MODULE 1

Cloud:

Cloud is a distributed collection of servers that host software and infrastructure, and it is accessed over the Internet.

Cloud computing:

Cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet.

Why cloud computing?/Advantages

A non-exhaustive list of reasons for the success of cloud computing includes these points:

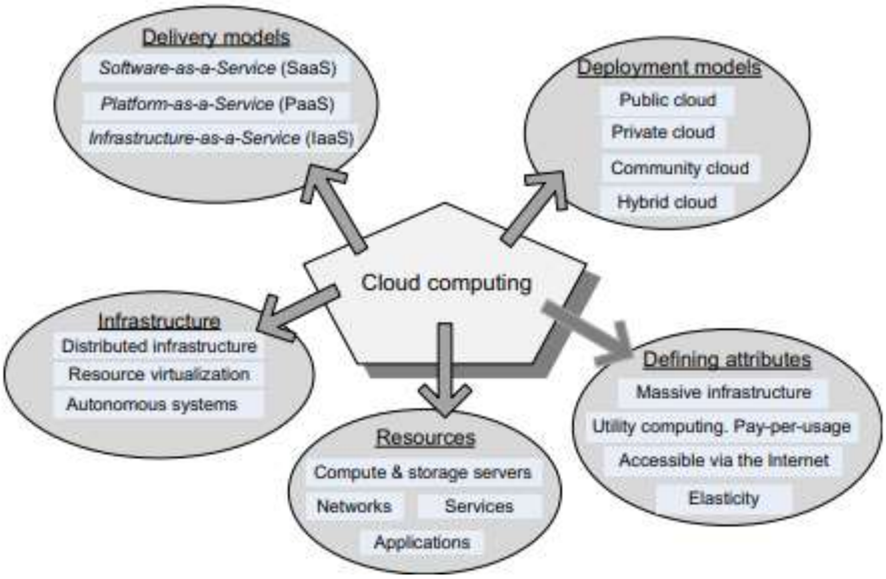
- Cloud computing is in a better position to exploit recent advances in software, networking, storage, and processor technologies.
- A cloud consists of a homogeneous set of hardware and software resources in a single administrative domain. In this setup, security, resource management, fault tolerance, 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.

Disadvantages/Obstacles:

- **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 economic 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**.
- **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.
- **Performance unpredictability** This is one of the consequences of resource sharing.
- **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 self-management seems to be a promising avenue.

Cloud Infrastructure:



Deployment models

Several types of cloud are:

• **Private cloud**. The infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on or off the premises of the organization.

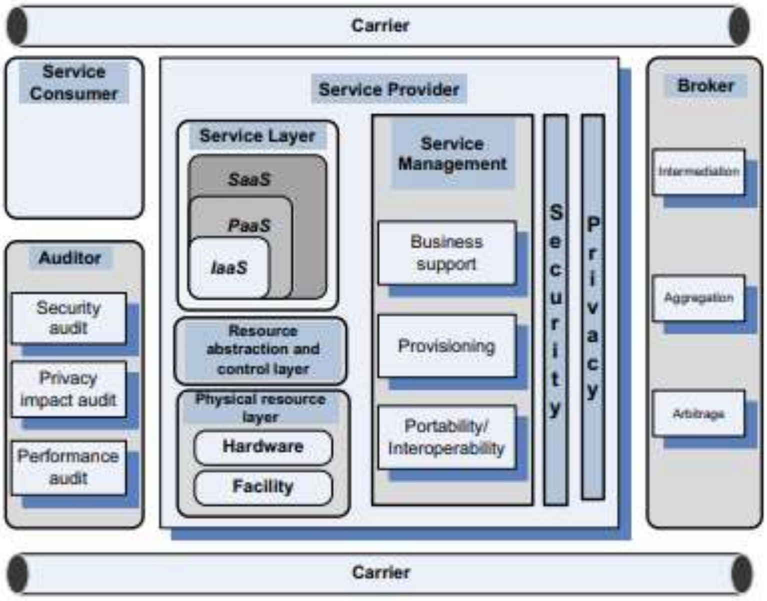
• **Community cloud**: The infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premises or off premises.

• **Public cloud**: The infrastructure is made available to the public or a large industry group and is owned by an organization selling cloud services.

• **Hybrid cloud** : The infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds)

Delivery models and services

NIST Reference Model



✓ Service consumer: The entity that maintains a business relationship with and uses service from service providers.

- ✓ Service provider: The entity responsible for making a service available to service consumers.
- ✓ Carrier: The intermediary that provides connectivity and transport of cloudservices between providers and consumers.
- ✓ Broker: an entity that manages the use, performance, and delivery of cloud services and negotiates relationships between providers and consumers.
- ✓ Auditor: a party that can conduct independent assessment of cloud services, information system operations, performance, and security of the cloud implementation.

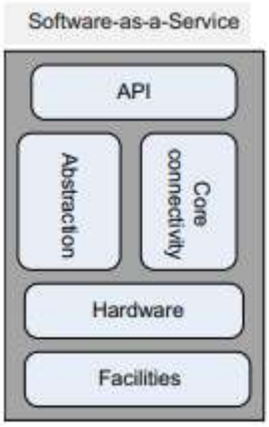
<u>Delivery models:</u>

1. Software-as-a-Service (SaaS)

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, except for limited user-specific application configuration settings.



Services offered include:

- Enterprise services such as workflow management, groupware and collaborative, supply chain, communications, digital signature, customer relationship management (CRM), desktop software, financial management

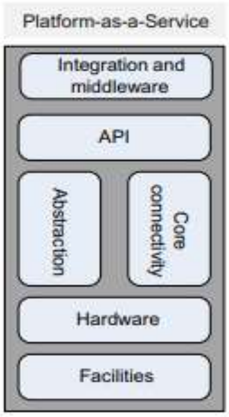
- Web 2.0 applications such as metadata management, social networking, blogs, wikiservices, and portal services
- The SaaS is not suitable for applications that require real-time response or those for which data is not allowed to be hosted externally.

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.

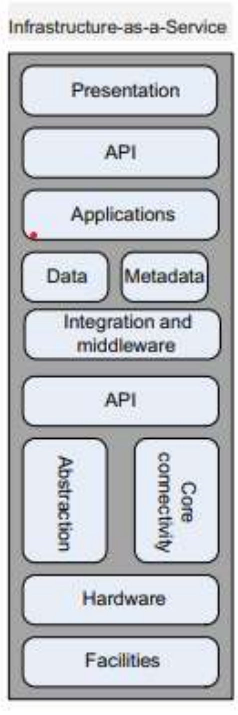
2. Platform-as-a-Service (PaaS)

- ✓ 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.
- ✓ PaaS is not particularly useful when the application must be portable.
- ✓ The major PaaS application areas are in software development where multiple developers and users collaborate and the deployment and testing services should be automated.



3. Infrastructure-as-a-Service (IaaS)

- ✓ Infrastructure-as-a-Service (IaaS) is 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 several 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.



Activities that are necessary to support the three delivery models:

1. **Service management and provisioning**: including virtualization, service provisioning, call center, operations management, systems management, QoS management, billing and accounting, asset management, SLA management, technical support, and backups.

2. **Security management:** including ID and authentication, certification and accreditation, intrusion prevention, intrusion detection, virus protection, cryptography, physical security, incident response, access control, audit and trails, and firewalls.
3. **Customer services**: such as customer assistance and online help, subscriptions, businessintelligence, reporting, customer preferences, and personalization.
4. **Integration services**: including data management and development.

Ethical issues in cloud computing

Cloud computing is based on a paradigm shift with profound implications for computing ethics. The main elements of this shift are:

(i) the control is relinquished to third-party services.

(ii) the data is stored on multiple sites administered by several organizations; and

(iii) multiple services interoperate across the network.

Risks: Unauthorized access, data corruption, infrastructure failure, and service unavailability.

- ✓ Whenever a problem occurs, it is difficult to identify the source and the entity causing it.
- ✓ Systems can span the boundaries of multiple organizations and cross security borders, a process called **de-perimeterization**.
- ✓ Privacy is affected by cultural differences; though some cultures favor privacy, other cultures emphasize community, and this leads to an ambivalent attitude toward privacy on the Internet, which is a global system.

Solution: The term governance means the manner in which something is governed or regulated, the method of management, or the system of regulations.

- ✓ Accountability is a necessary ingredient of cloud computing; adequate information about how data is handled within the cloud and about allocation of responsibility are key elements for enforcing ethics rules in cloud computing.
- ✓ Recorded evidence allows us to assign responsibility; but there can be tension between privacy and accountability, and it is important to establish what is being recorded and who has access to the records.

Cloud vulnerabilities

Vulnerabilities

- ✓ Clouds are affected by malicious attacks and failures of the infrastructure that can affect Internet domain name servers and prevent access to a cloud or can directly affect the clouds.

  For example, an attack at Akamai on June 15, 2004, caused a domain name outage anda major blackout that affected Google, Yahoo!, and many other sites.
- ✓ In May 2009 Google was the target of a serious denial-of-service (DoS) attack that took down services such Google News and Gmail for several days.
- ✓ Lightning caused a prolonged downtime at Amazon on June 29 and 30, 2012; the AWS cloud in the Eastern region of the United States, which consists of 10 data centers across four availability zones, was initially troubled by utility power fluctuations, probably caused by an electrical storm. The recovery from the failure took a very long time and exposed a range of problems.

  For example,

  a) one of the 10 centers failed to switch to backup generators before exhausting the power that could be supplied by uninterruptible power supply (UPS) units.

  b) AWS uses “control planes” to allow users to switch to resources in a different region, and this software component also failed.

  c) The booting process was faulty and extended the time to restart **EC2 (Elastic Computing) and EBS (Elastic Block Store)** services.
- ✓ Another critical problem was a bug in the elastic load balancer (ELB), which is used to route traffic to servers with available capacity.
- ✓ A similar bug affected the recovery process of the Relational Database Service (RDS).This event brought to light “hidden” problems that occur only under special circumstances.

Solution:

- ✓ Clustering the resources in data centers located in different geographical areas is one of the means used today to lower the probability of catastrophic failures.
- ✓ This geographic dispersion of resources can reduce communication traffic and energy costs by dispatching the computations to sites where electric energy is cheaper.
- ✓ It can improve performance by an intelligent and efficient load-balancing strategy.
- ✓ Sometimes a user has the option to decide where to run an application.

Major challenges faced by cloud computing:

Cloud computing inherits some of the challenges of parallel and distributed computing. The specific challenges differ for the three cloud delivery models, but in all cases the difficulties are created by the very nature of utility computing, which is based on **resource sharing and resource virtualization** and requires a different trust model than the ubiquitous user-centric model we have been accustomed to for a very long time. The main challenge is in,

Security:

- 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.
    i. Highly sensitive applications related to the management of the critical infrastructure, healthcare applications, and real time applications will most likely be hosted by private cloud.

    ii. Some applications may be best served by a hybrid cloud setup; such applications could keep sensitive data on a private cloud and use a public cloud for some of the processing.
- In this case a user interacts with cloud services through a well-defined interface; thus, in principle it is less challenging for the service provider to close some of the attack channels. Still, such services are vulnerable to **DoS attack** and the users are fearful of maliciousinsiders.
- **Data in storage** is most vulnerable to attack, so special attention should be devoted to the protection of storage servers.
- **Data replication** is necessary to ensure continuity of service in case storage system failure increases vulnerability.
- **Data encryption** may protect data in storage, but eventually data must be decrypted for processing, and then it is exposed to attack us.
- 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.
- 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.

b) **Resource management**:

- Any systematic rather than ad hoc resource management strategy requires the existence of controllers tasked to implement several classes of policies: admission control, capacity allocation, load balancing, energy optimization, and last but not least, to provide QoS guarantee.

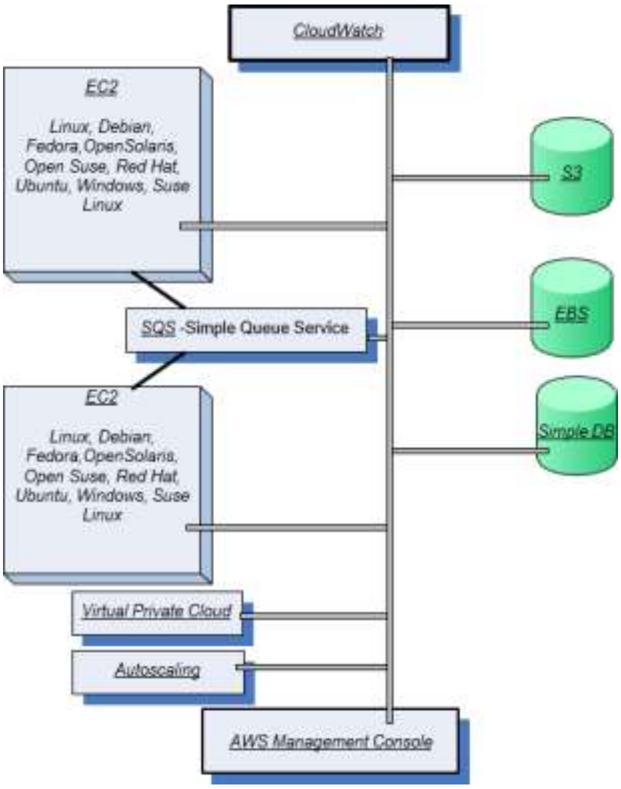
c) **Interoperability and standardization:**

- Vendor lock-in, the fact that a user is tied to a particular cloud service provider, is a majorconcern 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 technology is still evolving is not only challenging, but it can also be counter-productive because it may stifle innovation.

Cloud computing at Amazon

Amazon introduced Amazon Web Services (AWS), based on the IaaS delivery model.

In this model the cloud service provider offers an infrastructure consisting of compute and storage servers interconnected by high-speed networks that support a set of services to access these resources.


CloudWatch
EC2
Linux, Debian, Fedora,OpenSolaris, Open Suse, Red Hat, Ubuntu, Windows, Suse Linux
S3
SQS -Simple Queue Service
EBS
EC2
Linux, Debian, Fedora,OpenSolaris, Open Suse, Red Hat, Ubuntu, Windows, Suse Linux
Simple DB
Virtual Private Cloud
Autoscaling
AWS Management Console

1) Elastic Compute Cloud (EC2)

- It 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, Open-Solaris, 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.

  Functions:

  (i) Launch an instance from an existing AMI and terminate an instance.
  (ii) start and stop an instance.
  (iii) create a new image.
  (iv) Add tags to identify an image.
  (v) Reboot an instance.

- A user can interact with EC2 using a set of **SOAP** messages and can list available AMI images, boot an instance from an image, terminate an image, display the running instances of a user, display console output, and so on.
- 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; this mechanism allows a user to mask the failure of an instance and remap a public IP address to any instance of the account without the need to interact with the software support team.

2) Simple Storage System (S3)

- It 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.
- **Authentication** mechanisms ensure that data is kept secure; objects can be made public, and rights can be granted to other users.
- 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.
- **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.

3) **Elastic Block Store (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.

4) **Simple DB** is 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.

5) **Simple Queue Service (SQS**) is a hosted message queue. SQS is a system for supporting automated workflows; it allows multiple Amazon EC2 instances to coordinatetheir activities by sending and receiving SQS messages.

6) **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.

7) **Auto Scaling** exploits cloud elasticity and provides automatic scaling of EC2 instances. The service supports grouping of instances, monitoring of the instances in a group, and defining triggers and pairs of CloudWatch alarms and policies, which allow the size of the group to be scaled up or down.

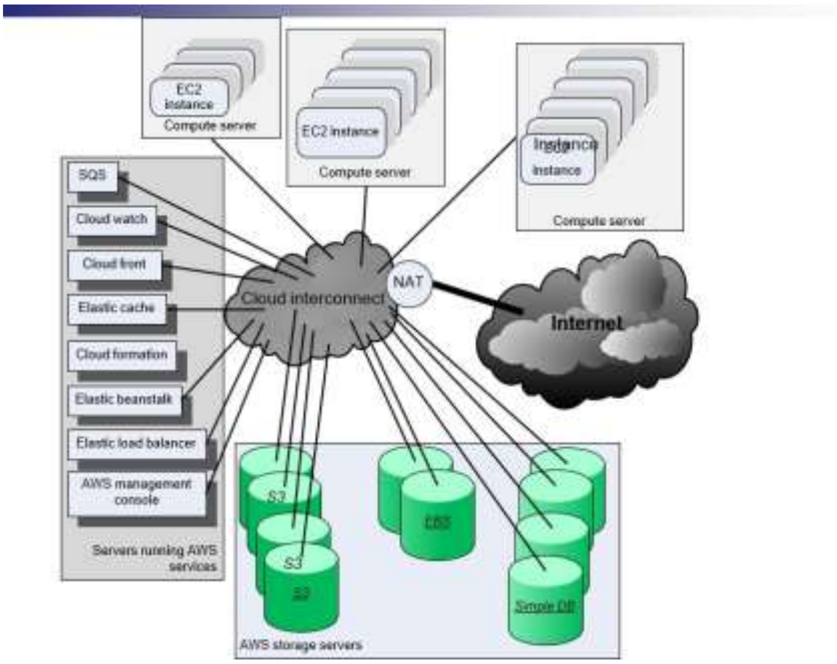
8) **Simple Workflow Service (SWF)**, which supports workflow management and allows scheduling, management of dependencies, and coordination of multiple EC2 instances.
9) **Elastic Cache:** a service enabling Web applications to retrieve data from a managed in-memory caching system rather than a much slower disk-based database; DynamoDB, ascalable and low-latency fully managed NoSQL database service.
10) **CloudFront**: a Web service for content delivery; and Elastic Load Balancer, a cloud service to automatically distribute the incoming requests across multiple instances of the application.
11) **Elastic Beanstalk:** a service that interacts with other AWS services, including EC2, S3, SNS, Elastic Load Balance, and Auto Scaling, automatically handles the deployment, capacity provisioning, load balancing,
    Some of the management functions provided by the service are:
    i. deployment of a new application version (or rollback to a previous version).
    ii. access to the results reported by CloudWatch monitoring service;
    iii. email notifications when application status changes or application servers are addedor removed; and
    iv. access to server login files without needing to login to the application servers.
12) **CloudFormation** allows the creation of a stack describing the infrastructure for an application.

    The user creates a template, a text file formatted as in Javascript Object Notation (JSON), describing the resources, the configuration values, and the interconnection among these resources.

Regions and Availability Zones.

- ✓ Today Amazon offers cloud services through a network of data centers on several continents.
- ✓ In each region there are several availability zones interconnected by high-speed networks; regions communicate through the Internet and do not share resources.
- ✓ An availability zone is a data center consisting of a large number of servers.
- ✓ A server may run multiple virtual machines or instances, started by one or more users; an instance may use storage services, S3, EBS), and Simple DB, as well as other services provided by AWS.
- ✓ Storage is automatically replicated within a region; S3 buckets are replicated within an availability zone and between the availability zones of a region, whereas EBS volumes are replicated only within the same availability zone.

- ✓ Critical applications are advised to replicate important information in multiple regions to be able to function when the servers in one region are unavailable due to catastrophic events.
- ✓ The billing rates in each region are determined by the components of the operating costs, including energy, communication, and maintenance costs.
- ✓ When launched, an instance is provided with a DNS name. This name maps to a private IP address for internal communication within the internal EC2 communication network and a public IP address for communication outside the internal Amazon network,
- ✓ The public IP address is assigned for the lifetime of an instance, and it is returned tothe pool of available public IP addresses when the instance is either stopped or terminated.
- ✓ An instance can request an elastic IP address, rather than a public IP address.



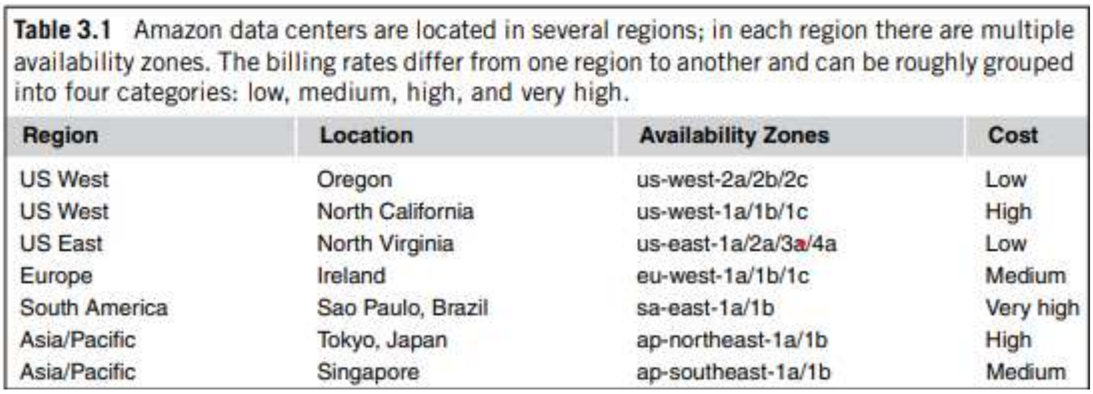


Table 3.1 Amazon data centers are located in several regions; in each region there are multiple availability zones. The billing rates differ from one region to another and can be roughly grouped into four categories: low, medium, high, and very high.

Region	Location	Availability Zones	Cost
US West	Oregon	us-west-2a/2b/2c	Low
US West	North California	us-west-1a/1b/1c	High
US East	North Virginia	us-east-1a/2a/3a/4a	Low
Europe	Ireland	eu-west-1a/1b/1c	Medium
South America	Sao Paulo, Brazil	sa-east-1a/1b	Very high
Asia/Pacific	Tokyo, Japan	ap-northeast-1a/1b	High
Asia/Pacific	Singapore	ap-southeast-1a/1b	Medium

The Charges for Amazon Web Services

Amazon charges a fee for EC2 instances, EBS storage, data transfer, and several other services. The charges differ from one region to another and depend on the pricing model. There are three pricing models for EC2 instances: **on-demand, reserved, and spot.**

- **On-demand instances** use a flat hourly rate, and the user is charged for the time aninstance is running; no reservation is required for this most popular model.
- For **reserved instances** a user pays a one-time fee to lock in a typically lower hourly rate. This model is advantageous when a user anticipates that the application will require asubstantial number of CPU cycles and this amount is known in advance. Additional capacity is available at the larger standard rate.
- In case **of spot instances,** users bid on unused capacity and their instances are launchedwhen the market price reaches a threshold specified by the user.

The EC2 system offers several instance types:

- **Standard instances.** Micro (StdM), small (StdS), large (StdL), extra-large (StdXL); small is the default.
- **High memory instances**. High-memory extra-large (HmXL), high-memory double extra-large (Hm2XL), and high-memory quadruple extra-large (Hm4XL). • High CPU instances. High-CPU extra-large (HcpuXL).
- **Cluster computing**. Cluster computing quadruple extra-large.

Cloud computing: the Google perspective

Google's effort is concentrated in the area of **Software-as-a-Service** (SaaS). Services such as **Gmail, Google Drive, Google Calendar, Picasa, and Google** Groups are free of charge for individual users and available for a fee for organizations. The data for these services is stored in data centers on the cloud.

- ✓ 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.
- ✓ **Google Calendar** 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.

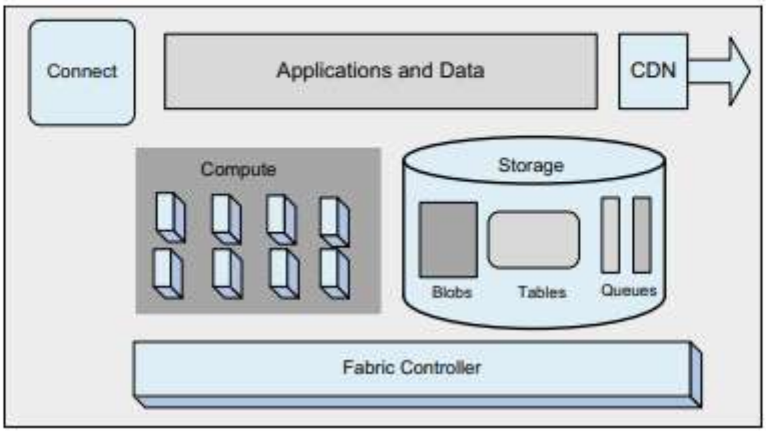
Google is also a leader in the **Platform-as-a-Service (PaaS)** space.

- ✓ **AppEngine** is a developer platform hosted on the cloud. documentation for Java.
- ✓ The deep Web is content stored in databases and served as pages created dynamically byquerying HTML forms. Such content is unavailable to crawlers that are unable to fill out such forms.

  Examples of deep Web sources are sites with geographic-specific information, such as local stores, services, and businesses; sites that report statistics and analysis produced by governmental and nongovernmental organizations; art collections; photo galleries; bus, train, and airline schedules; and so on. Structured content is created by labelling.
- ✓ **Flickr** and Google Co-op are examples of structures where labels and annotations are added to objects, images, and pages stored on the Web.
- ✓ **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 self-describing because each item follows a simple schema: (item type, attribute names). Few users are aware of this service.
- ✓ **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.

Microsoft Windows Azure and online services

- ✓ Azure and Online Services are, respectively, PaaS and SaaS cloud platforms from Microsoft.
- ✓ Windows Azure is an operating system, SQL Azure is a cloud-based version of the SQL Server, Azure AppFabric (formerly .NET Services) is a collection of services for cloud applications.



Windows Azure has three core components :

- **Compute**: which provides a computation environment to run the application.
- **Storage:**
  - **Blobs, tables, queues, and drives** are used as scalable storage. A blob contains binary data; a container consists of one or more blobs. Blobs can be up to a terabyte, and they may have associated metadata (e.g., the information about where a JPEG photograph was taken). Blobs allow a Windows Azure role instance to interact with persistent storage as though it were a local NTFS6 file system.
  - Queues enable Web role instances to communicate asynchronously with Worker role instances.
- **Fabric Controller**,
  - Deploys, manages, and monitors applications; it interconnects nodes consisting of servers,high-speed connections, and switches.
  - Scaling, load balancing, memory management, and reliability are ensured by a fabric controller, a distributed application replicated across a group of machines that owns all of the resources in its environment – computers, switches, load balancers – and it is aware of every Windows Azure application.
  - The **fabric controller** decides where new applications should run; it chooses the physical servers to optimize utilization using configuration information uploaded with each Windows Azure application.
  - Configuration is XML based. The fabric controller uses this configuration file to determine how many VMs to create.

- **The Content Delivery Network** (CDN) maintains cache copies of data to speed up computations.
- The **Connect** subsystem supports IP connections between the users and their applications running on Windows Azure.
- The **API interface** to Windows Azure is built on REST, HTTP, and XML.
- The platform includes **five services**: Live Services, SQL Azure, AppFabric, SharePoint, and Dynamics CRM.
- A **client library and tools** are also provided for developing cloud applications in Visual Studio.

The computations carried out by an application are implemented as one or more roles. We can distinguish as,

- Web role instances used to create Web applications.
- Worker role instances used to run Windows-based code.
- VM role instances that run a user provided image.

Open-source software platforms for private clouds

Opensource cloud computing platforms such as Eucalyptus , Open Nebula, and Nimbus can be used as a control infrastructure for a private cloud.

Schematically, a cloud infrastructure carries out the following steps to run an application:

- ✓ Retrieves the user input from the front end.
- ✓ Retrieves the disk image of a VM from a repository.
- ✓ Locates a system and requests the VMM running on that system to set up a VM.
- ✓ Invokes the DHCP and the IP bridging software to set up a MAC and IP address for the VM.

1) **Eucalyptus: E**lastic **U**tility **C**omputing **A**rchitecture for **L**inking **Y**our **P**rograms **T**o **U**seful **S**ystems. This open-source software is compatible with EC2 of AWS. The components are as seen below,

   - ✓ **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 cloud access to end users, developers, and administrators. It is accessible through command-line tools compatible with EC2 and through a Web-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.

The procedure to construct a virtual machine in Eucalyptus based on the generic one described in:

- ✓ 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 interfacecontroller (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.

2) **Open-Nebula** (www.opennebula.org) is a private cloud with users logging into the head node to access cloud functions. The system is centralized and its default configuration uses **NFS (Network File System).**

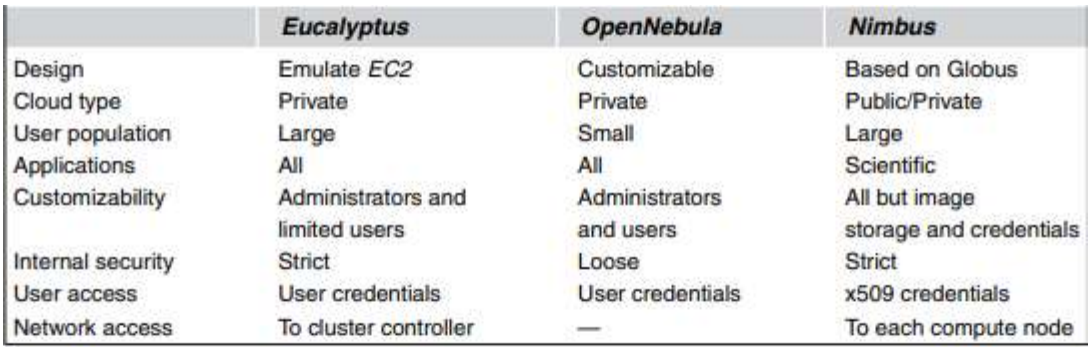
The procedure to construct a virtual machine centralized of several steps:

- The user signs into the head node using ssh.
- The system uses the **onevm** command to request a VM.
- The VM template disk image is transformed to fit the correct size and configuration within the NFS directory on the head node.
- The **oned** daemon on the head node uses ssh to log into a compute node.
- The compute node sets up network bridging to provide a virtual NIC with a virtual MAC.

- The files needed by the VMM are transferred to the compute node via the NFS.
- The VMM on the compute node starts the VM.
- The user is able to ssh directly to the VM on the compute node.

3) **Nimbus** (www.nimbusproject.org) is a cloud solution for scientific applications based on the Globus software. The system inherits from Globus the image storage, the credentials for user authentication, and the requirement that a running Nimbus process can ssh into all compute nodes. Customization in this system can only be done by the system administrators.

4) **OpenStack** is an open-source project started in 2009 at the National Aeronautics and Space Administration (NASA) in collaboration with Rackspace (www.rackspace.com) to develop a scalable cloud operating system for farms of servers using standard hardware. Though recently NASA has moved its cloud infrastructure to AWS in addition to Rackspace, several other companies, including HP, Cisco, IBM, and Red Hat, have an interest in OpenStack . The administrators and the users control their resources using an extensible Web application called the Dashboard.



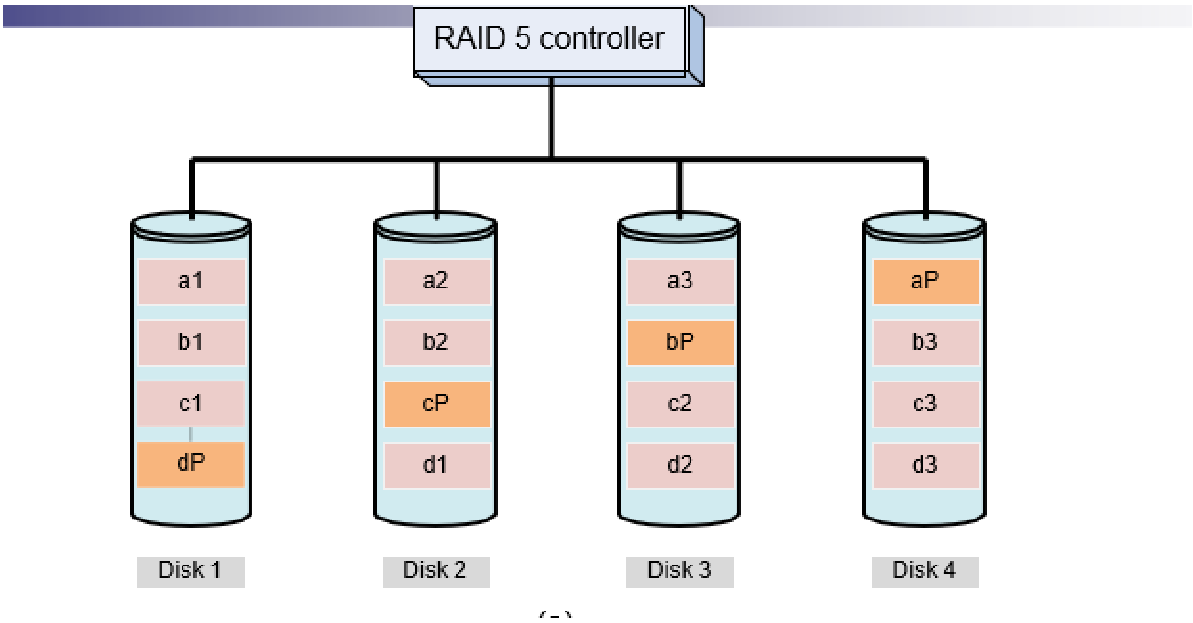
	Eucalyptus	*OpenNebula*	*Nimbus*
Design	Emulate *EC2*	Customizable	Based on Globus
Cloud type	Private	Private	Public/Private
User population	Large	Small	Large
Applications	All	All	Scientific
Customizability	Administrators and limited users	Administrators and users	All but image storage and credentials
Internal security	Strict	Loose	Strict
User access	User credentials	User credentials	x509 credentials
Network access	To cluster controller	—	To each compute node

Cloud storage diversity and vendor lock-in

- ✓ There are several risks involved when a large organization relies solely on a single cloud provider.
- ✓ As the short history of cloud computing shows, cloud services may be **unavailable** for a short or even an extended period of time. Such an interruption of service is likely to negatively impact the organization and possibly diminish or cancel completely the benefits of utility computing for that organization.
- ✓ The potential for permanent data loss in case of a catastrophic system failure poses an equally great danger.
- ✓ A solution to guarding against the problems posed by the **vendor lock-in** is to replicatethe data to multiple cloud service providers.

- ✓ The overhead to maintain data consistency could drastically affect the performance of the virtual storage system consisting of multiple full replicas of the organization's data spread over multiple vendors.
- ✓ Another solution could be based on an extension of the design principle of a RAID-5 system used for reliable data storage.

RAID-5 system uses block-level stripping with distributed parity over a disk array.



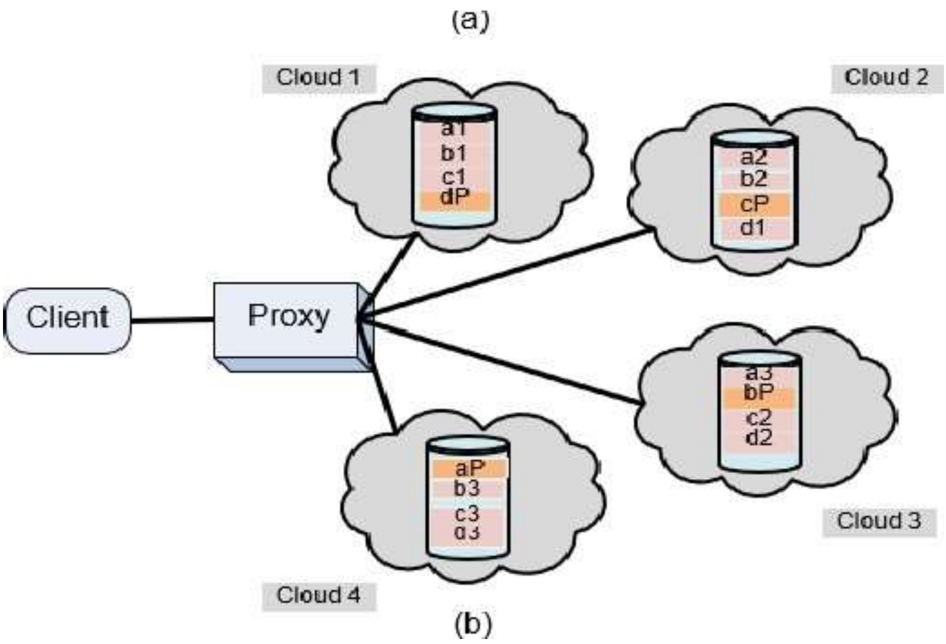
- ✓ The disk controller distributes the sequential blocks of data to the physical disks and computes a parity block by bitwise XOR-ing of the data blocks.
- ✓ **The parity block** is written on a different disk for each file to avoid the bottleneck possible when all parity blocks are written to a dedicated disk.
- ✓ This technique allows us to recover the data after a single disk loss. For example, if Disk 2 in is lost, we still have all the blocks of the third file, c1, c2, and c3, and we can recover the missing blocks for the others as follows:

  a2 = (a1) XOR (a P) XOR (a3)
  b2 = (b1) XOR (b P) XOR (b3)
  d1 = (d P) XOR (d2) XOR (d3)

- ✓ The **Redundant Array of Cloud Storage (RACS)** system uses the same data model andmimics the interface of the S3 provided by AWS.
- ✓ The **S3 system** stores the data in buckets, each bucket being a flat namespace with keys associated with objects of arbitrary size but less than 5 GB.

Then the system is able to recover from the failure of a single proxy; clients are connected to several proxies and can access the data stored on multiple clouds.

Cloud computing interoperability: the Intercloud



Problems

- ✓ Cloud interoperability could alleviate the concern that users could become hopelessly dependent on a single cloud service provider, the so-called vendor lock-in. Closer scrutiny shows that the extension of the concept of interoperability from networks to clouds is far from trivial.
- ✓ Network offers one high-level service, the transport of digital information from a source, a host outside a network, to a destination, another host, or another network that can deliver the information to its destination.
- ✓ The three elements on which agreements were reached are, respectively, the IP address, the IP protocol, and transport protocols such as TCP and UDP
- ✓ The situation is quite different in cloud computing. First, there are **no standards** for storage processing; second, the clouds we have seen so far are based on different delivery models: SaaS, PaaS, and IaaS.

Solution:

- ✓ First, we need a set of **standards for interoperability** covering items such as naming, addressing, identity, trust, presence, messaging, multicast, and time.
- ✓ Indeed, we need **common standards for identifying all the objects** involved as

- ✓ well as the means to transfer, store, and process information, and we also need a common clock to measure the time between two events.
- ✓ An Intercloud would then require the development of an **ontology** for cloud computing. Then each cloud service provider would have to create a description of all resources and services using this ontology. Due to the very large number of systems and services, the volume of information provided by individual cloud service providers would be so large that a distributed database not unlike the Domain Name Service (DNS) would have to be created and maintained.
- ✓ Each cloud would then require an interface, a so-**called Intercloud exchange**, to translate the common language describing all objects and actions included in a request originating from another cloud in terms of its internal objects and actions. To be more precise, a request originated in one cloud would have to be translated from the internal representation in that cloud to **a common representation** based on the shared ontology and then, at the destination, it would be translated into an internal representation that can be acted on by the destination cloud. This immediately raises the question of efficiency and performance.
- ✓ **Security** is a major concern for cloud users, and an Intercloud could only create new threats. The primary concern is that tasks will cross from one administrative domain to another and that sensitive information about the tasks and users could be disclosed duringthis migration.

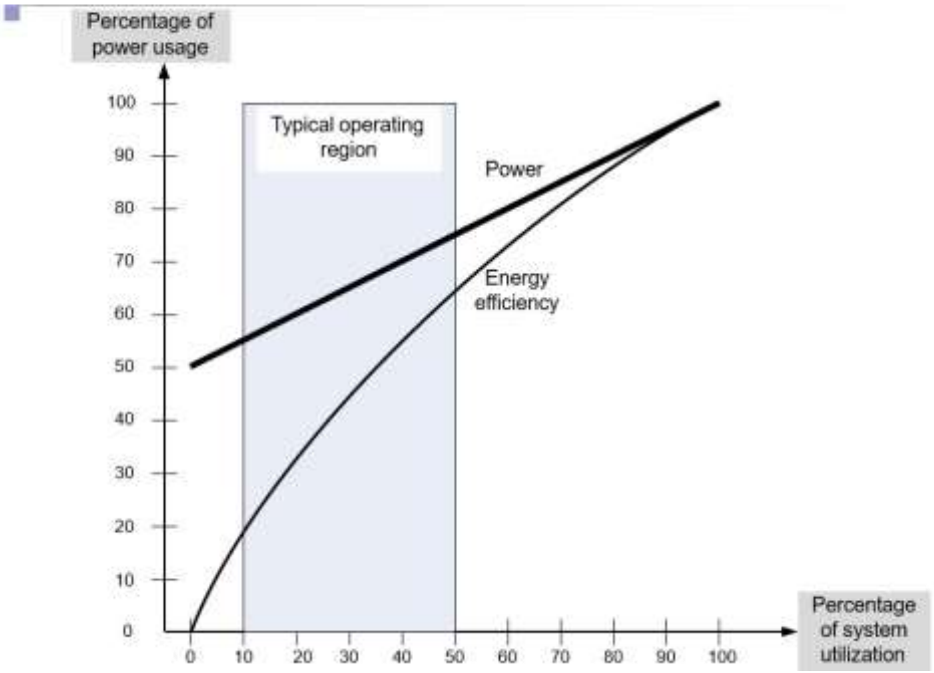
Energy use and ecological impact of large-scale data centers

- ✓ The operating efficiency of a system is captured by an expression of "performance per Watt of power."
- ✓ It is widely reported that, during the last two decades, the performance of computing systems has increased much faster than their operating efficiency.

  For example, during the period 1998–2007, the performance of supercomputers increasedby 7,000% whereas their operating efficiency increased by only 2,000%.
- ✓ In an ideal world, the energy consumed by an idle system should be near zero and should grow linearly with the system load.
- ✓ In real life, even machines whose power requirements scale linearly, use more than half the power when idle than they use at full load.
- ✓ Energy-proportional systems could lead to large savings in energy costs for computing clouds. An energy-proportional system consumes no power when idle, very little power under a light load, and gradually more power as the load increases. An ideal energy-proportional system is always operating at 100% efficiency.
- ✓ Energy saving in large-scale storage systems is also of concern.

- ✓ A strategy to reduce energy consumption is to concentrate the workload on a small number of disks and allow the others to operate in a low-power mode.
    - ▪ One of the techniques to accomplish this task is based on replication.
    - ▪ Another technique is based on data migration.
- ✓ The system uses data storage in virtual nodes managed with a distributed hash table.
- ✓ The migration is controlled by two algorithms, a short-term optimization algorithm, used for gathering or spreading virtual nodes according to the daily variation of the workload so that the number of active physical nodes is reduced to a minimum, and a long-term optimization algorithm, used for coping with changes in the popularity of data over a longer period.
- ✓ Many proposals argue that dynamic resource provisioning is necessary to minimize power consumption. Two main issues are critical for energy saving: the amount of resources allocated to each application and the placement of individual workloads.

  For example, a resource management framework combining a utility-based dynamic virtual machine provisioning manager with a dynamic VM placement manager to minimize power consumption and reduce SLA violations.
- ✓ The support for network-centric content consumes a very large fraction of the network bandwidth; according to the CISCO VNI forecast, consumer traffic was responsible for around 80% of bandwidth use in 2009 and is expected to grow at a faster rate than business traffic.
- ✓ Data intensity for various activities ranges from 20 MB/minute for HDTV streaming to 10 MB/minute for standard TV streaming, 1.3 MB/minute for music streaming,0.96 MB/minute for Internet radio, 0.35 MB/minute for Internet browsing, and 0.0025 MB/minute for e-book reading.
- ✓ The power consumption required by different types of human activities is partially responsible for the world's greenhouse gas emissions.
- ✓ According to a recent study, the greenhouse gas emissions due to data centers are estimated to increase from 116 × 106 tons of C O2 in 2007 to 257 tons in 2020, due primarily to increased consumer demand.
- ✓ Environmentally opportunistic computing is a macroscale computing idea that exploits the physical and temporal mobility of modern computer processes. A prototype called a Green Cloud.



Even when power requirements scale linearly with the load, the energy efficiency of a computing system is not a linear function of the load; even when idle, a system may use 50% of the power corresponding to the full load. Data collected over a long period of time shows that the typical operating region for the servers at a data center is from about 10% to 50% of the load.

Service- and compliance-level agreements:

✓ **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 receives rather than how the service provider delivers the services.

The objectives of the agreement are:

- 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 responsibilitiesof 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: (i) services, (ii) priorities, (iii)responsibilities, (iv) guarantees, and (v) warranties.

✓ 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. Penalties may also be specified in the case of noncompliance with the SLA. It is expected that any service-oriented architecture (SOA) will eventually include middleware supporting SLA management.

There are two well-differentiated phases in SLA management: the negotiation of the contract and the monitoring of its fulfilment in real time. In turn, automated negotiation has three main components:

(i) The object of negotiation, which define the attributes and constraints under negotiation.
(ii) The negotiation protocols, which describe the interaction between negotiating parties
(iii) The decision models are responsible for processing proposals and generating counter proposals.