming on a multiprocessor system using the divide-and-conquer technique, where given task is divided into subtasks and each subtask is processed on different processors.

Hardware Architectures for Parallel Processing

The hardware architecture of parallel computing is disturbed along the following categories as given below :

1. Single-instruction, single-data (SISD) systems
    - SISD computing system is a uni-processor machine capable of executing a single instruction, which operates on a single data stream.
    - Machine instructions are processed sequentially, hence computers adopting this model are popularly called sequential computers.
    - Most conventional computers are built using SISD model.

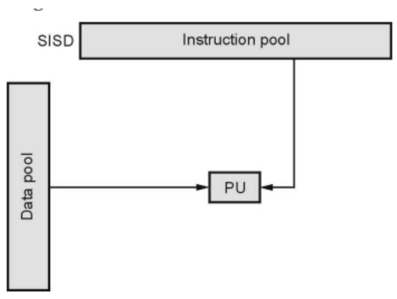


Fig. 1.4.4 : SISD architecture

2. Single-instruction, multiple-data (SIMD) systems
   • SIMD computing system is a multiprocessor machine capable of executing the same instruction on all the CPUs but operating on different data streams.
   • Machines based on this model are well suited for scientific computing since they involve lots of vector and matrix operations.

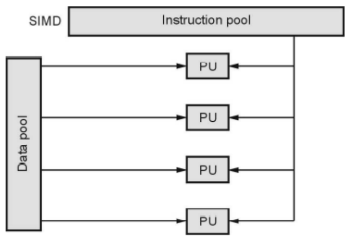


Fig. 1.4.5 : SIMD architecture.

3. Multiple-instruction, single-data (MISD) systems

   • MISD computing system is a multi processor machine capable of executing different instructions on different Pes all of them operating on the same data set.
   • Machines built using MISD model are not useful in most of the applications.
   • Few machines are built but none of them available commercially.

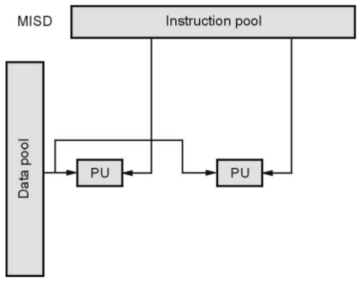


Fig. 1.4.6 : MISD architecture.

4. Multiple-instruction, multiple-data (MIMD) systems
   • MIMD computing system is a multi processor machine capable of executing multiple instructions on multiple data sets.
   • Each PE in the MIMD model has separate instruction and data streams, hence machines built using this model are well suited to any kind of application.
   • Unlike SIMD, MISD machine, PEs in MIMD machines work asynchronously

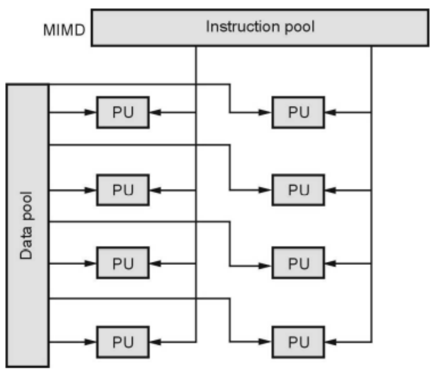


Fig. 1.4.7 : MIMD architecture.

Shared Memory Architecture for Parallel Computers

Uniform Memory Access (UMA)

- It is represented by symmetric multiprocessor machines
- Identical processes equal access and access time to memory

Non-Uniform Memory Access (NUMA)

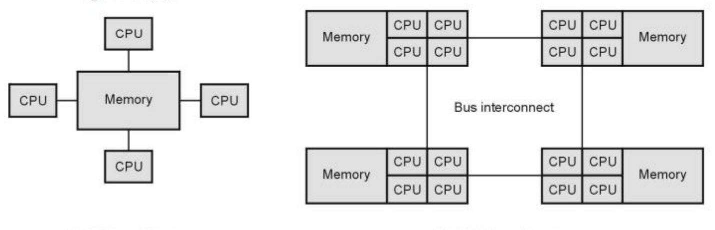
- It can directly access the memory of another SMP.
- Equal access time to all memories.
- Memory access across the link is slower.

Advantages

- It is a global address space.
- User friendly programming prospective to memory.
- Data sharing between tasks is both fast and uniform due to the memory CPU.

Disadvantages

- Scalability between memory and CPU
- Increases traffic associated with cache or memory.
- Synchronisation constructs the correct access to global memory.



(a) UMA architecture (b) NUMA architecture.

Distributed Memory Architecture for Parallel Computers

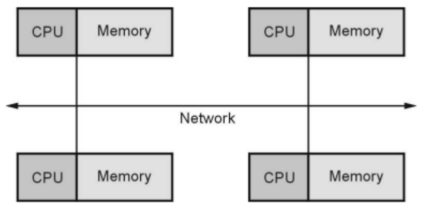


Fig. 1.4.9 : Distributed memory architecture.

Distributed Computing

- A distributed system is a collection of large amount of independent computers that appear to its users as a single coherent system.
- Distributed Systems have their own local memory

Advantages

- Memory is scalable with the number of processors
- Processes can be accessed rapidly with its own memory without any interference

- Cost effective use of commodity processors and networking
- Disadvantages
- The programmer is responsible for many of these details associated with data communication between processes
- Non-uniform memory access time data residing in a remote node to access local data

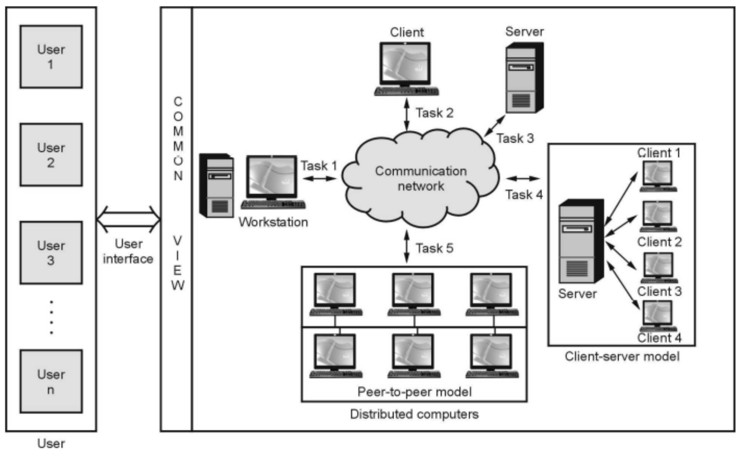
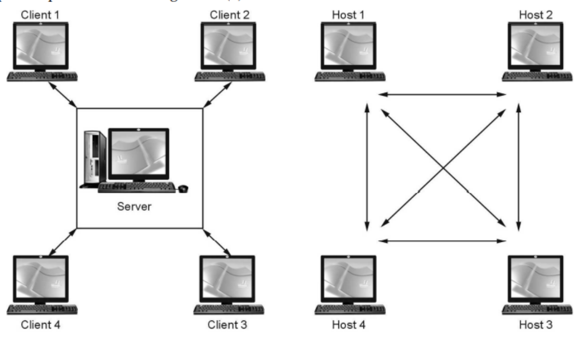


Fig. 1.4.10 : Conceptual view of distributed system

<u>Architectural Models for Distributed System</u>



(a) Client-Server model (b) Peer-to-Peer model

The client-server model is widely used architecture in most of the distributed technologies, where client process interacts with the server process to get access to the shared resources. The client-server model is usually based on a simple request/reply protocol implemented with send/receive primitives using communication middlewares like remote procedure calls (RPC) and remote method invocation (RMI). In the client server model, the client process requests for the procedure running on the server machine using underlying network. Once server procedure receives the request, it is executed and result is generated. The processed result is sent back to the client as a response by the server.

In the Peer-to-Peer (P2P) model, there is no distinction between the client and the server process. All computers in the network get same privileges and run same program with the same set of interfaces. The pattern of communication always depends on the type of application. The two disadvantages of P2P are the lack of scalability and high complexity.

In distributed system, the multiple concurrent processes used to interact continuously with each other over the network. The Interprocess communication (IPC) is the fundamental method of communication in distributed systems, its design and implementation. The IPC is used to synchronize the activities of processes locally and exchange the data between the processes. There are several models used in distributed system for making the remote communication between processes. Some of the most relevant models used for communication in distributed system are Remote Procedure Call (RPC), Message Oriented Middleware (MOM) and Remote Method Invocation (RMI).

The Web services and Web 2.0 are the fundamental building blocks for cloud computing. The front end of recent cloud platform is mostly built using Web 2.0 and related technologies while services of cloud are delivered through Web services or SOA based technologies.

On demand provisioning

The on-demand provisioning is another important benefit provided by cloud computing. The public cloud services are available globally through internet. The Cloud Service Providers (CSPs) are responsible for providing their cloud services through a single self-service portal where customers can pick the specific service whichever they want to use in their enterprise. As delivery of cloud services are provided through internet, they are available on-demand everywhere through a self-service portal.

The on-demand provisioning in cloud computing refers to process for the automated deployment, integration and consumption of cloud resources or services by an individuals or enterprise IT organizations. It incorporates the policies, procedures and an enterprise's objective in sourcing the cloud services and solutions from a cloud service provider. The on-demand provisioning is used to provision the cloud services dynamically on demand along with resources like hardware, software, data sets or servers running several choices of operating systems with customized software stacks over the internet. The on-demand provisioning is applicable for every cloud service or resources like hardware, software, data sets or servers running several choices of operating systems with customized software stacks dynamically over the internet.

It is essentially developed to meet the common challenge in computing like fluctuating demand for resources faced by enterprises. As demand for compute resources enterprises vary drastically time to time. Therefore, on-demand provisioning enables enterprises to access additional resources from anywhere, at any time and on any supported device dynamically.

Unit 2

SOA protocols (Rest, soap), web services

<u>REST and Systems of Systems</u>

Representational State Transfer (REST) is a software architectural style for distributed system that defines a set of constraints to be used for creating Web based services. It is mean to provide interoperability between the systems based on services running on the Internet. REST is defined by Roy Fielding (author of HTTP specifications) in his PhD dissertation on "Architectural Styles and the Design of Network-based Software Architectures". Today, it is being used by many of IT enterprises including Yahoo, Google, Amazon, IBM as well as social networking sites such as Twitter, Facebook, and LinkedIn etc.

The web services that follow the REST architectural style are called RESTful Web services. The RESTful web services allow the requesting systems to access and manipulate textual representations of web resources by using a uniform and predefined set of stateless operations. The generalized interaction in REST with HTTP specification is shown in Fig. 2.2.1.

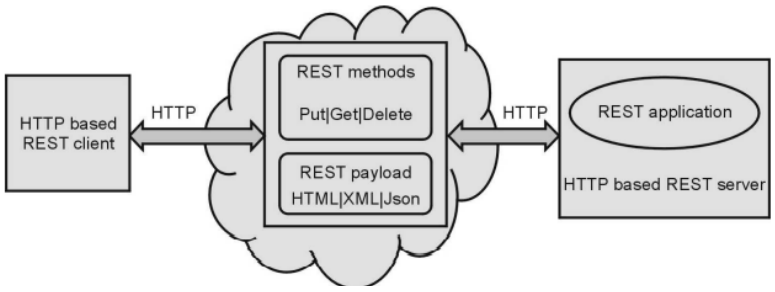


Fig. 2.2.1 Interaction in REST with HTTP specification

The REST architectural style has four basic principles which are explained as follows :

<u>a) Resource identification</u>

- In RESTful web services, the set of resources are often exposed by the publishers over the internet which are accessed by the clients through interaction mechanisms. The key component for information abstraction in REST is a resource. A resource can be any information stored in a document, image or temporal storage which uses conceptual mapping to a set of entities. Each resource in a REST has a unique name identified by a Uniform Resource Identifier (URI) similar to URL on web. The URI is utilized for giving a global addressing space tending to resources which are involved in an interaction between components and facilitates service discovery. The URIs can be bookmarked or traded through a hyperlink which gives greater readability.

b) Controlled Interfaces

In RESTful web services, the interaction is happened through client/server protocols based on HTTP standards. The primitives used to perform manipulation are fixed set of four CRUD (Create, Read, Update, Delete) operations which are implemented using HTTPs PUT, GET, POST and DELETE methods. The operations of REST methods are given in Table 2.2.1.

Method	Operation
PUT	Create a new resource
GET	Retrieve the current state of resource
POST	Update or transfers a new state to a resource
DELETE	Delete or destroy a resource

Table 2.2.1 REST Methods

c) Self-Descriptive Messages

A REST message contains brief description about message communication along with the processing information. It enables intermediate users to process the message without parsing the contents. The REST decouples the resources from their representations such that their content can be accessed in a variety of standard formats like HTML, XML, etc. It also provides the alternate representations of each resource in multiple formats. The message also contains metadata that can be used for detecting the transmission error, caching control, authentication, authorization, and access control.

d) Stateless Communications

In REST, the communication happens are mostly 'stateless' where messages do not have to rely on the state of the conversation. The stateless communication facilitates improved visibility, task of recovering from partial failures, and increased scalability. The limitations of stateless communication are degraded or decreased network performance because of collective repeated data. However, there are some communications happened using Stateful interactions which performs explicit state transfer such as URI rewriting, hidden form fields or cookies. To point the future state of communication, the current state can be embedded in a response message. Mostly the stateless RESTful web services are scalable in nature as they can serve very large number of clients with supporting caching mechanisms, clustering, and load balancing.

The common example of REST web service is Amazon AWS which uses various REST methods in its Simple Storage Service (S3). The Simple Storage Service uses bucket as a medium for storing the objects also called items. For manipulating the bucket, it makes HTTP requests to create, fetch, and delete buckets using PUT, GET, POST and DELETE methods.

The RESTful web services are mainly used in web 2.0 applications where the mashup allows to combine the capabilities of one web application into another, for example, taking the videos from online YouTube repository and put into a Facebook page.

Illustrate the following Virtualization in detail i. CPU virtualization ii. Memory Virtualization iii. I/O Devices

CPU Virtualization

The CPU Virtualization is related to range protection levels called rings in which code can execute. The Intel x86 architecture of CPU offers four levels of privileges known as Ring 0, 1, 2 and 3. Fig. 2.10.1 CPU Privilege Rings

Among that Ring 0, Ring 1 and Ring 2 are associated with operating system while Ring 3 is reserved for applications to manage access to the computer hardware. As Ring 0 is used by kernel because of that Ring 0 has the highest-level privilege while Ring 3 has lowest privilege as it belongs to user level application shown in Fig. 2.10.1. The user level applications typically run in Ring 3, the operating system needs to have direct access to the memory and hardware and must execute its privileged instructions in Ring 0.

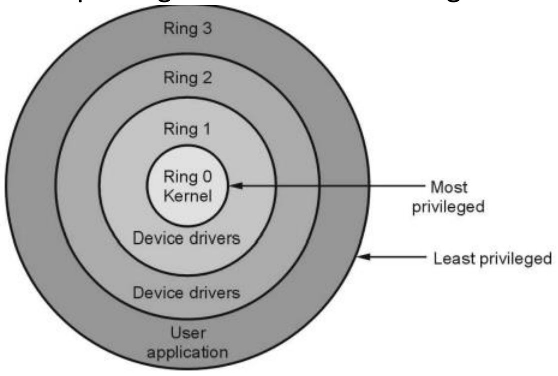


Fig. 2.10.1 CPU Privilege Rings

A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode. Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency. Other critical instructions should be handled carefully for correctness and stability. The critical instructions are divided into three categories: privileged instructions, 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.

A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode. When the privileged instructions including control- and behavior-sensitive instructions of a VM are exe-cuted, they are trapped in the VMM. In this case, the VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system. However, not all CPU architectures are virtualizable. RISC CPU architectures can be naturally virtualized because all control- and behavior-sensitive instructions are privileged instructions. On the contrary, x86 CPU architectures are not primarily designed to support virtualization. This is because about 10 sensitive instructions, such as SGDT and SMSW, are not privileged instructions. When these instruc-tions execute in virtualization, they cannot be trapped in the VMM.

On a native UNIX-like system, a system call triggers the 80h interrupt and passes control to the OS kernel. The interrupt handler in the kernel is then invoked to process the system call. On a para-virtualization system such as Xen, a system call in the guest OS first triggers the 80h interrupt nor-mally. Almost at the same time, the 82h interrupt in the hypervisor is triggered. Incidentally, control is passed on to the hypervisor as well. When the hypervisor completes its task for the guest OS system call, it passes control back to the guest OS kernel. Certainly, the guest OS kernel may also invoke the hypercall while it's running. Although paravirtualization of a CPU lets unmodified applications run in the VM, it causes a small performance penalty.

Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operat-ing 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.

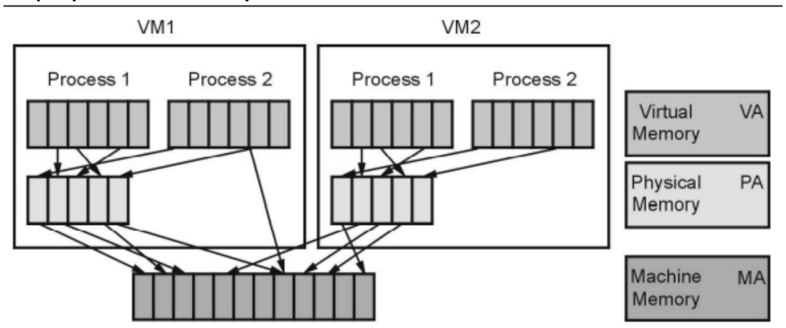


Fig. 2.11.1 Memory Virtualization

That means a two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory. Furthermore, MMU virtualization should be supported, which is transparent to the guest OS. The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs. But the guest OS cannot directly access the actual machine memory. The VMM is responsible for mapping the guest physical memory to the actual machine memory. Figure 3.12 shows the two-level memory mapping procedure.
Since each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the shadow page table. Nested page tables add another layer of indirection to virtual memory. The MMU already handles virtual-to-physical translations as defined by the OS. Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor. Since modern operating systems maintain a set of page tables for every process, the shadow page tables will get flooded. Consequently, the perfor-mance overhead and cost of memory will be very high.
VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation. Processors use TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access. When the guest OS changes the virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup. 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.

I/O Virtualization

I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware. At the time of this writing, there are three ways to implement I/O virtualization: full device emulation, para-virtualization, and direct I/O. Full device emulation is the first approach for I/O virtualization. Generally, this approach emulates well-known, real-world devices.

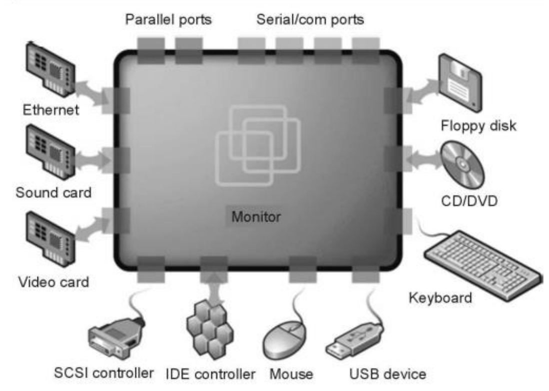


Fig. 2.12.1 Device and I/O virtualization

All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices. The full device emulation approach is shown in Figure 3.14.

A single hardware device can be shared by multiple VMs that run concurrently. However, software emulation runs much slower than the hardware it emulates [10,15]. The para-virtualization method of I/O virtualization is typically used in Xen. It is also known as the split driver model consisting of a frontend driver and a backend driver. The frontend driver is running in Domain U and the backend dri-ver is running in Domain 0. They interact with each other via a block of shared memory. The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs. Although para-I/O-virtualization achieves better device performance than full device emulation, it comes with a higher CPU overhead.

Direct I/O virtualization lets the VM access devices directly. It can achieve close-to-native performance without high CPU costs. However, current direct I/O virtualization implementations focus on networking for mainframes. There are a lot of challenges for commodity hardware devices. For example, when a physical device is reclaimed (required by workload migration) for later reassign-ment, 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) [47]. 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 virtua-lized I/O device, such as virtual network interfaces, virtual block devices (disk), virtual camera devices, and others. The guest OS interacts with the VIFs via VIF device drivers. Each VIF consists of two mes-sage queues. One is for outgoing messages to the devices and the other is for incoming messages from the devices. In addition, each VIF has a unique ID for identifying it in SV-IO

Unit 3

<u>NIST Cloud Computing Reference Architecture</u>

In this section, we will examine and discuss the reference architecture model given by

the National Institute of Standards and Technology (NIST). The model offers approaches for secure cloud adoption while contributing to cloud computing guidelines and standards.
Cloud Computing 3 - 5 Cloud Architecture, Services and Storage
TECHNICAL PUBLICATIONS® - An up thrust for knowledge
The NIST team works closely with leading IT vendors, developers of standards, industries and other governmental agencies and industries at a global level to support effective cloud computing security standards and their further development. It is important to note that this NIST cloud reference architecture does not belong to any specific vendor products, services or some reference implementation, nor does it prevent further innovation in cloud technology.
The NIST reference architecture is shown in Fig. 3.2.1.

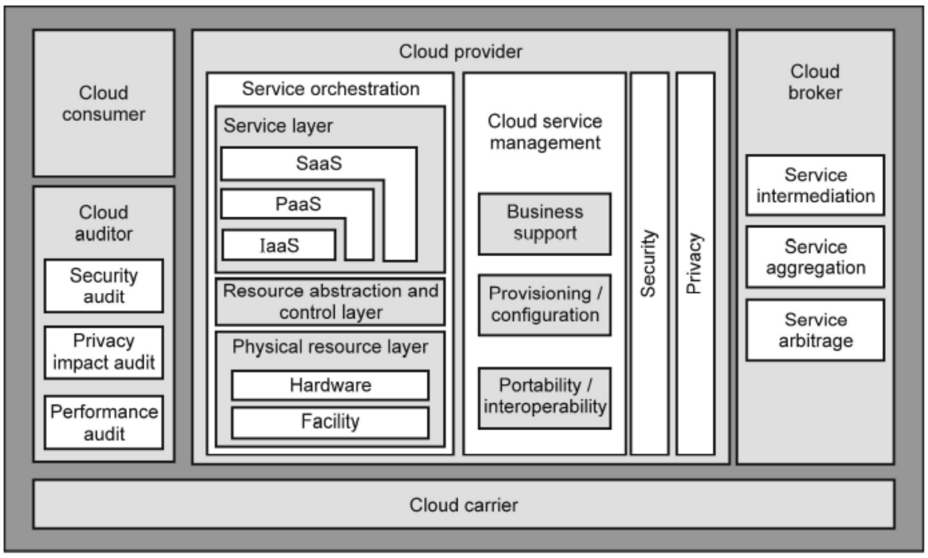


Fig. 3.2.1 : Conceptual cloud reference model showing different actors and entities
From Fig. 3.2.1, note that the cloud reference architecture includes five major actors :
 Cloud consumer
 Cloud provider
 Cloud auditor
 Cloud broker
 Cloud carrier
Each actor is an organization or entity plays an important role in a transaction or a process, or performs some important task in cloud computing. The interactions between these actors are illustrated in Fig. 3.2.2.
Fig. 3.2.2 : Interactions between different actors in a cloud
Now, understand that a cloud consumer can request cloud services directly from a CSP or from a cloud broker. The cloud auditor independently audits and then contacts other actors to gather information. We will now discuss the role of each actor in detail.
<u>Cloud Consumer</u>
A cloud consumer is the most important stakeholder. The cloud service is built to support a cloud consumer. The cloud consumer uses the services from a CSP or person or asks an organization that maintains a business relationship. The consumer then verifies the service catalogue from the cloud provider and requests an appropriate service or sets up service contracts for using the service. The cloud consumer is billed for the service used.
<u>Cloud Provider</u>
Cloud provider is an entity that offers cloud services to interested parties. A cloud

provider manages the infrastructure needed for providing cloud services. The CSP also runs the software to provide services and organizes the service delivery to cloud consumers through networks.

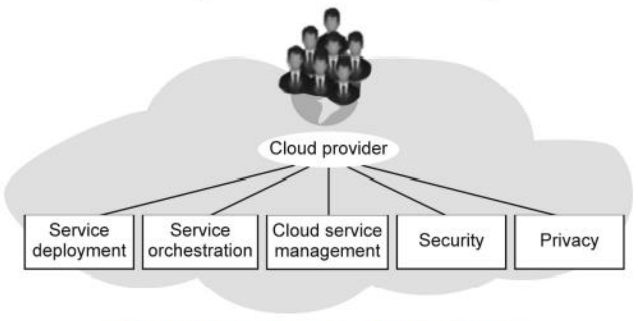


Fig. 3.2.7 : Major activities of a cloud provider

Cloud Auditor

The cloud auditor performs the task of independently evaluating cloud service controls to provide an honest opinion when requested. Cloud audits are done to validate standards conformance by reviewing the objective evidence. The auditor will examine services provided by the cloud provider for its security controls, privacy, performance, and so on.

Cloud Broker

The cloud broker collects service requests from cloud consumers and manages the use, performance, and delivery of cloud services. The cloud broker will also negotiate and manage the relationship between cloud providers and consumers. A cloud broker may provide services that fall into one of the following categories :

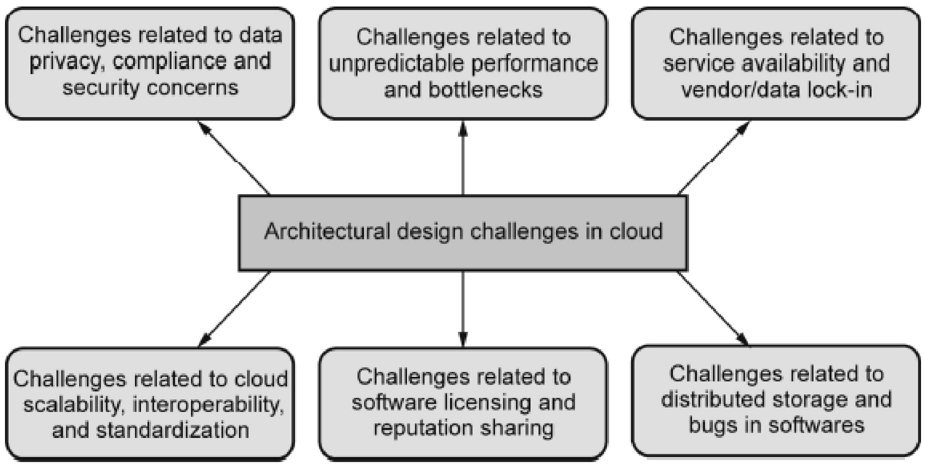
- Service intermediation : Here the cloud broker will improve some specific capabilities, and provide value added services to cloud consumers.
- Service aggregation : The cloud broker links and integrates different services into one or more new services.
- Service Arbitrage : This is similar to aggregation, except for the fact that services that are aggregated are not fixed. In service arbitrage, the broker has the liberty to choose services from different agencies.

3.2.5 Cloud Carrier

The cloud carrier tries to establish connectivity and transports cloud services between a cloud consumer and a cloud provider. Cloud carriers offer network access for consumers, by providing telecommunication links for accessing resources using other devices (laptops, computers, tablets, smartphones, etc.). Usually, a transport agent is an entity offering telecommunication carriers to a business organization to access resources. The cloud provider will set up SLAs with cloud carrier to ensure carrier transport is consistent with the level of SLA provided by the consumers. Cloud carriers provide secure and dedicated high - speed links with cloud providers and between different cloud entities.

Architectural Design Challenges

The cloud architecture design plays an important role in making cloud services successful in all aspects, but still it has some challenges. The major challenges involved in architectural design of cloud computing are shown in Fig. 3.5.1 and explained as follows.



Challenges related to data privacy, compliance and security concerns
Challenges related to unpredictable performance and bottlenecks
Challenges related to service availability and vendor/data lock-in
Architectural design challenges in cloud
Challenges related to cloud scalability, interoperability, and standardization
Challenges related to software licensing and reputation sharing
Challenges related to distributed storage and bugs in softwares

Fig. 3.5.1 : Architectural design challenges in cloud

3.5.1 Challenges related to Data Privacy, Compliance and Security Concerns

Presently, most of the cloud offerings are basically runs on public networks which renders the infrastructure more susceptible to attack. The most common attacks on the network include buffer overflows, DoS attacks, spyware, malware, root kits, trojan horses and worms. With well-known technologies such as encrypted data, virtual LANs and network middleboxes such as firewalls, packet filters etc., many challenges can be solved immediately. Newer attacks may result from hypervisor malware, guest hopping and hijacking or VM rootkits in a cloud environment. Another form of attack on VM migrations is the man-in-the-middle attack. The passive attacks typically steal personal data or passwords while active attacks can exploit data structures in the kernel that will cause significant damage to cloud servers.

To protect from cloud attacks, one could encrypt their data before placing it in a cloud. In many countries, there are laws that allow SaaS providers to keep consumer data and copyrighted material within national boundaries that also called as compliance or regulatory standards. Many countries still do not have laws for compliance; therefore, it is indeed required to check the cloud service providers SLA for executing compliance for services.

3.5.2 Challenges related to Unpredictable Performance and Bottlenecks

In cloud computing, the cloud platform is responsible for deploying and running services on the top of resource pool which has shared hardware from different physical servers. In a production environment, multiple Virtual Machines (VMs) shares the resources with each other like CPU, memory, I/O and network. Whenever I/O devices are shared between VMs, it may generate a big challenge during provisioning due to I/O interfaced between them. It may generate an unpredicted performance and may result into system bottlenecks. The problem becomes wider when such I/O resources are pulled across boundaries of cloud. In such scenarios, the accessibility may become complicated for data placement and transport. To overcome that, data transfer bottlenecks must be removed, bottleneck links must be widened and weak servers in cloud infrastructure should be removed. One solution for this challenge is to improve I / O architectures and operating systems used in physical servers, so that interrupts and I / O channels can be easily virtualized.

3.5.3 Challenges related to Service Availability and Vendor/Data Lock-in

Due to popularity of cloud computing, many organizations run their mission critical or business critical applications on cloud with shared infrastructure provided by cloud service providers. Therefore, any compromise in service availability may result into huge financial loss. Therefore, managing a single enterprise cloud service is often leads to single failure points. The solution related to this challenge is use of multiple cloud

providers. In such case, 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.
In such instances, even if an organization has several data centers located in various geographic regions the multiple cloud service providers can protect their cloud infrastructure and accounting systems and make them available continuously. The use of multiple cloud providers will also provide more protection against failures. Such implementation may ensure the high availability for the organizations. Distributed Denial of Service (DDoS) attacks are another obstacle to availability. Criminals are trying to slash SaaS providers' profits by making their services out of control. Some utility computing services give SaaS providers the ability to use quick scale - ups to protect themselves against DDoS attacks.
In some cases, due the failure of a single company who was providing cloud storages the lock - in concern arises. As well as because of some vendor - lock in solutions of cloud services providers, organizations face difficulties in migrating to new cloud service provider. Therefor to mitigate those challenges related to data lock in and vendor lock in, software stacks can be used to enhance interoperability between various cloud platforms as well as standardize APIs to rescue data loss due to a single company failure. It also supports "surge computing" that has the same technological framework in both public and private clouds and is used to catch additional tasks that cannot be performed efficiently in a private cloud's data center.

3.5.4 Challenges related to Cloud Scalability, Interoperability and Standardization

In cloud computing, pay-as-you-go model refers to utility - based model where bill for storage and the bandwidth of the network are calculated according to the number of bytes used. Depending on the degree of virtualization, computation is different. Google App Engine scales and decreases automatically in response to load increases; users are paid according to the cycles used. Amazon Web Service charges the number of instances used for VM by the hour, even though the computer is idle. The potential here is to scale up and down quickly in response to load variability, to save money, but without breaching SLAs. In virtualization, the Open Virtualization Format (OVF) defines an open, secure, portable, effective and extensible format for VM packaging and delivery. It also specifies a format to be used to distribute the program in VMs. It also specifies a transportation framework for VM templates, which can refer to various virtualization platforms with different virtualization levels.
The use of a different host platform, virtualization platform or guest operating system does not depend on this VM format. The solution is to address virtual platform - agnostic packaging with bundled device certification and credibility. The package provides support for virtual appliances that span more than one VM. The ability of virtual appliances needs to be proposed to operate on any virtual platform in terms of cloud standardization to allow VMs to run hypervisors on heterogeneous hardware platforms. The cloud platform should also introduce live cross - platform migration between x86 Intel and AMD technologies and support legacy load balancing hardware to avoid the challenges related to interoperability.

3.5.5 Challenges related to Software Licensing and Reputation Sharing

Most of the cloud computing providers primarily depended on open source software, as the commercial software licensing model is not suitable for utility computing. The key opportunity is either to stay popular with open source, or simply to encourage commercial software companies to adjust their licensing structure to suit cloud computing better. One may consider using both pay-for-use and bulk licensing schemes to broaden the scope of the company. Bad conduct by one client can affect the credibility of the cloud as a whole. For example, In AWS, spam - prevention services can restrict smooth VM installation by blacklisting of EC2 IP addresses. An advantage would be to build reputation - guarding services similar to those currently provided through "trusted e-mail" providers for providers hosted on smaller ISPs. Another legal issue concerns the transfer of legal responsibility. Cloud services require consumers to remain legally accountable and vice versa. This problem needs to be solved at SLA level.

3.5.6 Challenges related to Distributed Storage and Bugs in Softwares

In cloud applications the database services continuously grow. The potential is to build a storage infrastructure that not only fulfills this growth but also blends it with the cloud benefit of scaling up and down dynamically on demand. That involves the design of efficiently distributed SANs. The data centers will meet the standards of programmers in terms of scalability, system reliability and HA. A major problem in cloud computing is data consistency testing in SAN - connected data centers. Large - scale distributed bugs cannot be replicated, so debugging must take place on a scale in the data centers for production. Hardly any data center will deliver that convenience. One solution may be to focus on using VMs in cloud computing. The virtualization level can allow valuable information to be captured in ways that are impossible without using VMs. Debugging on simulators is another way to fix the problem, if the simulator is well designed.

Identify and access management (IAM).

The Identity and Access Management (IAM) are the vital function for every organisations, and SaaS customers have a fundamental expectation that their data is given the principle of the least privilege. The privilege principle says that only the minimum access is required to perform an operation that should be granted access only for the minimum amount of time required. Aspects of current models including trust principles, privacy implications and operational aspects of authentication and authorization, are challenged within the cloud environment in which services delivered on demand and can continuously evolve. To meet these challenges, SaaS providers need to align their efforts by testing new models and management processes for IAM to include end-to-end trust and identity across the cloud and their enterprises. The balance between usability and security will be seen as an additional issue. When a good balance is achieved, obstacles to the successful completion of their support and maintenance activities will impact both businesses and their groups.

As cloud composed of many services deployed on big infrastructure because of that it requires multiple security mechanisms to protect it from failure.

The Identity and access management is the security framework composed of policy and governance components used for creation, maintenance and termination of digital identities with controlled access of shared resources. It composed of multiple processes, components, services and standard practices. It focuses on two parts namely Identity management and access management. The directory services are used in IAM for creating a repository for identity management, authentication and access management. The IAM provides many features like user management, authentication management, authorization management, Credential and attribute management, Compliance management, Monitoring and auditing etc. The lifecycle of identity management is shown in Fig. 4.9.1.

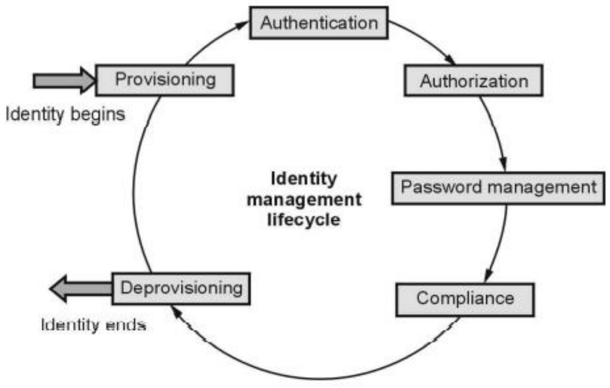


Fig. 4.9.1 Lifecycle of Identity Management

Fig. 4.9.1 Lifecycle of Identity Management
The IAM architecture is made up of several processes and activities (see Fig. 4.9.2). The processes supported by IAM are given as follows.
a) User management - It provides processes for managing the identity of different entities.
b) Authentication management - It provides activities for management of the process for determining that an entity is who or what it claims to be.
c) Access management - It provides policies for access control in response to request for resource by entity.
d) Data management - It provides activities for propagation of data for authorization to resources using automated processes.
e) Authorization management - It provides activities for determining the rights associated with entities and decide what resources an entity is permitted to access in accordance with the organization's policies.
f) Monitoring and auditing - Based on the defined policies, it provides monitoring, auditing, and reporting compliance by users regarding access to resources.
The activities supported by IAM are given as follows.
a) Provisioning - The provisioning has essential processes that provide users with necessary access to data and resources. It supports management of all user account operations like add, modify, suspend, and delete users with password management. By provisioning the users are given access to data, systems, applications, and databases based on a unique user identity. The deprovisioning does the reverse of provisioning which deactivate of delete the users identity with privileges.
b) Credential and attribute management - The Credential and attribute management prevents identity impersonation and inappropriate account use. It deals with management of credentials and user attributes such as create, issue, manage and revoke users to minimize the business risk associated with it. The individuals' credentials are verified during the authentication process. The Credential and attribute management processes include provisioning of static or dynamic attributes that comply with a password standard, encryption management of credentials and handling access policies for user attributes.
c) Compliance management - The Compliance management is the process used for monitoring the access rights and privileges and tracked to ensure the security of an enterprise's resources. It also helpful to auditors to verify the compliance to various access control policies, and standards. It includes practices like access
monitoring, periodic auditing, and reporting.
d) Identity federation management - Identity federation management is the process of managing the trust relationships beyond the network boundaries where organizations come together to exchange the information about their users and
entities.

e) Entitlement management - In IAM, entitlements are nothing but authorization policies. The Entitlement management provides processes for provisioning and deprovisioning of privileges needed for the users to access the resources including systems, applications, and databases.

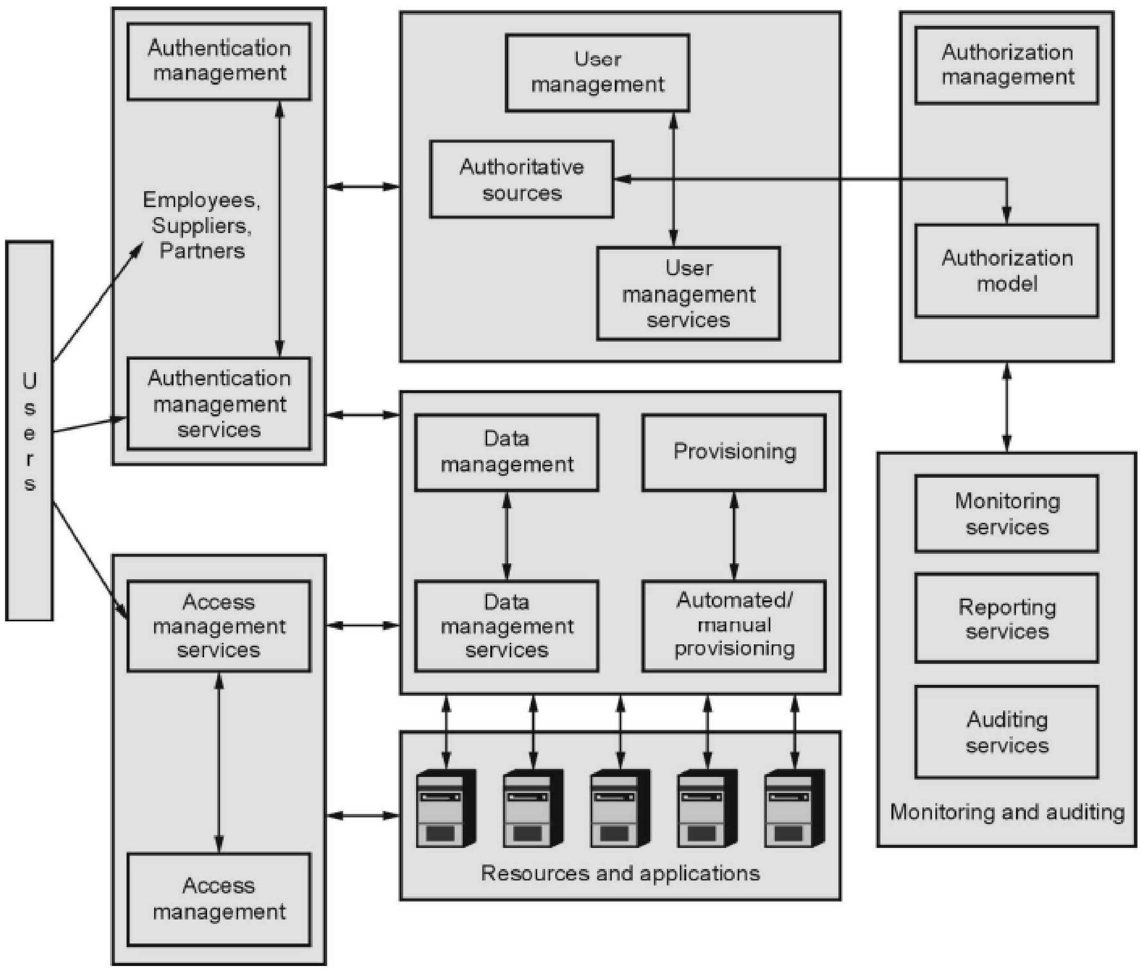


Fig.4.9.2 IAM Architecture

Hadoop

With the evolution of internet and related technologies, the high computational power, large volumes of data storage and faster data processing becomes the basic need for most of the organizations and it has been significantly increased over the period of time. Currently, organizations are producing huge amount of data at faster rate. The recent survey on data generation by various organization says that Facebook produces roughly 600+ TBs of data per day and analyzes 30+ Petabytes of user generated data, Boeing jet airplane generates more than 10 TBs of data per flight including geo maps, special images and other information, Walmart handles more than 1 million customer transactions every hour, which is imported into databases estimated to contain more than 2.5 petabytes of data.

So, there is a need to acquire, analyze, process, handle and store such a huge amount of data called big data. The different challenges associated with such big data are given as below

a) Volume : The Volume is related to Size of big data. The amount of data growing day by day is very huge. According to IBM, in the year 2000, 8 lakh petabytes of data were stored in the world.so challenge here is, how to deal with such huge Big

Data.
b) Variety : The Variety is related to different formats of big data. Nowadays most of the data stored by organizations have no proper structure called unstructured data. Such data has complex structure and cannot be represented using rows and columns. The challenge here is how to store different formats of data in databases.
c) Velocity : The Velocity is related to speed of data generation which is very fast. It is a rate at which data is captured, generated and shared. The challenge here is how to react to massive information generated in the time required by the application.
d) Veracity : The Veracity refers to uncertainty of data. The data stored in database sometimes is not accurate or consistent that makes poor data quality. The inconsistent data requires lot of efforts to process such data.
The traditional database management techniques are incapable to satisfy above four characteristics as well as doesn’t supports storing, processing, handling and analyzing big data. Therefore, these challenges associated with Big data can be solved using one of the most popular framework provided by Apache is called Hadoop.
The Apache Hadoop is an open source software project that enables distributed processing of large data sets across clusters of commodity servers using programming models. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance. It is a software framework for running the applications on clusters of commodity hardware with massive storage, enormous processing power and supporting limitless concurrent tasks or jobs.
The Hadoop core is divided into two fundamental components called HDFS and MapReduce engine. The HDFS is a distributed file system inspired by GFS that organizes files and stores their data on a distributed computing system, while MapReduce is the computation engine running on top of HDFS as its data storage manager. The working of HDFS and MapReduce are explained followed by Hadoop Ecosystem Components in further sections.
Hadoop Distributed File system (HDFS)
The Hadoop Distributed File system (HDFS) is the Hadoop implementation of distributed file system design that hold large amount of data. It provide easier access to stored data to many clients distributed across the network. It is highly fault tolerant and designed to be run - on low - cost hardware (called commodity hardware). The files in HDFS are stored across the multiple machine in redundant fashion to recover the data loss in case of failure.
It enables storage and management of large files stored on distributed storage medium over the pool of data node. A single name node runs in a cluster is associated with multiple data nodes that provide the management of hierarchical file organization and namespace. The HDFS file composed of fixed size blocks or chunks that are stored on data nodes. The name node is responsible for storing the metadata about each file that includes attributes of files like type of file, size, date and time of creation, properties of the files as well as the mapping of blocks to files at the data nodes. The data node treats each data block as a separate file and propagates the critical information with the name node. The HDFS provides fault tolerance through data replication that can be specified at the time of file creation with attribute name degree of replication (i.e., the number of copies made) which is progressively significant in bigger environments consisting of many racks of data servers. The significant benefits provided by HDFS are given as follows
 It provides streaming access to file system data
 It is suitable for distributed storage and processing
 It is optimized to support high streaming read operations with limited set.
 It supports file operations like read, write, delete but append not update.
 It provides Java APIs and command line command line interfaces to interact with

HDFS.
 It provides different File permissions and authentications for files on HDFS.
 It provides continuous monitoring of name nodes and data nodes based on continuous “heartbeat” communication between the data nodes to the name node.
 It provides Rebalancing of data nodes so as to equalize the load by migrating blocks of data from one data node to another.
 It uses checksums and digital signatures to manage the integrity of data stored in a file.
 It has built-in metadata replication so as to recover data during the failure or to protect against corruption.
 It also provides synchronous snapshots to facilitates rolled back during failure.

5.2.1 Architecture of HDFS

The HDFS follows Master-slave architecture using name node and data nodes. The name node act as a master while multiple data nodes worked as slaves. The HDFS is implemented as block structure file system where files are broken in to block of fixed size stored on Hadoop clusters. The HDFS architecture is shown in Fig. 5.2.1.

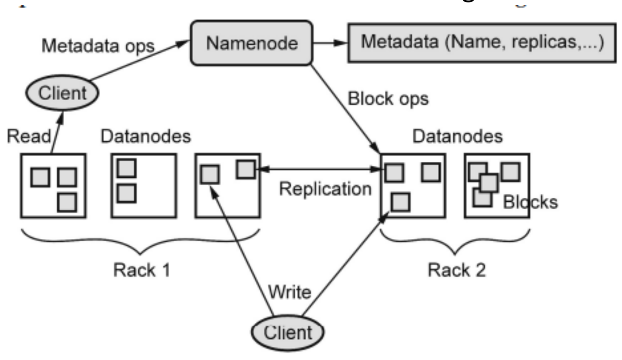


Fig. 5.2.1 : HDFS architecture

The Components of HDFS composed of following elements

1. Name Node

An HDFS cluster consists of single name node called master server that manages the file system namespaces and regulate access to files by client. It runs on commodity hardware that manages file system namespaces. It stores all metadata for the file system across the clusters. The name node serves as single arbitrator and repository for HDFS metadata which is kept in main memory for faster random access. The entire file system name space is contained in a file called FsImage stored on name nodes file system, while the transaction log record is stored in Editlog file.

2. Data Node

In HDFS there are multiple data nodes exist that manages storages attached to the node that they run on. They are usually used to store users’ data on HDFS clusters. Internally the file is splitted in to one or more blocks to data node. The data nodes are responsible for handling read/write request from clients. It also performs block creation, deletion and replication upon instruction from name node. The data node stores each HDFS data block in separate file and several blocks are stored on different data nodes. The requirement of such a block structured file system is to store, manage and access files metadata reliably.

The representation of name node and data node is shown in Fig. 5.2.2.

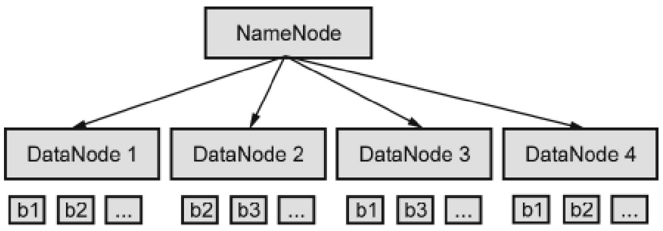


Fig. 5.2.2 : Representation of name node and data nodes

3. HDFS Client

In Hadoop distributed file system, the user applications access the file system using the HDFS client. Like any other file systems, HDFS supports various operations to read, write and delete files, and operations to create and delete directories. The user references files and directories by paths in the namespace. The user application does not need to aware that file system metadata and storage are on different servers, or that blocks have multiple replicas. When an application reads a file, the HDFS client first asks the name node for the list of data nodes that host replicas of the blocks of the file. The client contacts a data node directly and requests the transfer of the desired block. When a client writes, it first asks the name node to choose data nodes to host replicas of the first block of the file. The client organizes a pipeline from node-to-node and sends the data. When the first block is filled, the client requests new data nodes to be chosen to host replicas of the next block. The Choice of data nodes for each block is likely to be different.

4. HDFS Blocks

In general, the user's data stored in HDFS in terms of block. The files in file system are divided in to one or more segments called blocks. The default size of HDFS block is 64 MB that can be increase as per need.

The HDFS is fault tolerance such that if data node fails then current block write operation on data node is re-replicated to some other node. The block size, number of replicas and replication factors are specified in Hadoop configuration file. The synchronization between name node and data node is done by heartbeats functions which are periodically generated by data node to name node.

Apart from above components the job tracker and task trackers are used when MapReduce application runs over the HDFS. Hadoop core consists of one master job tracker and several task trackers. The job tracker runs on name node like a master while task trackers runs on data nodes like slaves.

The job tracker is responsible for taking the requests from a client and assigning task trackers to it with tasks to be performed. The job tracker always tries to assign tasks to the task tracker on the data nodes where the data is locally present. If for some reason the node fails the job tracker assigns the task to another task tracker where the replica of the data exists since the data blocks are replicated across the data nodes. This ensures that the job does not fail even if a node fails within the cluster.