1) **List and explain the common characteristics and advantages of Network centric computing Network Centric Content.**

Network-centric computing and network-centric content share a number of characteristics:

- ➔ Most applications are data-intensive. Computer simulation becomes a powerful tool for scientific research in virtually all areas of science, from physics, biology, and chemistry to archeology. Sophisticated tools for computer-aided design, such as Catia (Computer Aided Three-dimensional Interactive Application), are widely used in the aerospace and automotive industries. The widespread use of sensors contributes to increases in the volume of data. Multimedia applications are increasingly popular; the ever-larger media increase the load placed on storage, networking, and processing systems.
- ➔ Virtually all applications are network-intensive. Indeed, transferring large volumes of data requires high-bandwidth networks; parallel computing, computation steering, and data streaming are examples of applications that can only run efficiently on low-latency networks.
- ➔ The systems are accessed using thin clients running on systems with limited resources. In June 2011 Google released Google Chrome OS, designed to run on primitive devices and based on the browser with the same name.
- ➔ The infrastructure supports some form of workflow management. Indeed, complex computational tasks require coordination of several applications; composition of services is a basic tenet of Web 2.0.

The advantages of network-centric computing and network-centric content paradigms are, at the same time, sources for concern;

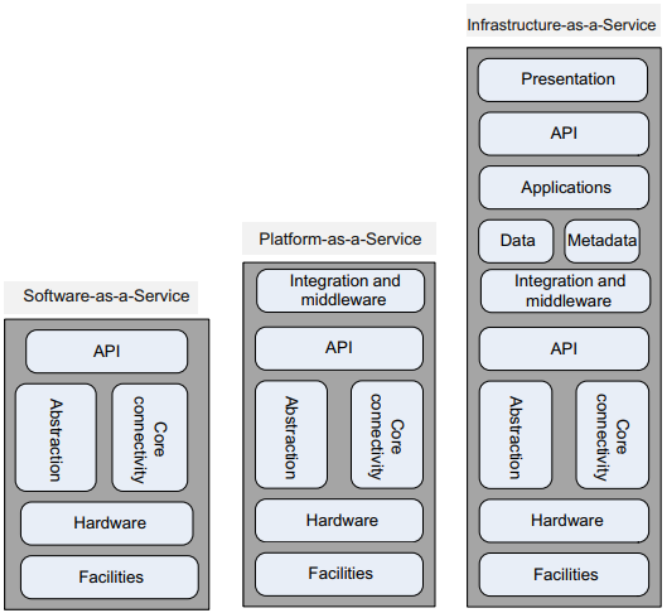
➔ Computing and communication resources (CPU cycles, storage, network bandwidth) are shared and resources can be aggregated to support data-intensive applications. Multiplexing leads to a higher resource utilization; indeed, when multiple applications share a system, their peak demands for resources are not synchronized and the average system utilization increases. On the other hand, the management of large pools of resources poses new challenges as complex systems are subject to phase transitions. New resource management strategies, such as self-organization, and decisions based on approximate knowledge of the state of the system must be considered. Ensuring quality-ofservice (QoS) guarantees is extremely challenging in such environments because total performance isolation is elusive.

➔ Data sharing facilitates collaborative activities. Indeed, many applications in science, engineering, and industrial, financial, and governmental applications require multiple types of analysis of shared data sets and multiple decisions carried out by groups scattered around the globe. Open software development sites are another example of such collaborative activities. Data sharing poses not only security and privacy challenges but also requires mechanisms for access control by authorized users and for detailed logs of the history of data changes.

➔ Cost reduction. Concentration of resources creates the opportunity to pay as you go for computing and thus eliminates the initial investment and reduces significantly the maintenance and operation costs of the local computing infrastructure.

➔ User convenience and elasticity, that is the ability to accommodate workloads with very large peakto-average ratios.

2) **Compare the three CC delivery models based on the limits of responsibility between a cloud user and the cloud service provider. Discuss the security and reliability provided by each model. Justify the difference with a neat diagram.**



Software-as-a-Service (SaaS)

gives the capability to use applications supplied by the service provider in a cloud infrastructure. The applications are accessible from various client devices through a thin-client interface such as a Web browser (e.g., Web-based email). The user does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. SaaS is not suitable for applications that require real-time response or those for which data is not allowed to be hosted externally.

The most likely candidates for SaaS are applications for which:

➔ Many competitors use the same product, such as **email**.

➔ Periodically there is a significant peak in demand, such as **billing** and **payroll**.

➔ There is a need for Web or mobile access, such as mobile **sales management** software.

➔ There is only a short-term need, such as collaborative **software** for a project.

Platform-as-a-Service (PaaS)

gives the capability to deploy consumer-created or acquired applications using programming languages and tools supported by the provider. The user does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, or storage. The user has control over the deployed applications and, possibly, over the application hosting environment configurations. Such services include session management, device integration, sandboxes, instrumentation and testing, contents management, knowledge management, and Universal Description, Discovery, and Integration (UDDI), a platform-independent Extensible Markup Language (XML)-based registry providing a mechanism to register and locate Web service applications.

PaaS is not particulary useful when the application must be portable, when proprietary programming languages are used, or when the underlaying hardware and software must be customized to improve the performance of the application. The major PaaS application areas

are in software development where multiple developers and users collaborate and the deployment and testing services should be automated.

Infrastructure-as-a-Service (IaaS)

offers the capability to provision processing, storage, networks, and other fundamental computing resources; the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of some networking components, such as host firewalls. Services offered by this delivery model include: server hosting, Web servers, storage, computing hardware, operating systems, virtual instances, load balancing, Internet access, and bandwidth provisioning. The IaaS cloud computing delivery model has a number of characteristics, such as the fact that the resources are distributed and support dynamic scaling, it is based on a utility pricing model and variable cost, and the hardware is shared among multiple users. This cloud computing model is particulary useful when the demand is volatile and a new business needs computing resources and does not want to invest in a computing infrastructure or when an organization is expanding rapidly.

3) List the reasons, why cloud computing could be successful, when other paradigms failed, and also mention the challenges faced by cloud computing.

technological advances, a realistic system model, user convenience, and financial advantages. A nonexhaustive list of reasons for the success of cloud computing includes these points:

➔ Cloud computing is in a better position to exploit recent advances in software, networking, storage, and processor technologies. Cloud computing is promoted by large IT companies where these new technological developments take place, and these companies have a vested interest in promoting the new technologies.

➔ A cloud consists of a homogeneous set of hardware and software resources in a single administrative domain. In this setup, security, resource management, fault tolerance, and quality of service are less challenging than in a heterogeneous environment with resources in multiple administrative domains.

➔ Cloud computing is focused on enterprise computing; its adoption by industrial organizations, financial institutions, healthcare organizations, and so on has a potentially huge impact on the economy.

➔ A cloud provides the illusion of infinite computing resources; its elasticity frees application designers from the confinement of a single system.

➔ A cloud eliminates the need for up-front financial commitment, and it is based on a pay-as-you-go approach. This has the potential to attract new applications and new users for existing applications, fomenting a new era of industrywide technological advancements.

Challenges

➔ The most significant challenge is security [19]; gaining the trust of a large user base is critical for the future of cloud computing. It is unrealistic to expect that a public cloud will provide a suitable environment for all applications

➔ The SaaS model faces similar challenges as other online services required to protect private information, such as financial or healthcare services.

➔ The IaaS model is by far the most challenging to defend against attacks. Indeed, an IaaS user has considerably more degrees of freedom than the other two cloud delivery models.

➔ Virtualization is a critical design option for this model, but it exposes the system to new sources of attack. The trusted computing base (TCB) of a virtual environment includes not only the hardware and the hypervisor but also the management operating system.

➔ The next major challenge is related to resource management on a cloud.

➔ The last major challenge we want to address is related to interoperability and standardization. Vendor lock-in, the fact that a user is tied to a particular cloud service provider, is a major concern for cloud users. Standardization would support interoperability and thus alleviate some of the fears that a service critical for a large organization may not be available for an extended period of time. But imposing standards at a time when a technology is still evolving is not only challenging, it can be counterproductive because it may stifle innovation.

4) List the procedure to construct a virtual machine using Eucalyptus. List and explain the components of the system.

Procedure

➔ The euca2ools front end is used to request a VM.

➔ The VM disk image is transferred to a compute node.

➔ This disk image is modified for use by the VMM on the compute node.

➔ The compute node sets up network bridging to provide a virtual network interface controller (NIC)8 with a virtual Media Access Control (MAC) address.

➔ In the head node the DHCP is set up with the MAC/IP pair.

➔ VMM activates the VM.

➔ The user can now ssh directly into the VM.

Components

➔ **Virtual machine**. Runs under several VMMs, including Xen, KVM, and Vmware.

➔ **Node controller**. Runs on every server or node designated to host a VM and controls the activities of the node. Reports to a cluster controller.

➔ **Cluster controller**. Controls a number of servers. Interacts with the node controller on each server to schedule requests on that node. Cluster controllers are managed by the cloud controller.

➔ **Cloud controller**. Provides the cloud access to end users, developers, and administrators. It is accessible through command-line tools compatible withEC2 and through aWeb-based Dashboard. Manages cloud resources, makes high-level scheduling decisions, and interacts with cluster controllers.

➔ **Storage controller.** Provides persistent virtual hard drives to applications. It is the correspondent of EBS. Users can create snapshots from EBS volumes. Snapshots are stored in Walrus and made available across availability zones.

➔ **Storage service (Walrus)**. Provides persistent storage and, similarly to S3, allows users to store objects in buckets.

5) Peer-to-peer systems and clouds share a few goals but not the means to accomplish them. Compare the two classes of systems in terms of architecture, resource management, scope, and security.

Refer 12th

6) **Write the differences that arise among the three storage systems in AWS and explain the monitoring system of AWS.**

Simple Storage System (S3):

- ➔ a storage service designed to store large objects. It supports a minimal set of functions: write, read, and delete.
- ➔ S3 allows an application to handle an unlimited number of objects ranging in size from one byte to five terabytes.
- ➔ An object is stored in a bucket and retrieved via a unique developer-assigned key.
- ➔ A bucket can be stored in a region selected by the user. S3 maintains the name, modification time, an access control list, and up to four kilobytes of user-defined metadata for each object.
- ➔ The object names Services offered by AWS are accessible from the AWS Management Console.
- ➔ S3 supports PUT, GET, and DELETE primitives to manipulate objects but does not support primitives to copy, rename, or move an object from one bucket to another. Appending to an object requires a read followed by a write of the entire object.
- ➔ S3 computes the MD52 of every object written and returns it in a field called Etag.
- ➔ The Amazon S3 SLA guarantees reliability.
- ➔ S3 uses standards-based REST and SOAP interfaces.

Elastic Block Storage(EBS)

- ➔ provides persistent block-level storage volumes for use with Amazon EC2 instances.
- ➔ A volume appears to an application as a raw, unformatted, and reliable physical disk
- ➔ the size of the storage volumes ranges from one gigabyte to one terabyte. The volumes are grouped together in availability zones and are automatically replicated in each zone.
- ➔ An EC2 instance may mount multiple volumes, but a volume cannot be shared among multiple instances.
- ➔ The EBS supports the creation of snapshots of the volumes attached to an instance and then uses them to restart an instance.
- ➔ The storage strategy provided by EBS is suitable for database applications, file systems, and applications using raw data devices.

Simple DB

- ➔ a nonrelational data store that allows developers to store and query data items via Web services requests.
- ➔ It supports store-and-query functions traditionally provided only by relational databases.
- ➔ Simple DB creates multiple geographically distributed copies of each data item and supports high-performanceWeb applications; at the same time,
- ➔ it automatically manages infrastructure provisioning, hardware and software maintenance, replication and indexing of data items, and performance tuning.

	Performance	Cost	Availability and Accessibility	Access Control	Storage and File Size Limits
Amazon S3	- Supports 100 PUT/LIST/DELETE requests per second - Scalable to 300 requests per second	- First 50 TB/month: $0.0245 per GB - Next 450 TB/month: $0.0235 per GB - Over 500 TB/month: - $0.0225 per GB	- 99.99 percent available - Accessible via internet using APIs	- Access is based on IAM - Uses bucket policies and user policies	- No limit on quantity of objects - Individual objects up to 5TB
AWS EBS	- Provisioned IOPS delivers 4000 input/output operations per second	- Use-based cost structure that varies between regions	- 99.99 percent available - Accessible via single EC2 instance	- Security groups - Use-based authentication (IAM)	- Max storage size of 16 TB - No file size limit on disk
AWS EFS	- Up to 7000 file system operations per second	- $0.30, $0.33, or $0.36 per GB-month depending on region	- No publicly available SLA - Accessible from multiple Availability Zones in the same region	- IAM user-based authentication - Security groups	- No limits on size of the system - 52 TB maximum for individual files

CloudWatch is a monitoring infrastructure used by application developers, users, and system administrators to collect and track metrics important for optimizing the performance of applications and for increasing the efficiency of resource utilization. Without installing any software, a user can monitor approximately a dozen preselected metrics and then view graphs and statistics for these metrics.

When launching an Amazon Machine Image (AMI), a user can start the CloudWatch and specify the type of monitoring. Basic Monitoring is free of charge and collects data at five-minute intervals for up to 10 metrics; Detailed Monitoring is subject to a charge and collects data at one-minute intervals. This service can also be used to monitor the latency of access to EBS volumes, the available storage space for RDS DB instances, the number of messages in SQS, and other parameters of interest for applications.

7) List any three Paas and Saas offered by Google and explain the same.

Sass :- Gmail, Google Drive, Google Calendar, Picasa, and Google Groups

- ➔ The **Gmail** service hosts emails on Google servers and, provides a Web interface to access them and tools for migrating from Lotus Notes and Microsoft Exchange.
- ➔ **Google Docs** is Web-based software for building text documents, spreadsheets, and presentations. It supports features such as tables, bullet points, basic fonts, and text size; it allows multiple users to edit and update the same document and view the history of document changes; and it provides a spell checker. The service allows users to import and export files in several formats, including Microsoft Office, PDF, text, and OpenOffice extensions.
- ➔ **Google Calenda**r is a browser-based scheduler; it supports multiple calendars for a user, the ability to share a calendar with other users, the display of daily/weekly/monthly views, and the ability to search events and synchronize with the Outlook Calendar.
- ➔ **Picasa** is a tool to upload, share, and edit images; it provides 1 GB of disk space per user free of charge. Users can add tags to images and attach locations to photos using Google Maps.
- ➔ Google Groups allows users to host discussion forums to create messages online or via email.

Pass :- AppEngine, Google Co-Op, Google Base, Google Drive

- ➔ AppEngine is a developer platform hosted on the cloud. Initially it supported only Python, but support for Java was added later and detailed documentation for Java is available. The database for code development can be accessed with Google Query Language (GQL) with a SQL-like syntax.

- ➔ Google Co-op allows users to create customized search engines based on a set of facets or categories. For example, the facets for a search engine for the database research community available at http://data.cs.washington.edu/coop/dbresearch/index.html are professor, project, publication, jobs.
- ➔ Google Base is a service allowing users to load structured data from different sources to a central repository that is a very large, self-describing, semi-structured, heterogeneous database. It is selfdescribing because each item follows a simple schema: (item type, attribute names). Few users are aware of this service. Google Base is accessed in response to keyword queries posed on Google.com, provided that there is relevant data in the database. To fully integrate Google Base, the results should be ranked across properties. In addition, the service needs to propose appropriate refinements with candidate values in select menus; this is done by computing histograms on attributes and their values during query time.
- ➔ Google Drive is an online service for data storage that has been available since April 2012. It gives users 5 GB of free storage and charges $4 per month for 20 GB. It is available for PCs, MacBooks, iPhones, iPads, and Android devices and allows organizations to purchase up to 16 TB of storage. Specialized structure-aware search engines for several interest areas, including travel, weather, and local services, have already been implemented. However, the data available on the Web covers a wealth of human knowledge; it is not feasible to define all the possible domains and it is nearly impossible to decide where one domain ends and another begins.

8) What is SLA? List the Objectives of SLA, Mention the common areas recorded by SLA and its coverage.

A *service-level agreement (SLA)* is a negotiated contract between two parties, the customer and the service provider. The agreement can be legally binding or informal and specifies the services that the customer receive rather than how the service provider delivers the services.

Objectives

- ➔ Identify and define customers' needs and constraints, including the level of resources, security, timing, and quality of service.
- ➔ Provide a framework for understanding. A critical aspect of this framework is a clear definition of classes of service and costs.
- ➔ Simplify complex issues; for example, clarify the boundaries between the responsibilities of the clients and those of the provider of service in case of failures.
- ➔ Reduce areas of conflict.
- ➔ Encourage dialogue in the event of disputes.
- ➔ Eliminate unrealistic expectations.

An SLA records a common understanding in several areas:

- ➔ services,
- ➔ priorities,
- ➔ responsibilities,
- ➔ guarantees,
- ➔ warranties.

An agreement usually covers:

- ➔ services to be delivered,
- ➔ performance,
- ➔ tracking and reporting,
- ➔ problem management,

- ➔ legal compliance and resolution of disputes,
- ➔ customer duties and responsibilities,
- ➔ security,
- ➔ handling of confidential information, and termination.

Each area of service in cloud computing should define a "target level of service" or a "minimum level of service" and specify the levels of availability, serviceability, performance, operation, or other attributes of the service, such as billing.

9) Define cloud computing. List the delivery and deployment models with examples. Compare them.

Cloud computing is a model for enabling ubiquitous(found everywhere), 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.

Delivery Models

Service Model	Benefits	Risks	Best Fit
IaaS	On-Demand Infrastructure	Security Data persistence Data aggregation Business Risk of Outages Service failures can affect multiple tenants and customers	Non-legacy apps Consolidation Efforts Hosting for Dev & Test
PaaS	Standardized development environment Rapid development & testing	Similar risks as above Vendor Lock-in	New application development Application development that uses provider building blocks to reduce time-to-market
SaaS	Re-usable services Only requires limited configuration & management	Similar risks as IaaS Lack of control Vendor Lock-in	Configuration over customization Commoditized applications

Deployment Models

Private Cloud

It is a cloud-based infrastructure used by stand-alone organizations. It offers greater control over security. The data is backed up by a firewall and internally, and can be hosted internally or externally. Private clouds are perfect for organizations that have high-security requirements, high management demands, and availability requirements.

Public Cloud

This type of cloud services is provided on a network for public use. Customers have no control over the location of the infrastructure. It is based on a shared cost model for all the users, or in the form of a licensing policy such as pay per user. Public deployment models in the cloud are perfect for organizations with growing and fluctuating demands. It is also popular among businesses of all sizes for their web applications, webmail, and storage of non-sensitive data.

Community Cloud

It is a mutually shared model between organizations that belong to a particular community such as banks, government organizations, or commercial enterprises. Community members generally share similar issues of privacy, performance, and security. This type of deployment model of cloud computing is managed and hosted internally or by a third-party vendor.

Hybrid Cloud

This model incorporates the best of both private and public clouds, but each can remain as separate entities. Further, as part of this deployment of cloud computing model, the internal, or external providers can provide resources. A hybrid cloud is ideal for scalability, flexibility, and security. A

perfect example of this scenario would be that of an organization who uses the private cloud to secure their data and interacts with its customers using the public cloud.

10) List the Obstacles of cloud computing that provides opportunity for research fart.

- ➔ ***Availability of service***. What happens when the service provider cannot deliver? Can a large company such as General Motors move its IT to the cloud and have assurances that its activity will not be negatively affected by cloud overload? A partial answer to this question is provided by service-level agreements (SLAs). A temporary fix with negative economical implications is overprovisioning, that is, having enough resources to satisfy the largest projected demand.
- ➔ ***Vendor lock-in***. Once a customer is hooked to one provider, it is hard to move to another. The standardization efforts at National Institute of Standards and Technology (NIST) attempt to address this problem.
- ➔ ***Data confidentiality and auditability***. This is indeed a serious problem;
- ➔ ***Data transfer bottlenecks***. Many applications are data-intensive. A very important strategy is to store the data as close as possible to the site where it is needed. Transferring 1 TB of data on a 1 Mbps network takes 8 million seconds, or about 10 days; it is faster and cheaper to use courier service and send data recoded on some media than to send it over the network. Very high-speed networks will alleviate this problem in the future; for example, a 1 Gbps network would reduce this time to 8,000 s, or slightly more than 2 h.
- ➔ **Performance unpredictability.** This is one of the consequences of resource sharing. Strategies for performance isolation.
- ➔ **Elasticity**, the ability to scale up and down quickly. New algorithms for controlling resource allocation and workload placement are necessary. Autonomic computing based on self-organization and selfmanagement seems to be a promising avenue.

11) Outline the features and functions of the following services offered by AWS: EC2, S3, SQS, Cloud Watch.

EC2

Elastic Compute Cloud (EC2)1 is a Web service with a simple interface for launching instances of an application under several operating systems, such as several Linux distributions, Microsoft Windows Server 2003 and 2008, OpenSolaris, FreeBSD, and NetBSD.

An instance is created either from a predefined Amazon Machine Image (AMI) digitally signed and stored in S3 or from a user-defined image. The image includes the operating system, the run-time environment, the libraries, and the application desired by the user. AMI images create an exact copy of the original image but without configuration-dependent information such as the hostname or the MAC address.

EC2 allows the import of virtual machine images from the user environment to an instance through a facility called VM import. It also automatically distributes the incoming application traffic among multiple instances using the elastic load-balancing facility. EC2 associates an elastic IP address with an account;

- ➔ Launch an instance from an existing AMI and terminate an instance.
- ➔ start and stop an instance.
- ➔ create a new image.
- ➔ add tags to identify an image.
- ➔ reboot an instance.

S3

Simple Storage System (S3) is a storage service designed to store large objects. It supports a minimal set of functions: write, read, and delete. S3 allows an application to handle an unlimited

number of objects ranging in size from one byte to five terabytes. An object is stored in a bucket and retrieved via a unique developer-assigned key. A bucket can be stored in a region selected by the user. S3 maintains the name, modification time, an access control list, and up to four kilobytes of user-defined metadata for each object. The object names Services offered by AWS are accessible from the AWS Management Console.
S3 supports PUT, GET, and DELETE primitives to manipulate objects but does not support primitives to copy, rename, or move an object from one bucket to another. Appending to an object requires a read followed by a write of the entire object.
S3 computes the MD5 of every object written and returns it in a field called ETag. A user is expected to compute the MD5 of an object stored or written and compare this with the ETag; if the two values do not match, then the object was corrupted during transmission or storage. The Amazon S3 SLA guarantees reliability.
S3 uses standards-based REST and SOAP interfaces., the default download protocol is HTTP, but BitTorrent3 protocol interface is also provided to lower costs for high-scale distribution.

SQS
Simple Queue Service (SQS) is a hosted message queue. SQS is a system for supporting automated workflows; it allows multiple Amazon EC2 instances to coordinate their activities by sending and receiving SQS messages.
Any computer connected to the Internet can add or read messages without any installed software or special firewall configurations.
Applications using SQS can run independently and asynchronously and do not need to be developed with the same technologies.
A received message is “locked” during processing; if processing fails, the lock expires and the message is available again. The time-out for locking can be changed dynamically via the ChangeMessageVisibility operation.
Developers can access SQS through standards-based SOAP and Query interfaces. Queues can be shared with other AWS accounts and anonymously; queue sharing can also be restricted by IP address and time-of-day.

Cloud Watch
CloudWatch is a monitoring infrastructure used by application developers, users, and system administrators to collect and track metrics important for optimizing the performance of applications and for increasing the efficiency of resource utilization. Without installing any software, a user can monitor approximately a dozen preselected metrics and then view graphs and statistics for these metrics.
When launching an Amazon Machine Image (AMI), a user can start the CloudWatch and specify the type of monitoring. Basic Monitoring is free of charge and collects data at five-minute intervals for up to 10 metrics; Detailed Monitoring is subject to a charge and collects data at one-minute intervals. This service can also be used to monitor the latency of access to EBS volumes, the available storage space for RDS DB instances, the number of messages in SQS, and other parameters of interest for applications.

12) **Peer-to-peer systems and clouds share a few goals but not the means to accomplish them. Compare the two classes of systems in terms of architecture, resource management, scope, and security.**

Peer-to-peer systems allow individuals to share data and computing resources, primarily storage space without sharing the cost of these resources; sharing other types of resources, such as computing cycles, is considerably more difficult. P2P systems have several desirable properties

➔ Require a minimally dedicated infrastructure, since resources are contributed by the participating systems.

➔ Are highly decentralized.

➔ Are scalable; the individual nodes are not required to be aware of the global state.

➔ Are resilient to faults and attacks, since few of their elements are critical for the delivery of service and the abundance of resources can support a high degree of replication.

➔ Individual nodes do not require excessive network bandwidth the way servers used in case of the client-server model do.

➔ The systems are shielded from censorship due to the dynamic and often unstructured system architecture.

Decentralization raises the question of whether P2P systems can be managed effectively and provide the security required by various applications. The fact that they are shielded from censorship makes them a fertile ground for illegal activities, including distribution of copyrighted content.

A peer-to-peer system is based on an ad-hoc infrastructure where individuals contribute resources to the system, but join and leave the system as they wish; thus, a high degree of redundancy is necessary, the information must be replicated on multiple sites. A peer-to-peer system maintains a directory of data and sites; this directory can be at a central site or distributed

The need to make computing and storage more affordable and to liberate users from the concerns regarding system and software maintenance reinforced the idea of concentrating computing resources in large cloud computing centers

The defining attributes of the cloud computing paradigm are:

- ➔ Cloud computing uses Internet technologies to offer elastic services. The term elastic computing refers to the ability to dynamically acquire computing resources and support a variable workload. A cloud service provider maintains a massive infrastructure to support elastic services.
- ➔ The resources used for these services can be metered and the users can be charged only for the resources they use.
- ➔ Maintenance and security are ensured by service providers.
- ➔ Economy of scale allows service providers to operate more efficiently due to specialization and centralization.
- ➔ Cloud computing is cost-effective due to resource multiplexing; lower costs for the service provider are passed on to the cloud users.
- ➔ The application data is stored closer to the site where it is used in a device- and locationindependent manner; potentially, this data storage strategy increases reliability and security and, at the same time, it lowers communication costs.

A cloud computing center is managed by a cloud service provider which owns the servers and the networking infrastructure. Services are provided based on SLAs (Service Level Agreement). A cloud provides reliable storage and the data is automatically replicated. The level of user control of the cloud resources is different for the three cloud delivery models. Security is a major concern in both cases, especially in the case of cloud computing.

13) **An organization debating whether to install a private cloud or to use a public cloud, e.g., the AWS, for its computational and storage needs, asks your advice. What information will you require to base your recommendation on, and how will you use each one of the following items: (a) the description of the algorithms and the type of the applications the organization will run; (b) the system software used by these applications; (c) the resources needed by each application; (d) the size of the user population; (e) the relative experience of the user population; (d) the costs involved.**

Public clouds have distinct cost advantages over private clouds; there is no initial investment in the infrastructure, no recurring costs for administration, maintenance, energy consumption, and for the user support personnel. The main concern is security and privacy. An organization with very strict security and privacy concerns is very unlikely to use a public cloud.

The type of applications play a critical role, scientific and engineering computations which require a low latency interconnection network and enjoy only fine-grain parallelism are unlikely to fare well on either a public or a private cloud. A large user population is more likely to use identical or similar software and to cooperate by sharing the raw data and the results; thus, a private cloud seems more advantageous in this case. Some of the services offered by private clouds target experiences users, e.g., AWS services such as ElasticBeanstalk, while others are accessible to lay persons.