CBCS SCHEME

USN | 1 | C | R | 2 | 0 | M | C | 0 | 2 | 9 |

20MCA342

Third Semester MCA Degree Examination, Feb./Mar. 2022
Cloud Computing

Time: 3 hrs. Max. Marks: 100

Note: *Answer any FIVE full questions, choosing ONE full question from each module.*

Module-1

1 a. Explain with suitable diagram the Cloud Computing reference model. (10 Marks)
b. Discuss in detail the major deployment models for Cloud Computing. (10 Marks)

OR

2 a. What is Cloud Computing? Explain its characteristics and benefits. (10 Marks)
b. Discuss about the milestones which have led to Cloud Computing. (10 Marks)

Module-2

3 a. Explain with suitable diagram, Virtual Machine Architecture. (10 Marks)
b. What are the most popular and important messages passing techniques? (10 Marks)

OR

4 a. Discuss in detail about Call and Return Architectures. (10 Marks)
b. Explain with suitable diagram , the RPC reference model. (10 Marks)

Module-3

5 a. What are the functions enabled by managed execution? Explain. (10 Marks)
b. Explain with neat diagram, the machine reference model. (10 Marks)

OR

6 a. Discuss in detail about the characteristics of Virtualized solutions. (10 Marks)
b. Explain in detail the Popek and Goldberg theorems in Virtualization. (10 Marks)

Module-4

7 a. Explain with suitable diagram, the Cloud Computing Architecture. (10 Marks)
b. What are the Open challenges in Cloud Computing? Explain. (10 Marks)

OR

8 a. Discuss Platform – as – a – Service reference model. (10 Marks)
b. With suitable example, explain in detail the types of Clouds. (10 Marks)

Module-5

9 a. What are the different scientific applications in Cloud Computing? (10 Marks)
b. Explain in detail about Dropbox and Icloud. (10 Marks)

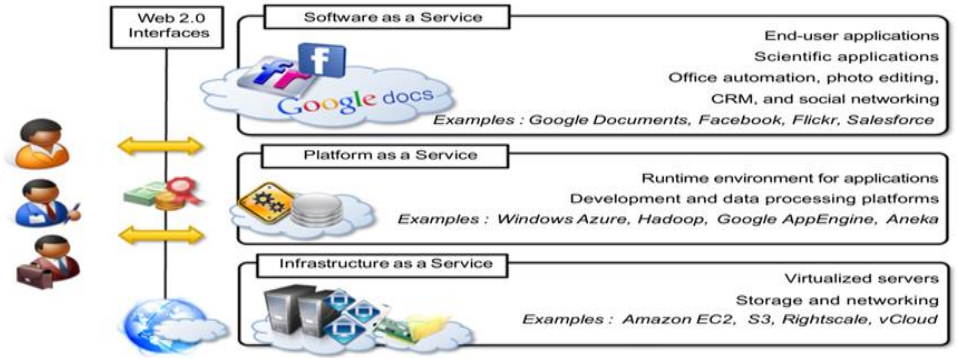
OR

10 a. Explain in detail the Business and Consumer application of Cloud Computing. (10 Marks)
b. Discuss about media applications of Cloud Computing. (10 Marks)

* * * * *

1. a. Explain with suitable diagram the cloud computing reference model. (10)

- A fundamental characteristic of cloud computing is the capability to deliver, on demand, a variety of IT services that are quite diverse from each other.
- three major categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).



i. Infrastructure as a Service (IaaS):

- deliver infrastructure on demand in the form of virtual hardware, storage, and networking
- Virtual hardware is utilized to provide compute on demand in the form of virtual machine instances.
- These are created at users' request on the provider's infrastructure
- users are given tools and interfaces to configure the software stack installed in the virtual machine.
- The pricing model is usually defined in terms of dollars per hour
- Virtual storage is delivered in the form of raw disk space or object store.
- Example: Amazon Elastic Compute Cloud (Amazon EC2), DigitalOcean

ii. Platform as a Service (PaaS):

- deliver scalable and elastic runtime environments on demand and host the execution of applications.
- These services are backed by a core middleware platform that is responsible for creating the abstract environment where applications are deployed and executed.
- users are requested to focus on the logic of the application developed by leveraging the provider's APIs and libraries
- increases the level of abstraction at which cloud computing is leveraged but also constrains the user in a more controlled environment.
- Example: Microsoft Azure, force.com, Magento commerce cloud

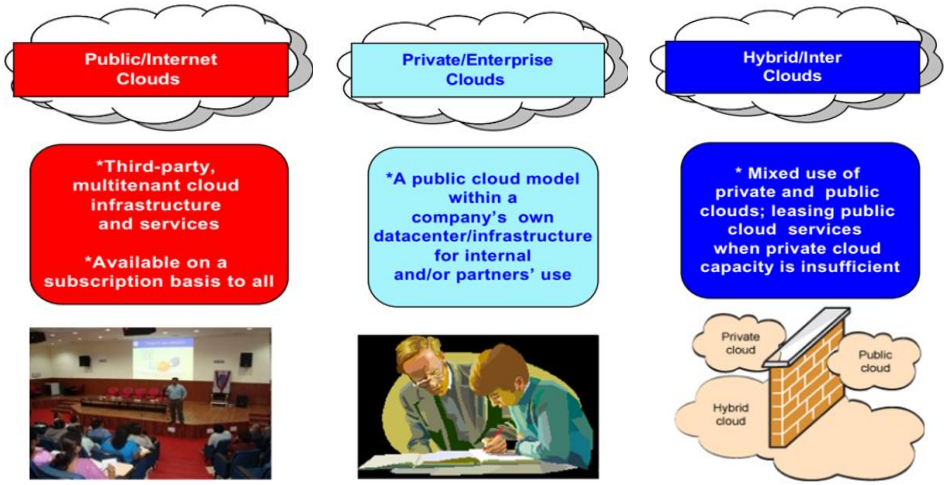
iii. Software as a Service (SaaS):

- provide (desktop) applications such as office automation, photo editing and services on demand
- These applications are shared across multiple users whose interaction is isolated from the other users.
- Example: Google Apps, DropBox

- IaaS solutions: for users who want to leverage cloud computing from building dynamically scalable computing systems (website development)
- PaaS solutions: provide scalable programming platforms for developing applications and are more appropriate when new systems have to be developed.
- SaaS solutions: for users who want to use cloud without using any software development, installation, configuration, and maintenance.

1. b. Discuss in detail the major deployment models for cloud computing. (10)

- three major models for deploying and accessing cloud computing environments: (i) Public Clouds (ii) Private/ Enterprise Clouds and (iii) Hybrid Clouds



i. Public clouds: necessary IT infrastructure is established by a third-party service provider that makes it available to any consumer on a subscription basis.

ii. Private clouds: Large organizations that own massive computing infrastructures can still benefit from cloud computing by replicating the cloud IT service delivery model in-house

iii. Hybrid clouds: partially composed of public cloud resources and privately-owned infrastructures

2. a. What is cloud computing? Explain its characteristics and benefits. (10)

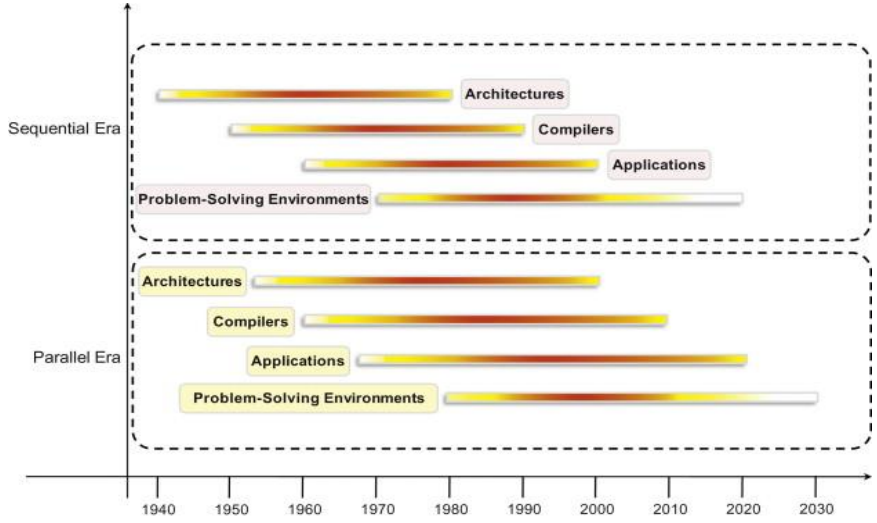
- Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
- Cloud computing has some interesting characteristics that bring benefits to both cloud service consumers (CSCs) and cloud service providers (CSPs). These characteristics are:

- No up-front commitments
- On-demand access
- Nice pricing
- Simplified application acceleration and scalability
- Efficient resource allocation
- Energy efficiency
- Seamless creation and use of third-party services
- The most evident benefit from the use of cloud computing systems and technologies is the increased economical return due to the reduced maintenance costs and operational costs related to IT software and infrastructure.
- Location-independent resource pooling
- Rapid elasticity
- Measured service
- Ubiquitous network access
- On-demand self-service

- maintenance costs are significantly reduced: by renting the infrastructure and the application services, organizations are no longer responsible for their maintenance.

2. b. Discuss about the milestones which have led to cloud computing. (10)

- The four key elements of computing developed during these eras are:
  - Architectures
  - Compilers
  - Applications
  - problem-solving environments
- every aspect of this era underwent a three-phase process:
  - research and development (R&D)
  - commercialization, and
  - commoditization



3. a. Explain with suitable diagram, Virtual Machine Architecture. (10)

- The virtual machine class of architectural styles is characterized by the presence of an abstract execution environment (generally referred as a virtual machine) that simulates features that are not available in the hardware or software.
- The general interaction flow for systems implementing this pattern is the following: the program (or the application) defines its operations and state in an abstract format, which is interpreted by the virtual machine engine.
  - Rule-Based Style: This architecture is characterized by representing the abstract execution environment as an inference engine. The input data for applications is generally represented by a set of assertions or facts that the inference engine uses to activate rules or to apply predicates, thus transforming data. The output can either be the product of the rule activation or a set of assertions that holds true for the given input data. The set of rules or predicates identifies the knowledge base that can be queried to infer properties about the system.
  - Interpreter Style: The core feature of the interpreter style is the presence of an engine that is used to interpret a pseudo-program expressed in a format acceptable for the interpreter. The interpretation of the pseudo-program constitutes the execution of the program itself.
- Virtual machine architectural styles are characterized by an indirection layer between applications and the hosting environment. This design has the major advantage of decoupling applications from the underlying hardware and software environment, but at the same time it introduces some disadvantages, such as a slowdown in performance.

3. b. What are the most popular and important messages passing techniques? (10)

- **Message passing**: This paradigm 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 (RPC)**: This 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. Messages, automatically created by the RPC implementation, convey the information about the procedure to execute along with the required parameters and the return values. The use of messages within this context is also referred as marshaling of parameters and return values.
- **Distributed objects:** This is an implementation of the RPC model for the object-oriented paradigm and contextualizes this feature for the remote invocation of methods exposed by objects. Each process registers a set of interfaces that are accessible remotely. Client processes can request a pointer to these interfaces and invoke the methods available through them. The underlying runtime infrastructure is in charge of transforming the local method invocation into a request to a remote process and collecting the result of the execution. The communication between the caller and the remote process is made through messages. With respect to the RPC model that is stateless by design, distributed object models introduce the complexity of object state management and lifetime. The methods that are remotely executed operate within the context of an instance, which may be created for the sole execution of the method, exist for a limited interval of time, or are independent from the existence of requests. Examples of distributed object infrastructures are Common Object Request Broker Architecture (CORBA), Component Object Model (COM, DCOM, and COM1), Java Remote Method Invocation (RMI), and .NET Remoting.
- **Distributed agents and active objects**: Programming paradigms based on agents and active objects involve by definition the presence of instances, whether they are agents of objects, despite the existence of requests. This means that objects have their own control thread, which allows them to carry out their activity. These models often make explicit use of messages to trigger the execution of methods, and a more complex semantics is attached to the messages.
- **Web services**: Web service technology provides an implementation of the RPC concept over HTTP, thus allowing the interaction of components that are developed with different technologies. A Web service is exposed as a remote object hosted on a Web server, and method invocations are transformed in HTTP requests, opportunely packaged using specific protocols such as Simple Object Access Protocol (SOAP) or Representational State Transfer (REST).

4. a. Discuss in detail about call and return architectures. (10)

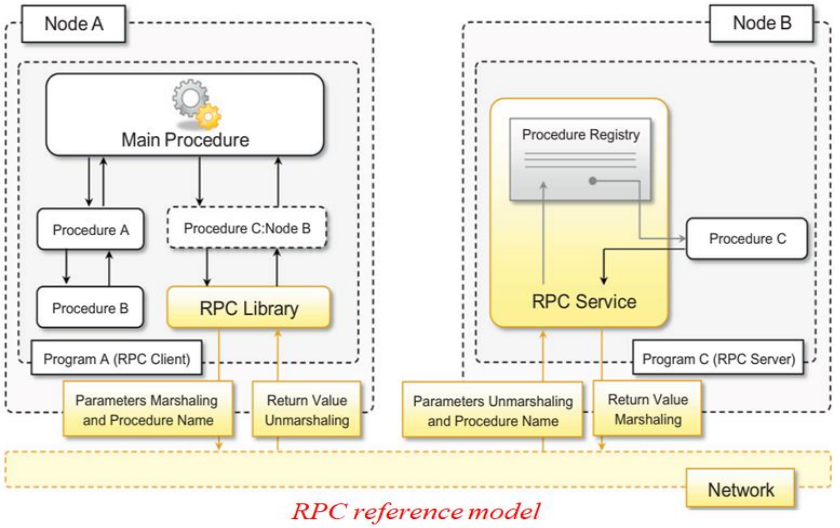
- This category identifies all systems that are organised into components mostly connected together by method calls. The activity of systems modeled in this way is characterized by a chain of method calls whose overall execution and composition identify the execution of one or more operations. The internal organization of components and their connections may vary.
  - **Top-Down Style**: This architectural style is quite representative of systems developed with imperative programming, which leads to a divide-and-conquer

approach to problem resolution. Systems developed according to this style are composed of one large main program that accomplishes its tasks by invoking subprograms or procedures. The components in this style are procedures and subprograms, and connections are method calls or invocation. The calling program passes information with parameters and receives data from return values or parameters. Method calls can also extend beyond the boundary of a single process by leveraging techniques for remote method invocation, such as remote procedure call (RPC) and all its descendants.

- **Object-Oriented Style**: This architectural style encompasses a wide range of systems that have been designed and implemented by leveraging the abstractions of object-oriented programming (OOP). Systems are specified in terms of classes and implemented in terms of objects. Classes define the type of components by specifying the data that represent their state and the operations that can be done over these data.
- **Layered Style**: 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. Specific protocols and interfaces define how adjacent layers interact. It is possible to model such systems as a stack of layers, one for each level of abstraction.

4. b. Explain with suitable diagram, the RPC reference model. (10)

- fundamental abstraction enabling the execution of procedures on client's request.
- 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
- Even though it is a quite old technology, RPC is still used today as a fundamental component for IPC in more complex systems.



RPC reference model

- ***Marshaling***: identifies the process of converting parameter and return values into a form that is more suitable to be transported over a network through a sequence of bytes. [packing of arguments(or parameters) into a message packet]
- ***Unmarshaling***: the opposite procedure. [unpacking of arguments(or parameters) received from the call packet].
- The RPC runtime is also for handling the request-reply interaction that happens between the client and the server process in a completely transparent manner.

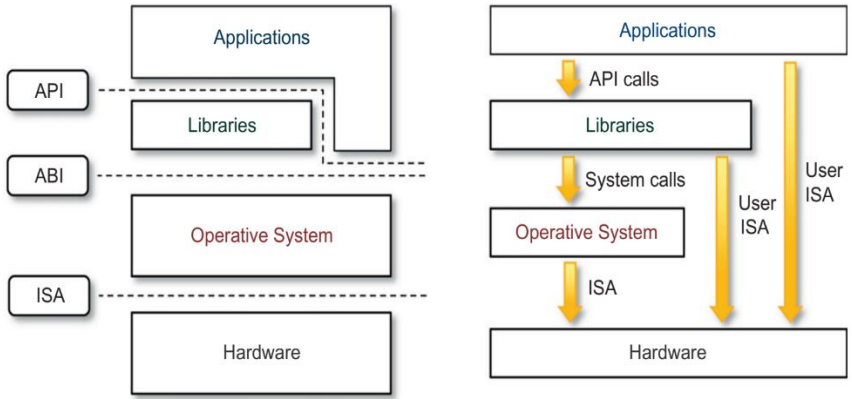
- RPC for IPC consists of the following steps:
  - Design and implementation of the server procedures that will be exposed for remote invocation.
  - Registration of remote procedures with the RPC server on the node where they will be made available.
  - Design and implementation of the client code that invokes the remote procedure(s)

5. a. What are the functions enabled by managed execution? (10) Explain.

- 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 environments 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. This function is naturally implemented in middleware for distributed computing.
  - **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. Isolation brings several benefits; for example, 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. The virtual machine can filter the activity of the guest and prevent harmful operations against the host.

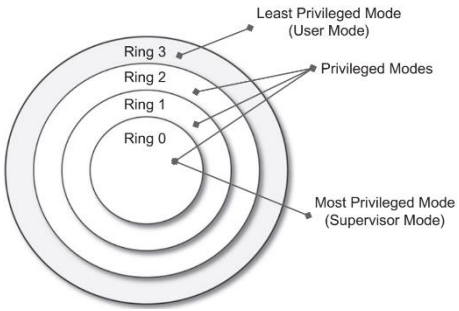
5. b. Explain with neat diagram, the machine reference model. (10)

- Virtualizing an execution environment at different levels of the computing stack requires a reference model that defines the interfaces between the levels of abstractions, which hide implementation details. From this perspective, virtualization techniques actually replace one of the layers and intercept the calls that are directed toward it.



- 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 interrupt management. ISA is the interface between hardware and software, and it 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, and call conventions and defines a format for executable programs. 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/or the underlying operating system.
- For any operation to be performed in the application level API, ABI and ISA are responsible for making it happen. The high-level abstraction is converted into machine-level instructions to perform the actual operations supported by the processor. The machine-level resources, such as processor registers and main memory capacities, are used to perform the operation at the hardware level of the central processing unit (CPU).
- a possible implementation features a hierarchy of privileges 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.



6. a. Discuss in detail about the characteristics of virtualised solutions. **(10)**

(i) 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.
- All the operations of the guest are generally performed against the virtual machine
- Resources exposed by the host can then be hidden or simply protected from the guest
- sensitive information that is contained in the host can be naturally hidden without the need to install complex security policies.
- Increased security is a requirement when dealing with untrusted code (Example: applets downloaded from the Internet run in a sandboxed version of JVM)

(ii) Managed Execution:

- Virtualization of the execution environment not only allows increased security, but a wider range of features also can be implemented like sharing, aggregation, emulation, and isolation are the most relevant features
  - Sharing: a particularly important feature in virtualized data centers, used to reduce the number of active servers and limit power consumption.
  - Aggregation: A group of separate hosts can be tied together and represented to guests as a single virtual host
  - Emulation: a completely different environment with respect to the host can be emulated, thus allowing the execution of guest programs requiring specific characteristics that are not present in the physical host

- Isolation: Virtualization allows providing guests 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
- Advantages of Isolation:
- it provides a separation between the host and the guest
- it allows multiple guests to run on the same host without interfering with each other

(iii) Portability

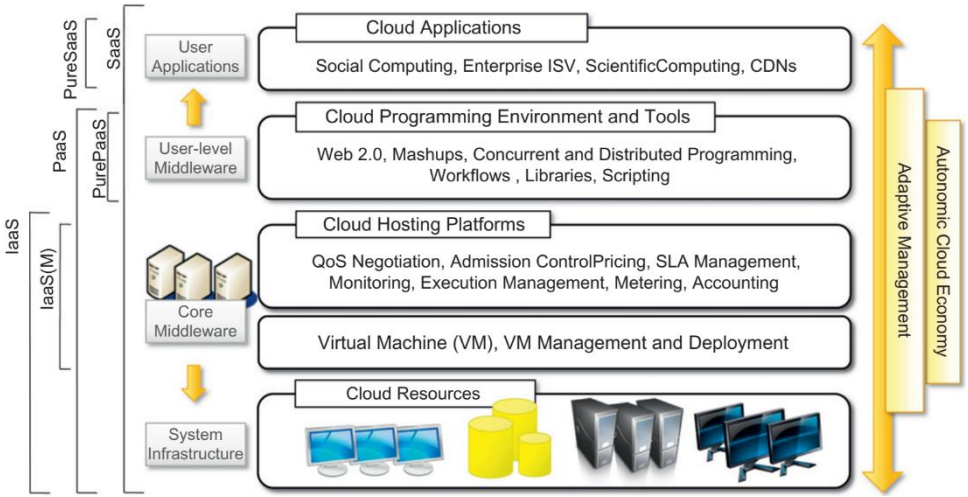
- Portability allows having your own system always with you and ready to use as long as the required virtual machine manager is available
- the guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines
- Virtual images are generally proprietary formats that require a specific virtual machine manager to be executed
- In the case of programming-level virtualization, the binary code representing application components (jars or assemblies) can be run without any recompilation on any implementation of the corresponding virtual machine

6. b. Explain in detail the Popek and Goldberg theorems in Virtualisation. (10)

- Theorem-1: For any conventional third-generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.
- Theorem 2: A conventional third-generation computer is recursively virtualizable if
- It is virtualizable and.
- A VMM without any timing dependencies can be constructed for it.
- Theorem 3: A hybrid VMM may be constructed for any conventional third generation machine, in which the set of user sensitive instructions are a subset of the set of privileged instructions.

7. a. Explain with suitable diagram, the cloud computing architecture. (10)

- It is possible to organize all the concrete realizations of cloud computing into a layered view covering the entire stack from hardware appliances to software systems
- Cloud resources are harnessed to offer “computing horsepower” required for providing services
- Cloud infrastructure can be heterogeneous in nature because a variety of resources, such as clusters and even networked PCs, database systems and other storage services can be used to build it.
- core middleware: provide an appropriate runtime environment for applications and to best utilize resources
- virtualization technologies: used to guarantee runtime environment customization, application isolation, sandboxing, and Quality of Service (QoS)
- Hypervisors: manage the pool of resources and expose the distributed infrastructure as a collection of virtual machines
- virtual machine technology: partition the hardware resources such as CPU and memory and to virtualize specific devices, thus meeting the requirements of users and applications



- Infrastructure-as-a-Service (IaaS): The combination of cloud hosting platforms and resources
- IaaS solutions are suitable for designing the system infrastructure
- IaaS (M): the management layer, often integrated with other IaaS solutions that provide physical infrastructure and adds value to them.
- service is provided by cloud programming environments and tools include Web-based interfaces, command-line tools, and frameworks for concurrent and distributed programming
- users develop their applications specifically for the cloud by using the API exposed at the user-level middleware
- Platform-as-a-Service (PaaS): the service offered to the user is a development platform rather than an infrastructure
- Pure PaaS: only the user-level middleware is offered, and it has to be complemented with a virtual or physical infrastructure
- Software-as-a-Service (SaaS): The top layer of the reference model contains services delivered at the application level
- SaaS: Web-based applications that rely on the cloud to provide service to end users
- The horsepower of the cloud provided by IaaS and PaaS solutions allows independent software vendors to deliver their application services over the Internet.
- SaaS implementations should feature autonomic behavior (elastically scaling on demand) automatically
- PaaS and IaaS generally provide this functionality as a part of the API exposed to users
- Everything as a Service (XaaS): 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 where PaaS solutions are deployed
- cloud computing is an interesting option for reducing startups' capital investment in IT, allowing them to quickly commercialize their ideas and grow their infrastructure according to their revenues.

7. b. What are the open challenges in cloud computing? Explain. (10)

(i) Cloud interoperability and standards

- introducing standards and allowing interoperability between solutions offered by different vendors are objectives of fundamental importance
- Vendor lock-in constitutes one of the major strategic barriers against the seamless adoption of cloud computing at all stages
- Yet the first steps toward a standardization process have been made, and a few organizations:
  - the Cloud Computing Interoperability Forum,

- the Open Cloud Consortium,
- the DMTF Cloud Standards Incubator,
- Open Cloud Manifesto - which embodies the point of view of various stakeholders on the benefits of open standards in the field
- The Open Virtualization Format (OVF) - provide a common format for storing the information and metadata describing a virtual machine image

- the lack of a common set of APIs make the interaction with cloud-based solutions vendor specific

(ii) Scalability and fault tolerance

- Clouds allow scaling beyond the limits of the existing in-house IT resources, whether they are infrastructure (compute and storage) or applications services
- the ability to tolerate failure becomes fundamental, sometimes even more important than providing an extremely efficient and optimized system

(iii) Security, trust, and privacy

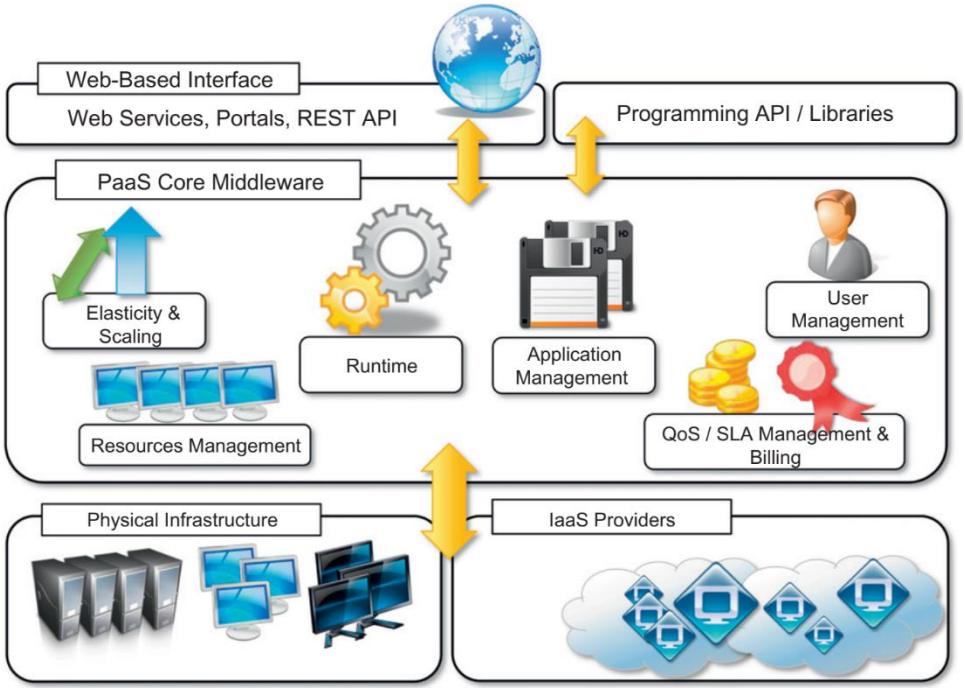
- The traditional cryptographic technologies are used to prevent data tampering and access to sensitive information.
- The massive use of virtualization technologies exposes the existing system to new threats, which previously were not considered applicable
- a lack of control over the environment in which the application is executed, which is made possible by leveraging the cloud

(iv) Organizational aspects

- Cloud computing introduces a significant change in the way IT services are consumed and managed
- storage, compute power, network infrastructure, and applications are delivered as metered services over the Internet
- a billing model that is new within typical enterprise IT departments, which requires a certain level of cultural and organizational process maturity
- a wide acceptance of cloud computing will require a significant change to business processes and organizational boundaries

8. a. Discuss platform-as-a-service reference model. (10)

- provides a development and deployment platform for running applications in the cloud.
- They constitute the middleware on top of which applications are built.



- 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.
- From a user point of view, the core middleware exposes interfaces that allow programming and deploying applications on the cloud.
- These can be in the form of a Web-based interface or in the form of programming APIs and libraries
- Some implementations provide a completely Web-based interface hosted in the cloud and offering a variety of services
- PaaS solutions can offer middleware for developing applications together with the infrastructure (middleware + infrastructure[1]) or simply provide users with the software that is installed on the user premises (infrastructure[2])

1 - the PaaS provider also owns large data-centers where applications are executed
2 - the middleware constitutes the core value of the offering (PurePaaS)

- **PaaS-I:**
  - completely follow the cloud computing style for application development and deployment
  - offer an Integrated Development Environment (IDE) hosted within the Web browser where applications are designed, developed, composed, and deployed
- **PaaS-II:**
  - Focused on providing a scalable infrastructure for Web application, mostly websites
  - Developers generally use the providers' APIs to develop applications
- **PaaS-III**: provide a cloud programming platform for any kind of application, not only Web applications
- The PaaS umbrella encompasses a variety of solutions for developing and hosting applications in the cloud
- essential characteristics that identify a PaaS solution:
  - Runtime framework: represents the "software stack" of the PaaS model; executes end-user code according to the policies set by the user and the provider
  - Abstraction: PaaS solutions offer a way to deploy and manage applications on the cloud rather than a bunch of virtual machines on top of which the IT infrastructure is built and configured
  - Automation: automate the process of deploying applications to the infrastructure, scaling them by provisioning additional resources when needed, according to the SLA made between the customers and the provider
  - Cloud services: provide developers and architects with services and APIs helping them to simplify the creation and delivery of elastic and highly available cloud applications, which includes specific components for developing applications, advanced services for application monitoring, management, and reporting
  - Another essential component for a PaaS-based approach is the ability to integrate third-party cloud services offered from other vendors by leveraging service-oriented architecture, happen through interfaces and protocols
  - PaaS environments deliver a platform for developing applications, which exposes a well-defined set of APIs and, in most cases, binds the application to the specific runtime of the PaaS provider
  - PaaS solutions can cut the cost across development, deployment, and management of applications
  - It helps management reduce the risk of ever-changing technologies by offloading the cost of upgrading the technology to the PaaS provider

- The PaaS approach, when bundled with underlying IaaS solutions, helps even small start-up companies quickly offer customers integrated solutions on a hosted platform at a very minimal cost.

8. b. With suitable example, explain in detail the types of clouds. (10)

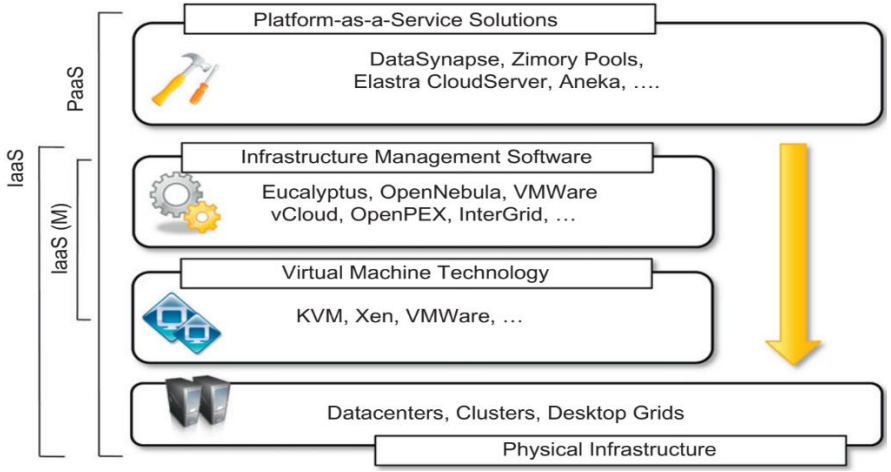
- Clouds constitute the primary outcome of cloud computing.
- They are a type of parallel and distributed system harnessing physical and virtual computers presented as a unified computing resource.
- Clouds build the infrastructure on top of which services are implemented and delivered to customers.
- Such infrastructures can be of different types and provide useful information about the nature and the services offered by the cloud.
- classification according to the administrative domain of a cloud:
  - *"It identifies the boundaries within which cloud computing services are implemented, provides hints on the underlying infrastructure adopted to support such services, and qualifies them".*
- four different types of cloud:
  i. Public clouds: The cloud is open to the wider public.
  ii. Private clouds: implemented within the private premises of an institution and generally made accessible to the members of the institution or a subset of them.
  iii. Hybrid or heterogeneous clouds: combination of the two previous solutions and most likely identifies a private cloud that has been augmented with resources or services hosted in a public cloud.
  iv. Community clouds: characterized by a multi-administrative domain involving different deployment models (public, private, and hybrid), and it is specifically designed to address the needs of a specific industry.

(i) Public clouds

- services offered are made available to anyone, from anywhere, and at any time through the Internet
- they are a distributed system, most likely composed of one or more data-centers connected together, on top of which the specific services offered by the cloud are implemented
- Any customer can easily sign in with the cloud provider, enter the credential and billing details, and use the services offered
- offer solutions for minimizing IT infrastructure costs and serve as a viable option for handling peak loads on the local infrastructure

(ii) Private clouds

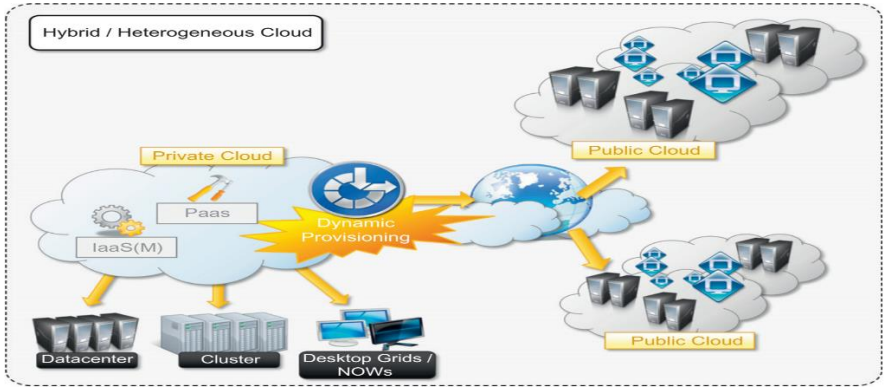
- virtual distributed systems that rely on a private infrastructure and provide internal users with dynamic provisioning of computing resources.
- Instead of a pay-as-you-go model, there could be other schemes in place, taking into account the usage of the cloud and proportionally billing the different departments or sections of an enterprise.



- Key Advantages:
  - Customer Information Protection: security concerns are less critical
  - Infrastructure ensuring SLAs: QoS by means of system monitoring and maintenance, and disaster recovery, and other uptime services
  - Compliance with standard procedures and operations: for third party specific procedures when deploying and executing applications

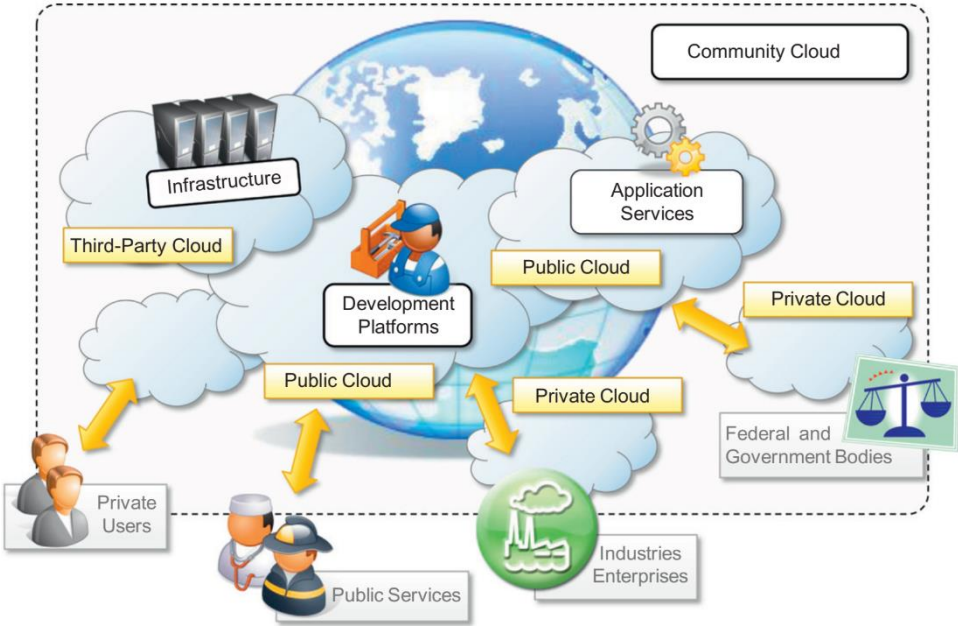
(iii) Hybrid / heterogeneous clouds

- allow enterprises to exploit existing IT infrastructures, maintain sensitive information within the premises, and naturally grow and shrink by provisioning external resources and releasing them when they're no longer needed
- Security concerns are then only limited to the public portion of the cloud
- a heterogeneous distributed system resulting from a private cloud that integrates additional services or resources from one or more public clouds
- dynamic provisioning is a fundamental component
- Hybrid clouds address scalability issues by leveraging external resources for exceeding capacity demand
- These resources or services are temporarily leased for the time required and then released. This practice is also known as cloudbursting



(iv) Community clouds

- distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector.
- The National Institute of Standards and Technologies (NIST) characterizes community clouds as follows:
- *"The infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on-premise or off-premise".*
- Community clouds are also different from private clouds, where the services are generally delivered within the institution that owns the cloud



- The benefits of these community clouds are the following:
- Openness: community clouds are open systems in which fair competition between different solutions can happen
- Community: the infrastructure turns out to be more scalable because the system can grow simply by expanding its user base
- Graceful failures: Since there is no single provider or vendor in control of the infrastructure, there is no single point of failure

9. a. What are the different scientific applications in cloud computing? (10)

- Scientific applications are a sector that is increasingly using cloud computing systems and technologies.
- The immediate benefit seen by researchers and academics is the potentially infinite availability of computing resources and storage at sustainable prices compared to a complete in-house deployment.
- **Healthcare: ECG analysis in the cloud**
- Cloud computing technologies allow the remote monitoring of a patient's heartbeat data, data analysis in minimal time, and the notification of first-aid personnel and doctors should these data reveal potentially dangerous conditions.
- This way a patient at risk can be constantly monitored without going to a hospital for ECG analysis.
- The Web service forms the front-end of a platform that is entirely hosted in the cloud and that leverages the three layers of the cloud computing stack: SaaS, PaaS, and IaaS.
- The Web service constitute the SaaS application that will store ECG data in the Amazon S3 service and issue a processing request to the scalable cloud platform.
- The runtime platform is composed of a dynamically sizable number of instances running the workflow engine and Aneka.
- The number of workflow engine instances is controlled according to the number of requests in the queue of each instance, while Aneka controls the number of EC2 instances
- **Protein structure prediction**
- Protein structure prediction is a computationally intensive task that is fundamental to different types of research in the life sciences.
- Among these is the design of new drugs for the treatment of diseases.
- One project that investigates the use of cloud technologies for protein structure prediction is Jeeva – an integrated Web portal that enables scientists to offload the prediction task to a computing cloud based on Aneka
- **Gene expression data analysis for cancer diagnosis**
- Gene expression profiling is the measurement of the expression levels of thousands of genes at once.
- It is used to understand the biological processes that are triggered by medical treatment at a cellular level.
- The classification of gene expression data samples into distinct classes is a challenging task.
- **Geoscience: satellite image processing**
- Geoscience applications collect, produce, and analyze massive amounts of geospatial and nonspatial data.
- As the technology progresses and our planet becomes more instrumented (i.e., through the deployment of sensors and satellites for monitoring), the volume of data that needs to be processed increases significantly.
- In particular, the geographic information system (GIS) is a major element of geoscience applications.

- Cloud computing is an attractive option for executing these demanding tasks and extracting meaningful information to support decision makers.

9. b. Explain in detail about Dropbox and Icloud. (10)

- Productivity applications replicate in the cloud some of the most common tasks that we are used to performing on our desktop: from document storage to office automation and complete desktop environments hosted in the cloud.

i. DropBox

- With the development of cloud technologies, online storage solutions have turned into SaaS applications and become more usable as well as more advanced and accessible
- Dropbox: an online application that allows users to synchronize any file across any platform and any device in a seamless manner
- Dropbox provides users with a free amount of storage that is accessible through the abstraction of a folder.
- key Advantage:
- availability on different platforms (Windows, Mac, Linux, and mobile) and the capability to work seamlessly and transparently across all of them.

ii. iCloud:

- cloud-based document-sharing application provided by Apple to synchronize iOS-based devices in a completely transparent manner.
- Documents, photos, and videos are automatically synched as changes are made, without any explicit operation.
- there are no plans to provide iCloud with a Web-based interface that would make user content accessible from even unsupported platforms.

10. a. Explain in detail the business and consumer application of cloud computing. (10)

- **CRM and ERP**

- Customer relationship management (CRM) and enterprise resource planning (ERP) applications are market segments that are flourishing in the cloud, with CRM applications the more mature of the two.
- Cloud CRM applications constitute a great opportunity for small enterprises and start-ups to have fully functional CRM software without large up-front costs and by paying subscriptions.
- having access to the business and customer data from everywhere and from any device, has fostered the spread of cloud CRM applications.
- ERP solutions on the cloud are less mature and have to compete with well-established in-house solutions.
- ERP systems integrate several aspects of an enterprise: finance and accounting, human resources, manufacturing, supply chain management, project management, and CRM
- Microsoft Dynamics CRM is the solution implemented by Microsoft for CRM.
- Dynamics CRM is available either for installation on the enterprise's premises or as an online solution priced as a monthly per-user subscription.
- The system is completely hosted in Microsoft's data-centers across the world
- Each CRM instance is deployed on a separate database, and the application provides users with facilities for marketing, sales, and advanced CRM
- Dynamics CRM Online features can be accessed either through a Web browser interface or programmatically by means of SOAP and RESTful Web services.
- Dynamics CRM can also leverage the capability of Windows Azure for the development and integration of new features.

- NetSuite provides a collection of applications that help customers manage every aspect of the business enterprise.
- Its offering is divided into three major products: NetSuite Global ERP, NetSuite Global CRM+, and NetSuite Global Ecommerce.
- Moreover, an all-in-one solution: NetSuite One World, integrates all three products together.
- The NetSuite Business Operating System (NS-BOS) is a complete stack of technologies for building SaaS business applications that leverage the capabilities of NetSuite products.
- The entire infrastructure is hosted in the NetSuite data-centers, which provide warranties regarding application uptime and availability.

10. b. Discuss about media applications of cloud computing. (10)

- Facebook
- Facebook is probably the most evident and interesting environment in social networking.
- the social network is backed by two data centers that have been built and optimized to reduce costs and impact on the environment
- a completely customized stack of opportunely modified and refined open-source technologies constitutes the back-end of the largest social network
- these technologies constitute a powerful platform for developing cloud applications.
- This platform primarily supports Facebook itself and offers APIs to integrate third-party applications with Facebook's core infrastructure to deliver additional services such as social games and quizzes created by others.
- The reference stack serving Facebook is based on LAMP (Linux, Apache, MySQL, and PHP).
- This collection of technologies is accompanied by a collection of other services developed in-house.
- These services are developed in a variety of languages and implement specific functionalities such as search, news feeds, notifications, and others
- Most of the user data are served by querying a distributed cluster of MySQL instances, which mostly contain key-value pairs.
- The development of services is facilitated by a set of internally developed tools.
- Thrift: a collection of abstractions (and language bindings) that allow cross-language development.
- It allows services developed in different languages to communicate and exchange data.
- Bindings for Thrift in different languages take care of data serialization and deserialization, communication, and client and server boilerplate code.
- This simplifies the work of the developers, who can quickly prototype services and leverage existing ones.
- Other relevant services and tools are Scribe, which aggregates streaming log feeds, and applications for alerting and monitoring.