UNIT II

Cloud Infrastructure: At Amazon, The Google Perspective, Microsoft Windows Azure, Open Source Software Platforms, Cloud storage diversity, Inter cloud, energy use and ecological impact, responsibility sharing, user experience, Software licensing, Cloud Computing : Applications and Paradigms: Challenges for cloud, existing cloud applications and new opportunities, architectural styles, workflows, The Zookeeper, HPC on cloud.

Cloud Infrastructure:

- Amazon is a pioneer in IaaS, Google's efforts are focused on SaaS and PaaS delivery models, and Microsoft is involved in PaaS.
- Private clouds are an alternative to public clouds. Open-source cloud computing platforms such as Eucalyptus, OpenNebula, Nimbus, and OpenStack can be used as a control infrastructure for a private cloud.

Cloud computing at Amazon:

In mid-2000 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.

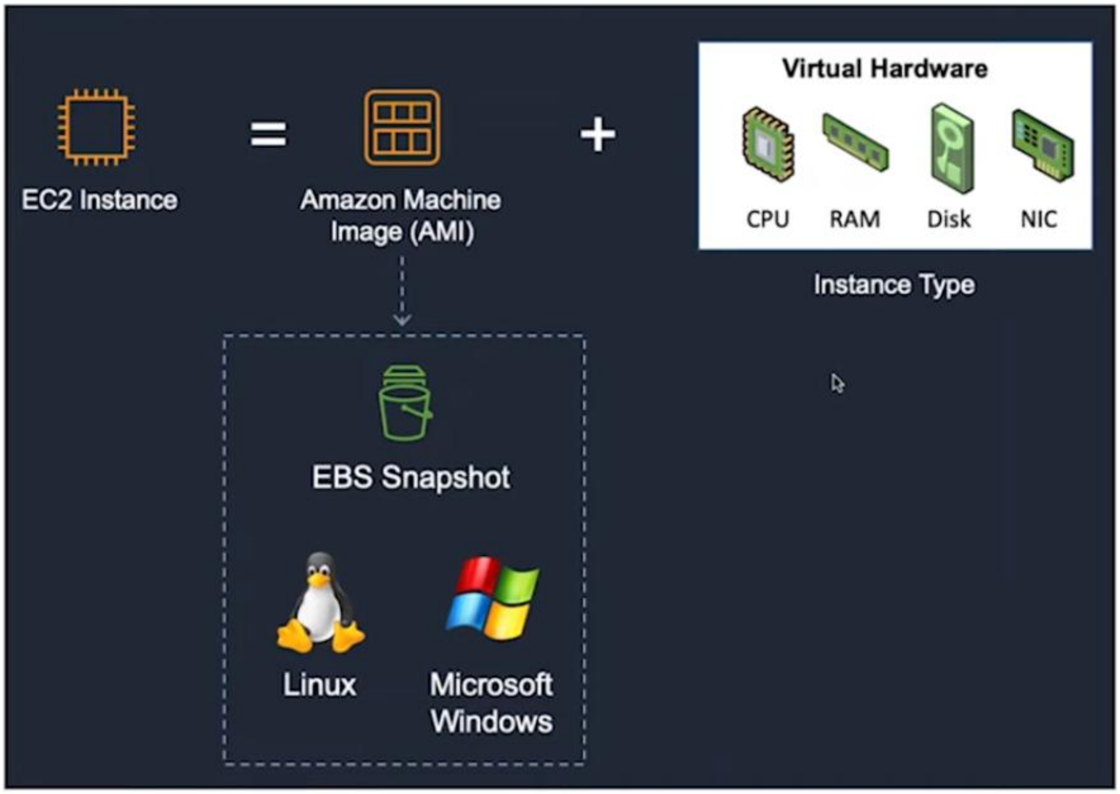
- It is reported that in 2012, businesses in 200 countries used the AWS, demonstrating the international appeal of this computing paradigm.
- A significant number of large corporations as well as start-ups take advantage of computing services supported by the AWS infrastructure.
- For example, one start-up reports that its monthly computing bills at Amazon are in the range of $100,000, whereas it would spend more than $2,000,000 to compute using its own infrastructure, without benefit of the speed and flexibility offered by AWS.

Amazon Web Services:

- Amazon Web Services (AWS) is Amazon's cloud web hosting platform that offers flexible, reliable, scalable, easy-to-use, and cost-effective solutions.
- Amazon was the first provider of cloud computing; it announced a limited public beta release of its Elastic Computing platform called EC2 in August 2006.
- Elastic Compute Cloud (EC2) 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.

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.



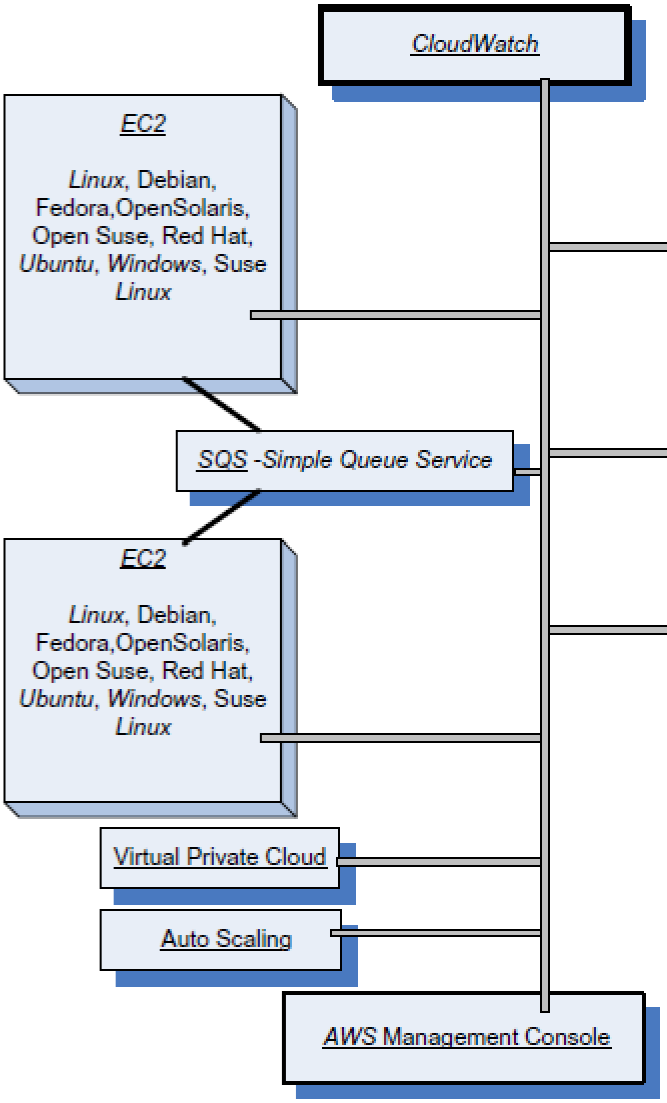
Simple Storage System (S3) is a storage service designed to store large objects. It supports a minimalset of functions: write, read, and delete.

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. 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.

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.

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.


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

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.

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.

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.

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.

Virtual Private Cloud (VPC) provides a bridge between the existing IT infrastructure of an organization and the AWS cloud. The existing infrastructure is connected via a virtual private network (VPN) to a set of isolated AWS compute resources.

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.

Users have several choices for interacting with and managing AWS resources from either a Web browser or from a system running Linux or Microsoft Windows:

1. The *AWS*WebManagement Console, available at http://aws.amazon.com/console/; this is the easiest way to access all services, but not all options may be available in this mode.
2. Command-line tools; see http://aws.amazon.com/developertools.
3. AWS SDK libraries and toolkits provided for several programming languages, including Java, PHP,4 C#, and Obj C.
4. Raw REST requests

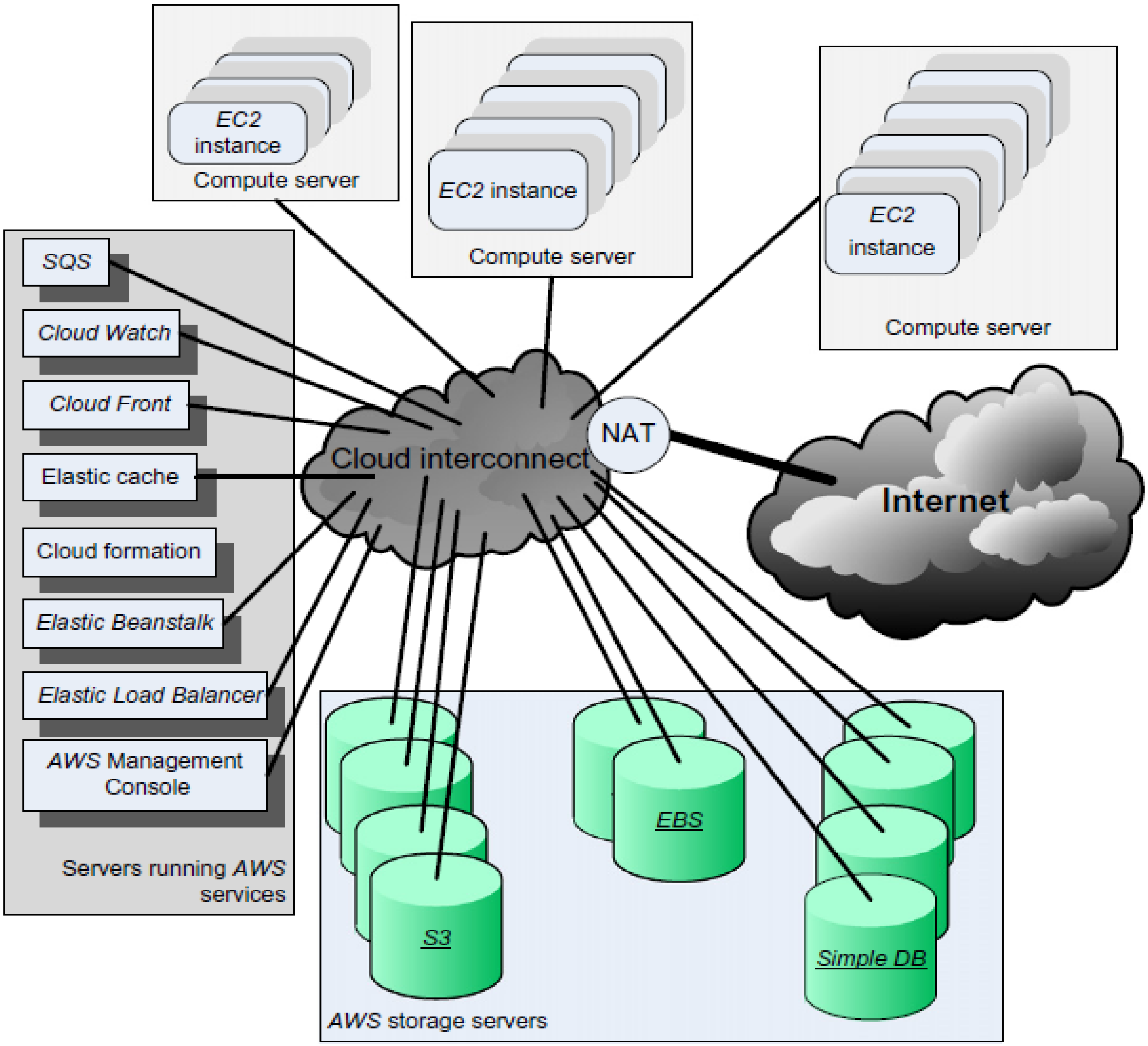
Regions and Availability Zones.

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.

A user can request virtual servers and storage located in one of the regions. The user can also request virtual servers in one of the availability zones of that region. The Elastic Compute Cloud (EC2) service allows a user to interact and to manage the virtual servers.

A user can request virtual servers and storage located in one of the regions. The user can also request virtual servers in one of the availability zones of that region. The Elastic Compute Cloud (EC2) service allows a user to interact and to manage the virtual servers.



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; http://aws.amazon.com/ec2/pricing for the current pricing structure.

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 an instance is running; no reservation is required for this most popular model.

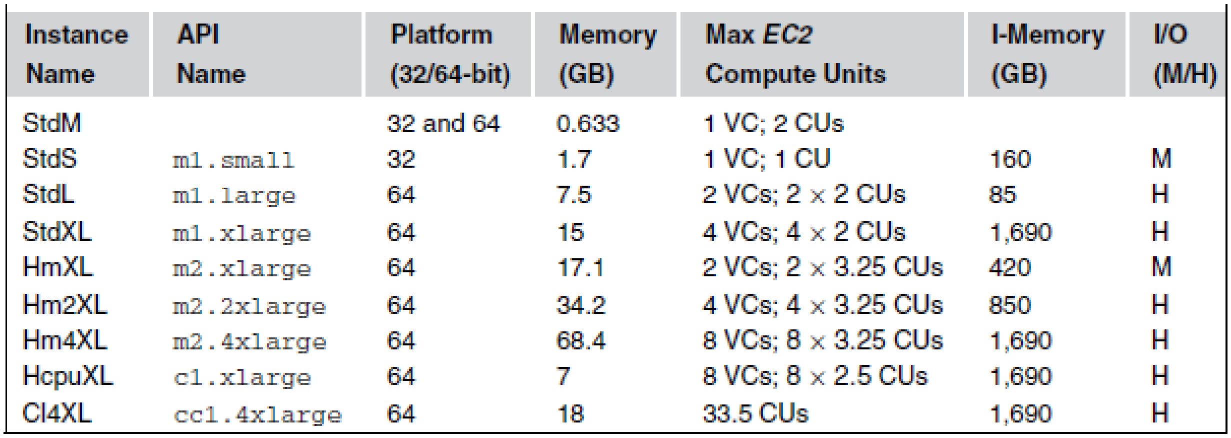
The *EC2* system offers several instance types:

- 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

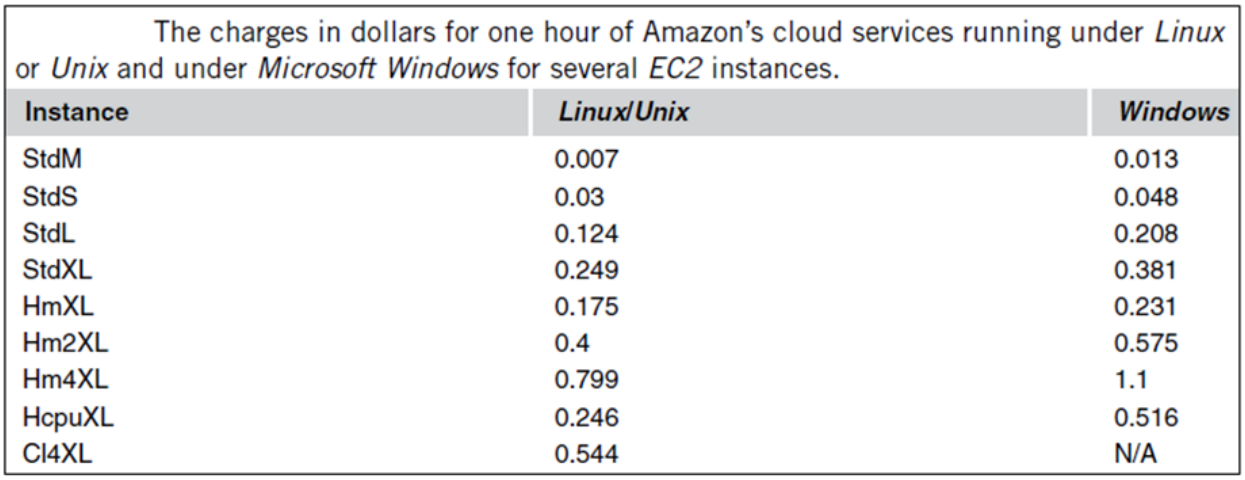
(Hm2XL), and high-memory quadruple extra-large (Hm4XL).

- High CPU instances. High-CPU extra-large (HcpuXL).
- Cluster computing. Cluster computing quadruple extra-large (Cl4XL).

The nine instances supported by *EC2*:



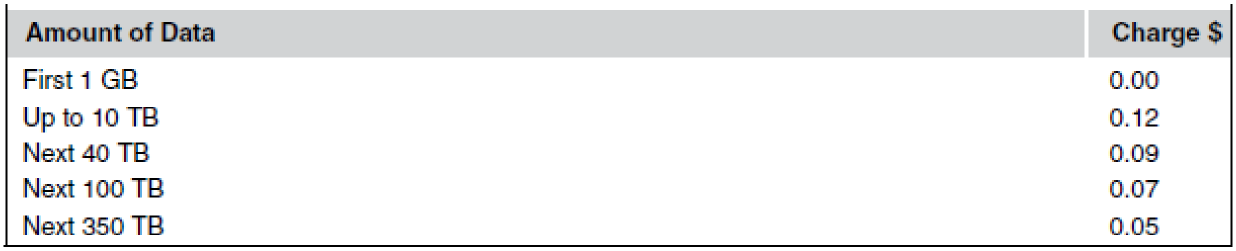
Instance Name	API Name	Platform (32/64-bit)	Memory (GB)	Max *EC2* Compute Units	I-Memory (GB)	I/O (M/H)
StdM		32 and 64	0.633	1 VC; 2 CUs		
StdS	`m1.small`	32	1.7	1 VC; 1 CU	160	M
StdL	`m1.large`	64	7.5	2 VCs; 2 × 2 CUs	85	H
StdXL	`m1.xlarge`	64	15	4 VCs; 4 × 2 CUs	1,690	H
HmXL	`m2.xlarge`	64	17.1	2 VCs; 2 × 3.25 CUs	420	M
Hm2XL	`m2.2xlarge`	64	34.2	4 VCs; 4 × 3.25 CUs	850	H
Hm4XL	`m2.4xlarge`	64	68.4	8 VCs; 8 × 3.25 CUs	1,690	H
HcpuXL	`c1.xlarge`	64	7	8 VCs; 8 × 2.5 CUs	1,690	H
Cl4XL	`cc1.4xlarge`	64	18	33.5 CUs	1,690	H



The charges in dollars for one hour of Amazon's cloud services running under *Linux* or *Unix* and under *Microsoft Windows* for several *EC2* instances.

Instance	*Linux*/*Unix*	*Windows*
StdM	0.007	0.013
StdS	0.03	0.048
StdL	0.124	0.208
StdXL	0.249	0.381
HmXL	0.175	0.231
Hm2XL	0.4	0.575
Hm4XL	0.799	1.1
HcpuXL	0.246	0.516
Cl4XL	0.544	N/A

Monthly charges in dollars for data transfer out of the US West (Oregon) region.



Amount of Data	Charge $
First 1 GB	0.00
Up to 10 TB	0.12
Next 40 TB	0.09
Next 100 TB	0.07
Next 350 TB	0.05

Cloud computing: the Google perspective

- Google’s effort is concentrated in the area of Software-as-a-Service (SaaS).
- It is estimated that the number of servers used by Google was close to 1.8 million in January 2012 and was expected to reach close to 2.4 million in early 2013.
- 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.
- These services are running on a cloud and can be invoked from a broad spectrum of devices, including mobile ones such as iPhones, iPads, Black-Berrys, and laptops and tablets. The data for these services is stored in data centers on the cloud.

The *Gmail* service hosts emails on Google servers and, provides aWeb 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. The service allows users to import and export files in several formats, including Microsoft Office, PDF, text, and OpenOffice extensions.

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. Google Calendar is accessible from mobile devices. Event reminders can be received via SMS, desktop popups, or emails. It is also possible to share your calendar with other Google Calendar users.

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.

Google is also a leader in the Platform-as-a-Service (PaaS) space. 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 GoogleQuery Language (GQL) with a SQL-like syntax.

Search engine crawlers rely on hyperlinks to discover new content. The deep Web is content stored in databases and served as pages created dynamically by querying HTML 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.

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 was a database by Google. People are able to post their content, such as classified ads, recipes, and events, to the Google Base database. 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.

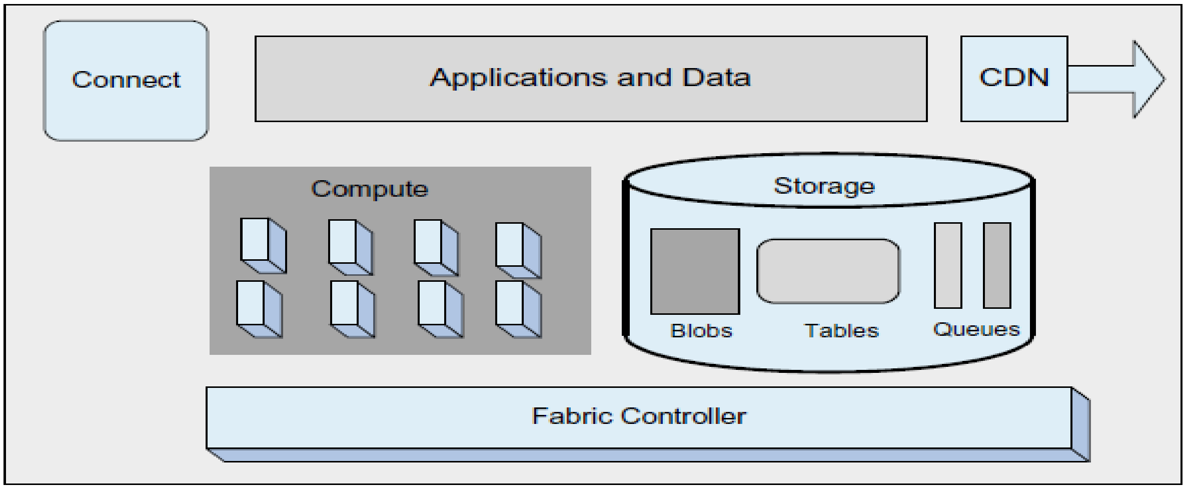
Google Drive is an online service for data storage that has been available since April 2012. It gives users 5 GB of free storage and charges $4 per month for 20 GB. It is available for PCs, MacBooks,iPhones, iPads, and Android devices and allows organizations to purchase up to 16 TB of storage.

Google has also redefined the laptop with the introduction of the Chromebook, a purelyWeb-centric device running Chrome OS. Cloud-based applications, extreme portability, built-in 3G connectivity,almost instant-on, and all-day battery life are the main attractions of this device with a keyboard.

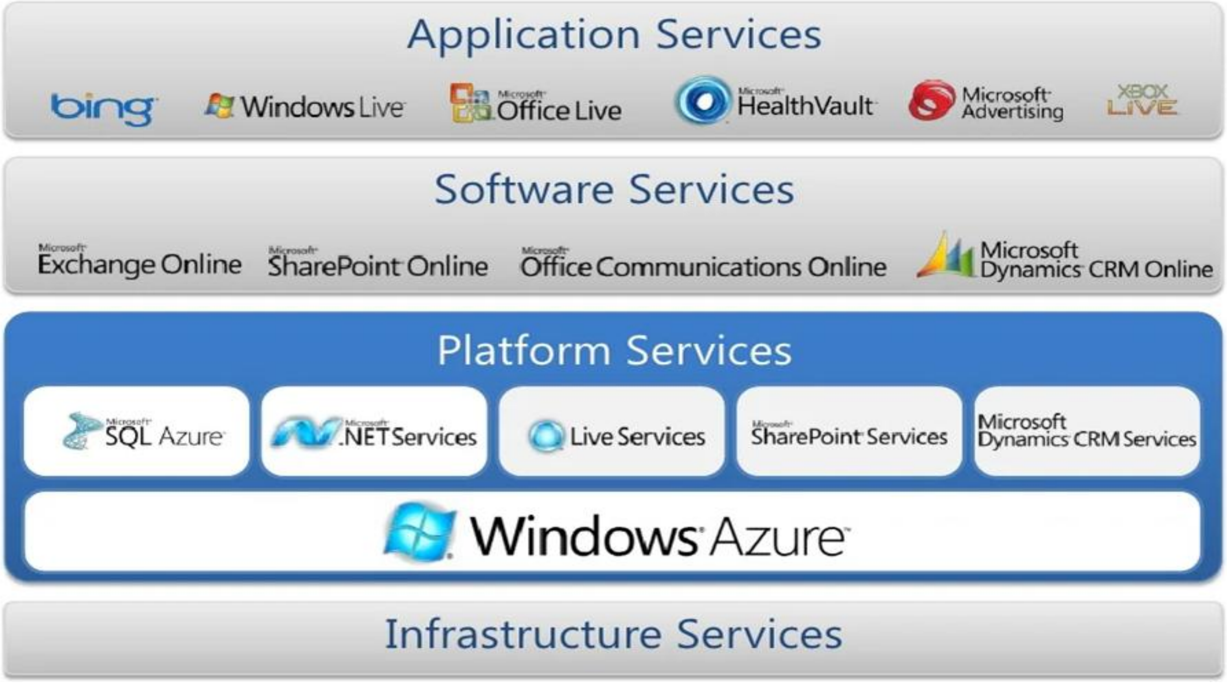
Microsoft Windows Azure and online services

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

Windows Azure has three core components:
Compute, which provides a computation environment; **Storage** for scalable storage; and Fabric **Controller,** which deploys, manages, and monitors applications; it interconnects nodes consisting of servers, high-speed connections, and switches.



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; an application typically runs multiple instances of a role. We can distinguish (i)Web role instances used to create Web applications; (ii) Worker role instances used to run Windows-based code; and (iii) VM role instances that run a user-provided Windows Server 2008 R2 image.

Scaling, load balancing, memory management, and reliability are ensured by a fabric controller, 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.

The Microsoft Azure platform currently does not provide or support any distributed parallel computing frameworks, such as MapReduce, Dryad, or MPI, other than the support for implementing basic queue-based job scheduling

Open-source software platforms for private clouds

A private cloud has essentially the same structural components as a commercial one: the servers, the network, virtual machines monitors (VMMs) running on individual systems, an archive containing disk images of virtual machines (VMs), a front end for communication with the user, and a cloud control infrastructure.

Open source cloud computing platforms such as **Eucalyptus, OpenNebula, and Nimbus** can be used as a control infrastructure for a private cloud.

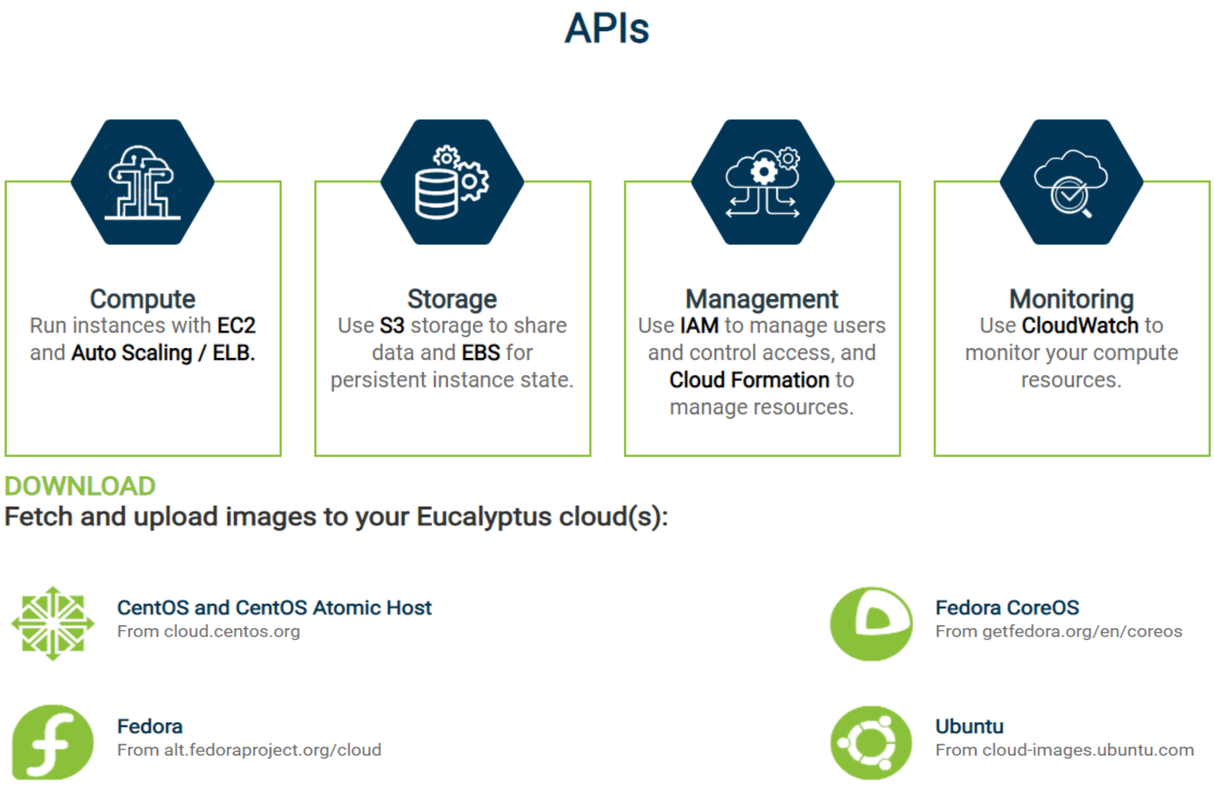
Schematically, a cloud infrastructure carries out the following steps to run an application:

- 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.

Eucalyptus:

Eucalyptus is open source software for building AWS-compatible private and hybrid clouds.

As an Infrastructure as a Service (IaaS) product, Eucalyptus allows your users to provision your compute and storage resources on-demand.



APIs

Compute
Run instances with **EC2** and **Auto Scaling / ELB.**

Storage
Use **S3** storage to share data and **EBS** for persistent instance state.

Management
Use **IAM** to manage users and control access, and **Cloud Formation** to manage resources.

Monitoring
Use **CloudWatch** to monitor your compute resources.

DOWNLOAD
Fetch and upload images to your Eucalyptus cloud(s):

CentOS and CentOS Atomic Host
From cloud.centos.org

Fedora CoreOS
From getfedora.org/en/coreos

Fedora
From alt.fedoraproject.org/cloud

Ubuntu
From cloud-images.ubuntu.com

The systems supports several operating systems including CentOS 5 and 6, RHEL 5 and 6, Ubuntu 10.04 LTS, and 12.04 LTS.

The components of the system are:

- **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.
- **Cloud controller.** Provides the cloud access to end users, developers, and administrators.
- **Storage controller.** Provides persistent virtual hard drives to applications. It is the correspondent of EBS.
- **Storage service (Walrus).** Provides persistent storage and, similarly to S3, allows users to store objects in buckets.

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

- 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) 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 users can now ssh10 directly into the VM.

<u>Open-Nebula:</u>

OpenNebula is a simple, feature-rich and flexible solution for the management of virtualized data centres. It enables private, public and hybrid clouds.

OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures. The OpenNebula platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of Infrastructure as a Service.

The two primary uses of the OpenNebula platform are data center virtualization and cloud deployments based on the KVM hypervisor, LXD/LXC system containers, and AWS Firecracker microVMs.

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

(i) The user signs into the head node using ssh;

(ii) The system uses the on evm command to request a VM;

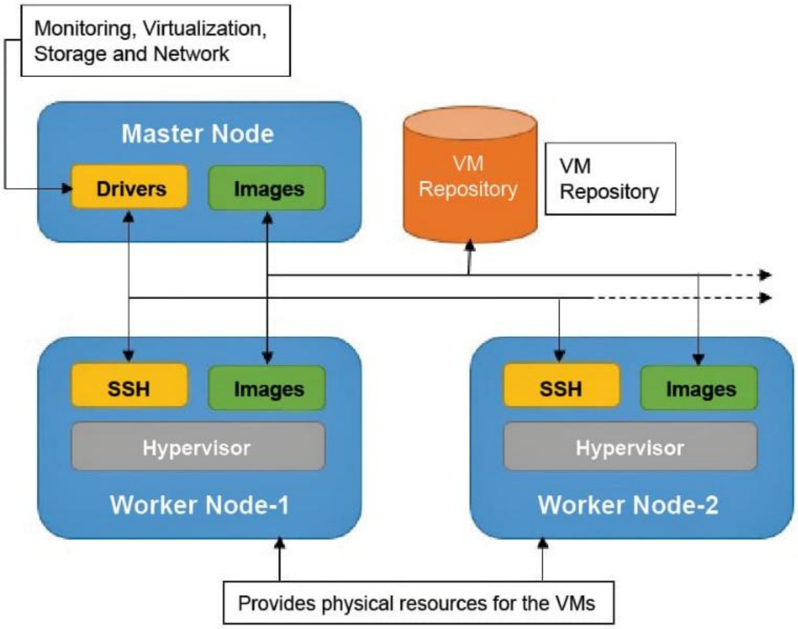
(iii) The VM template disk image is transformed to fit the correct size and configuration within the NFS directory on the head node;

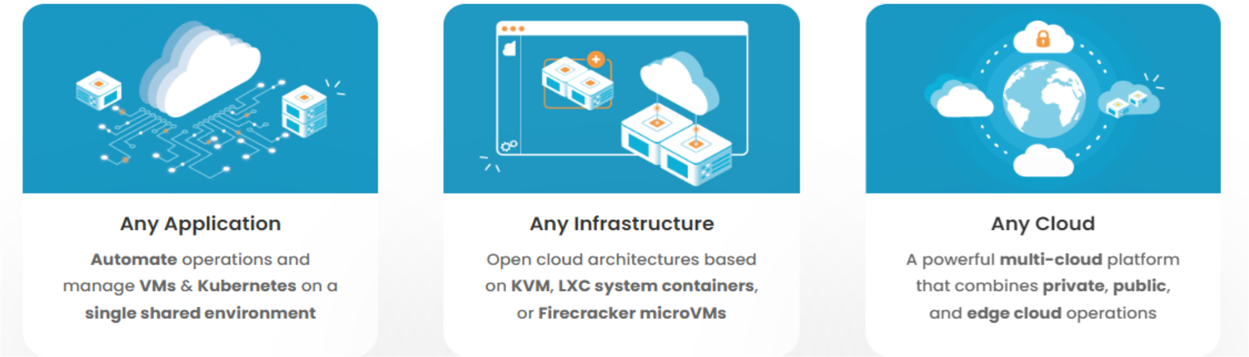
(iv) The oned daemon on the head node uses ssh to log into a compute node;

(v) The compute node sets up network bridging to provide a virtual NIC with a virtual MAC; (vi) the files needed by the VMM are transferred to the compute node via the NFS;

(vii) The VMM on the compute node starts the VM;

(viii) The user is able to ssh directly to the VM on the compute node.





Any Application

Automate operations and manage **VMs** & **Kubernetes** on a **single shared environment**

Any Infrastructure

Open cloud architectures based on **KVM**, **LXC system containers**, or **Firecracker microVMs**

Any Cloud

A powerful **multi-cloud** platform that combines **private**, **public**, and **edge cloud** operations

Nimbus:

- Open-source toolkit converting computer clusters into Infrastructure-as-a-Service cloud to provide computing for science communities.
- Nimbus Platform is an integrated set of tools, operating in a multi-cloud environment that delivers the power and versatility of infrastructure clouds to scientific users.

Nimbus is comprised of two products:

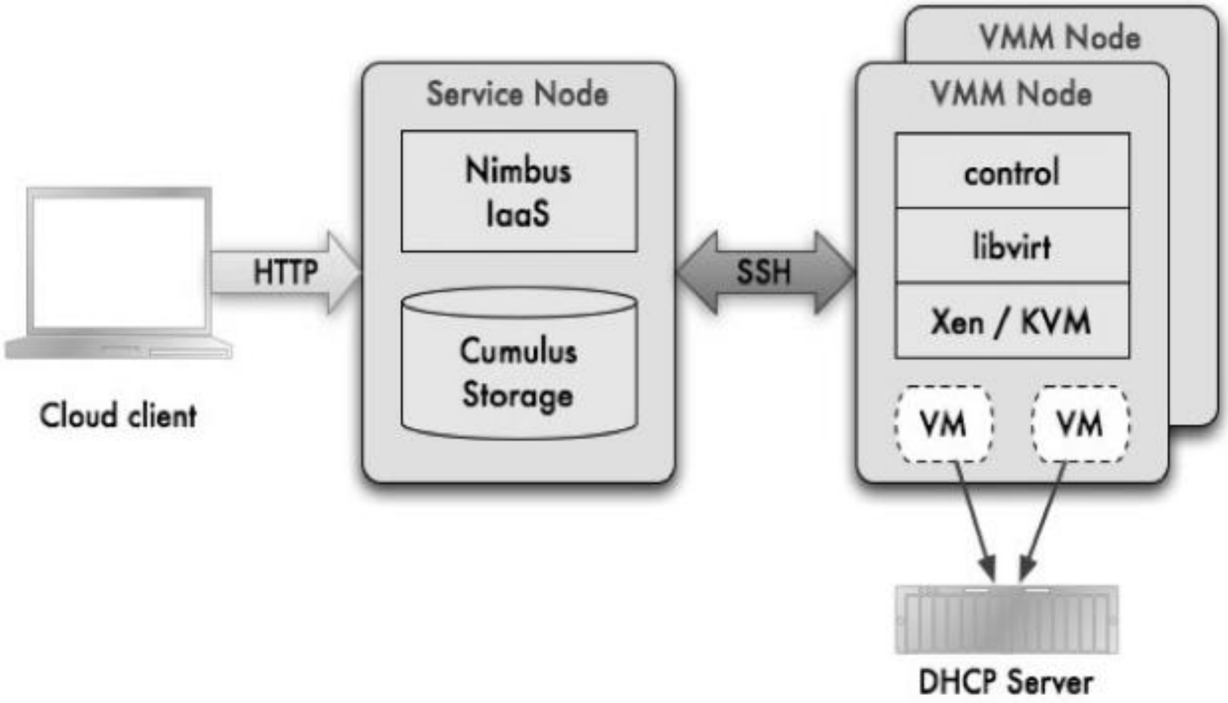
Nimbus Infrastructure is an open source EC2/S3-compatible Infrastructure-as-a-Service implementation specifically targeting features of interest to the scientific community such as support for proxy credentials, batch schedulers, best-effort allocations and others.

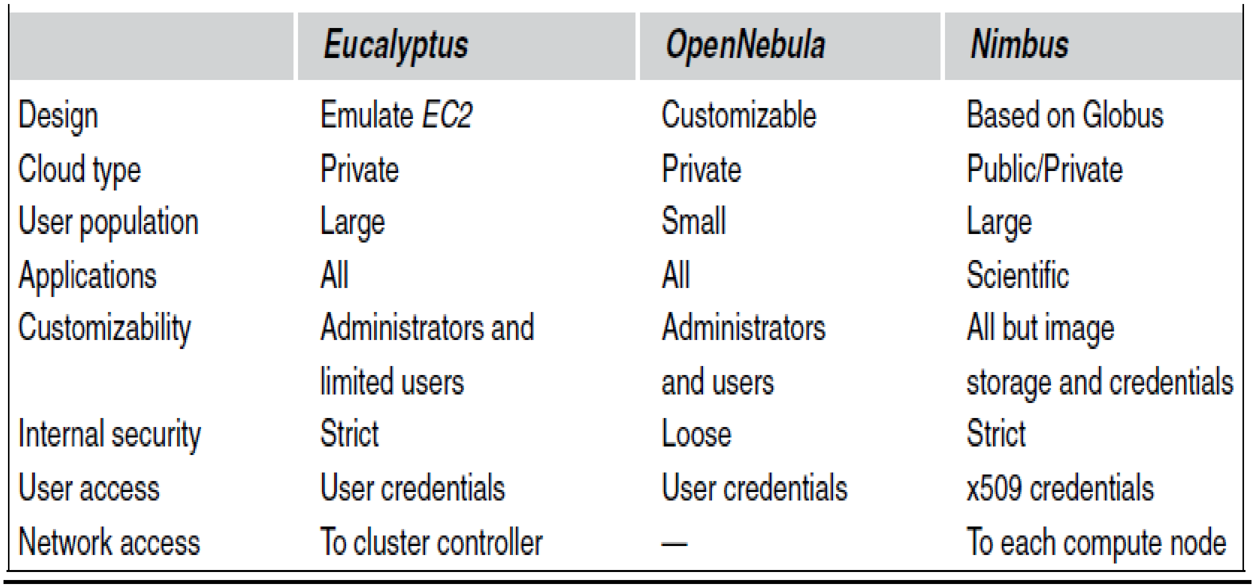
Nimbus Platform allows you to reliably deploy, scale, and manage cloud resources.

The Nimbus cloud client allows the user to provision customized compute nodes, called a workspace, and maintains full control over it using a leasing model based on the Amazon's Elastic Compute Cloud (EC2) service.

The Nimbus cloud-computing infrastructure allows scientists working on data-intensive research to create and use such virtual machines with a cloud provider.

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.





	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.
- Cloud services may be unavailable for a short or even an extended period of time. Cloud Service Provider (CSP) may decide to increase the prices for service and charge more for computing cycles, memory, storage space, and network bandwidth than other CSPs.

How can companies avoid the risks of vendor lock-in?

- Evaluate cloud services carefully
- Ensure data can be moved easily
- Backups
- Multi-cloud or hybrid cloud strategy
- RAID-5 system used for reliable data storage.

A solution to guarding against the problems posed by the vendor lock-in is to replicate the data to multiple cloud service providers. Straightforward replication is very costly and, at the same time, poses technical challenges.

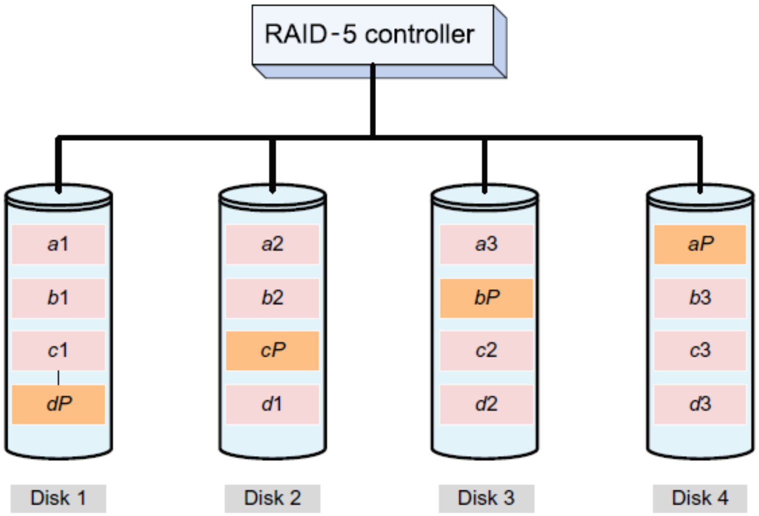
The overhead to maintain data consistency could drastically affect the performance of the virtual storage system consisting of multiple full replicas of the organization’s data spread over multiple vendors.

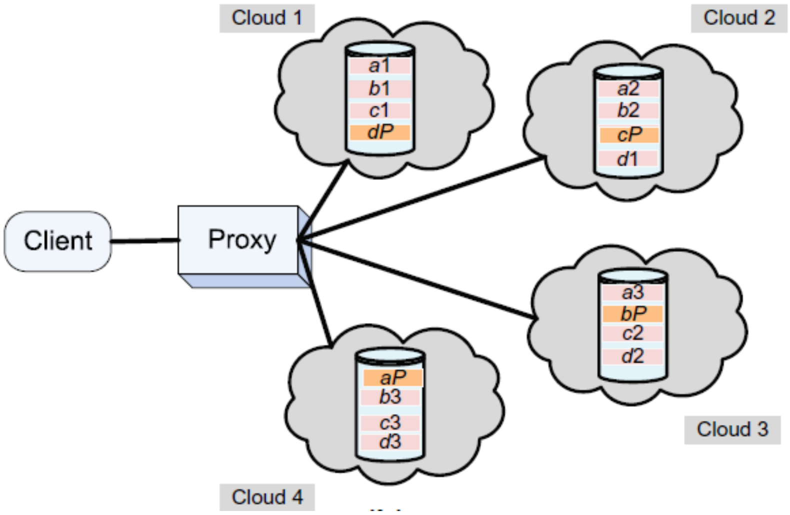
Another solution could be based on an extension of the design principle of a RAID-5 system used for reliable data storage.

A RAID-5 system uses block-level stripping with distributed parity over a disk array, as shown The disk controller distributes the sequential blocks of data to the physical disks and computes a parity block by bit-wise 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, as is done in the case of RAID-4 systems. This technique allows us to recover the data after a single disk loss.

For example, if Disk 2 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:

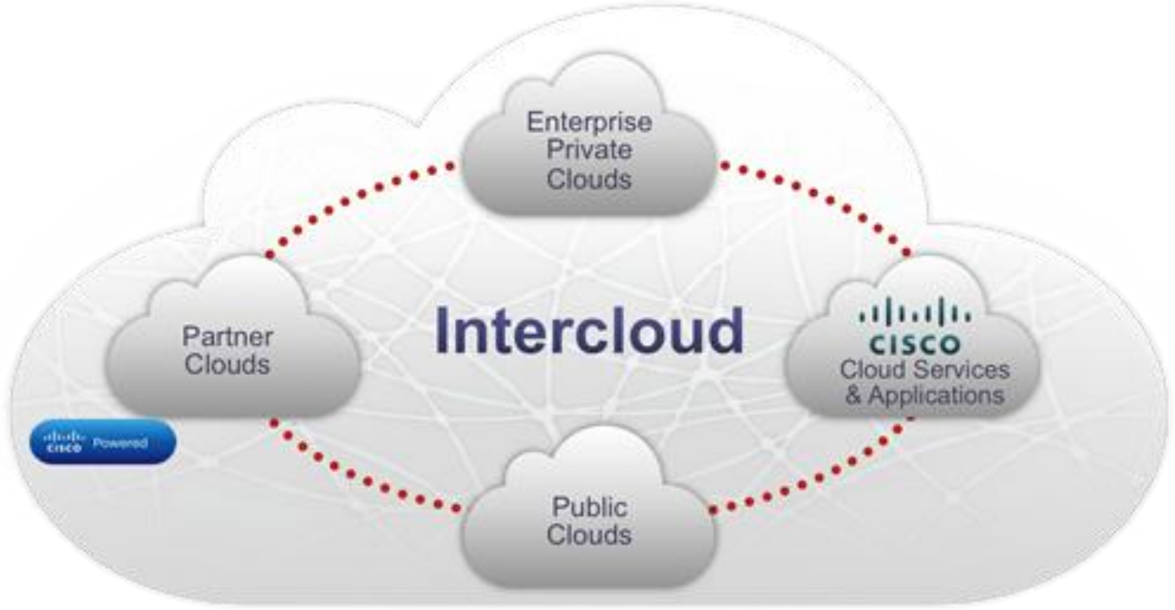
$$\begin{aligned} a2 &= (a1) \text{ XOR } (aP) \text{ XOR } (a3) \\ b2 &= (b1) \text{ XOR } (bP) \text{ XOR } (b3)\ . \\ d1 &= (dP) \text{ XOR } (d2) \text{ XOR } (d3) \end{aligned}$$





Cloud computing interoperability: the Intercloud

Intercloud or ‘cloud of clouds’ is a term refer to a theoretical model for cloud computing services based on the idea of combining many different individual clouds into one seamless mass in terms of on-demand operations.


Enterprise Private Clouds
Partner Clouds
Intercloud
cisco Cloud Services & Applications
cisco Powered
Public Clouds

The situation is quite different in cloud computing. **First**, there are no standards for storage of processing; **second,** the clouds we have seen so far are based on different delivery models: SaaS, PaaS, and IaaS. Moreover, the set of services supported by each of these delivery models is not only large, it is open; new services are offered every few months.

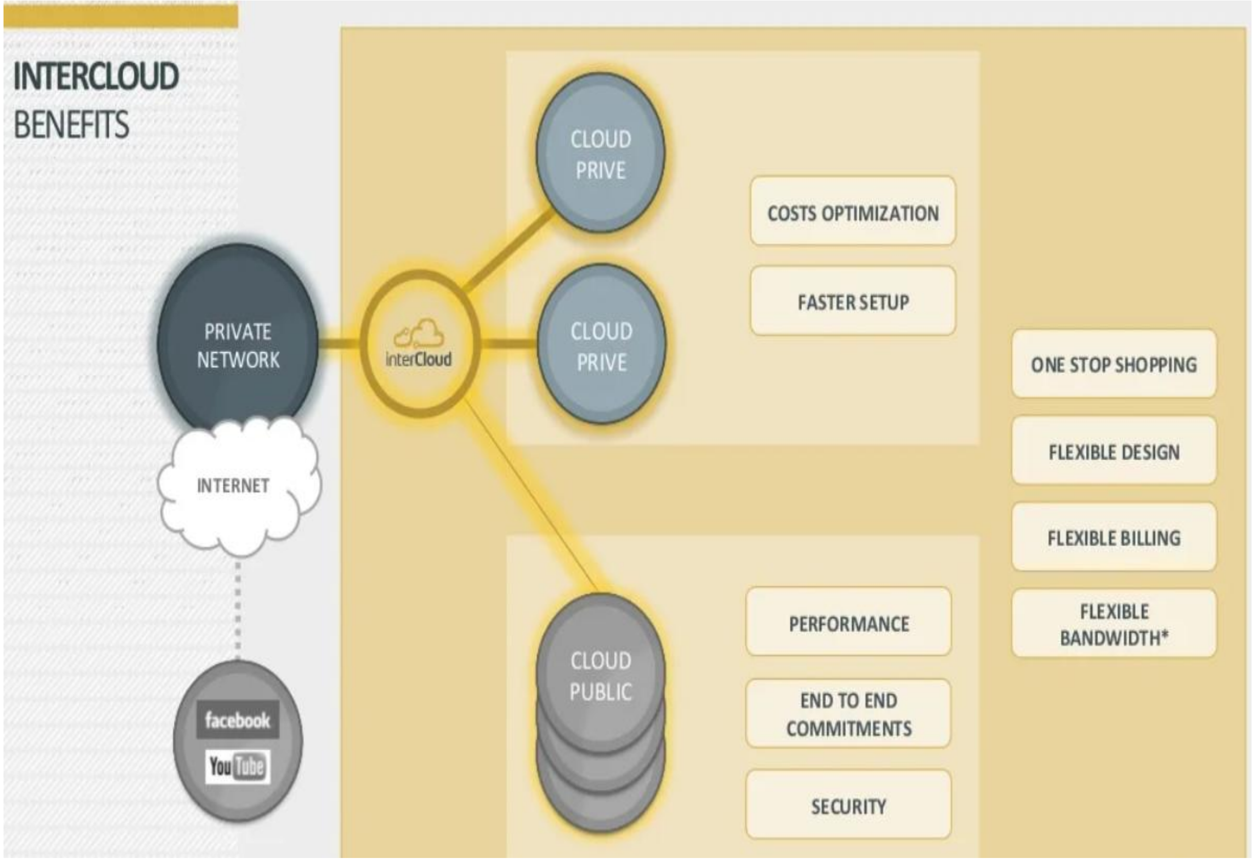
For example, in October 2012 Amazon announced a new service, the AWS GovCloud (US).

An Intercloud would then require the development of an ontology11 for cloud computing. Then each cloud service provider would have to create a description of all resources and services using this ontology.

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.

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 during this migration.

The Public Key Infrastructure (PKI), an all-or-nothing trust model, is not adequate for an Intercloud, where the trust must be nuanced.



<u>Energy use and ecological impact of large-scale data centers</u>

- Cloud Data Centers use an estimated 200 terawatt hours (TWh) each year. This is more than the annual energy consumption of some countries.

- 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.

- By definition, an ideal energy-proportional system is always operating at 100% efficiency. Humans are a good approximation of an ideal energy proportional system; human energy consumption is about 70W at rest and 120W on average on a daily basis and can go as high as 1,000–2,000W during a strenuous, short effort.

- less than 50% for dynamic random access memory (DRAM), 25% for disk drives, and 15% for networking switches.

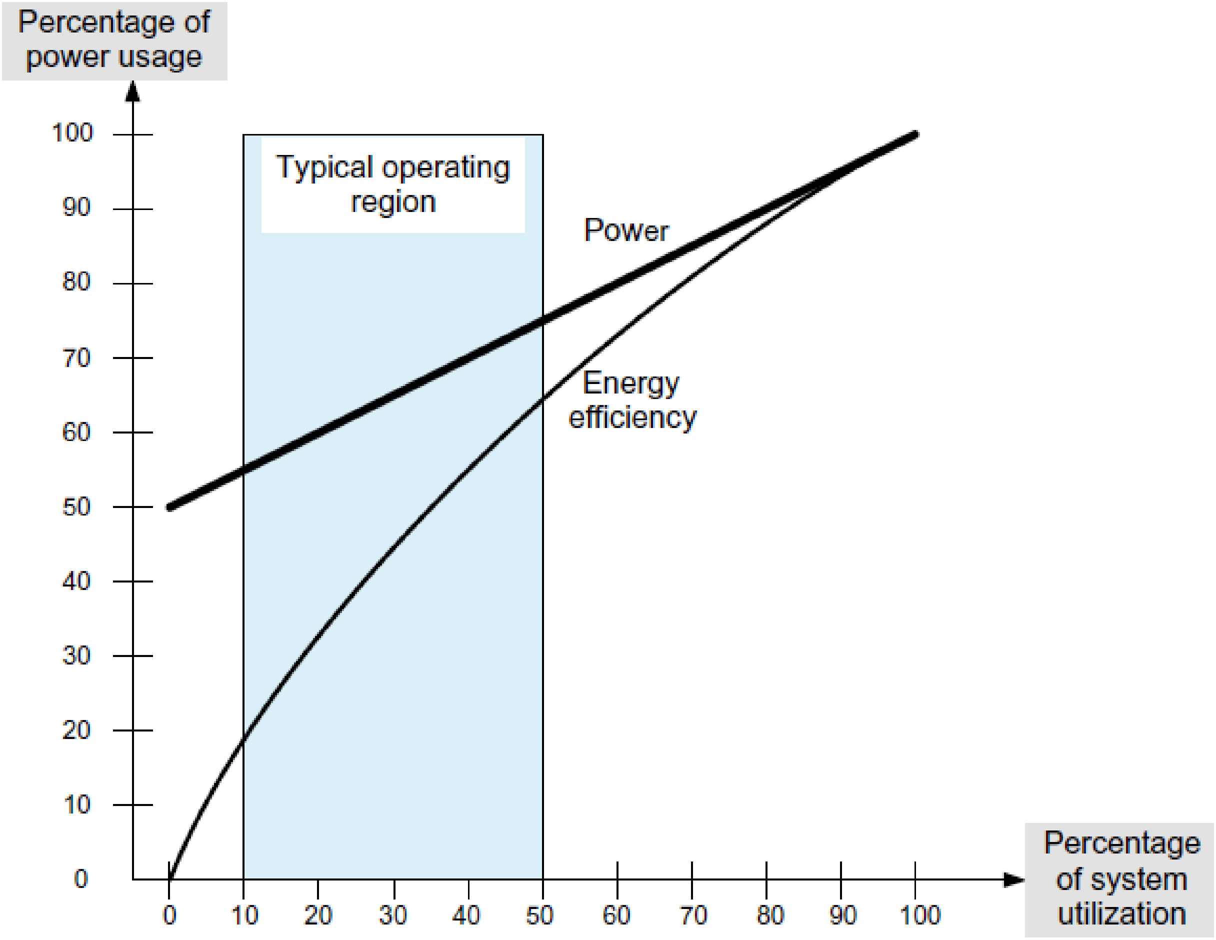
- A number of proposals have emerged for energy-proportional networks; the energy consumed by such networks is proportional to the communication load.

- 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.

- The energy consumption of large-scale data centers and their costs for energy and for cooling are significant now and are expected to increase substantially in the future. In 2006, the 6,000 data centers in the United States reportedly consumed 61×10^9 KWh of energy, 1.5% of all electricity consumption in the country, at a cost of $4.5 billion

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 is presented



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.

Responsibility sharing between user and cloud service provider

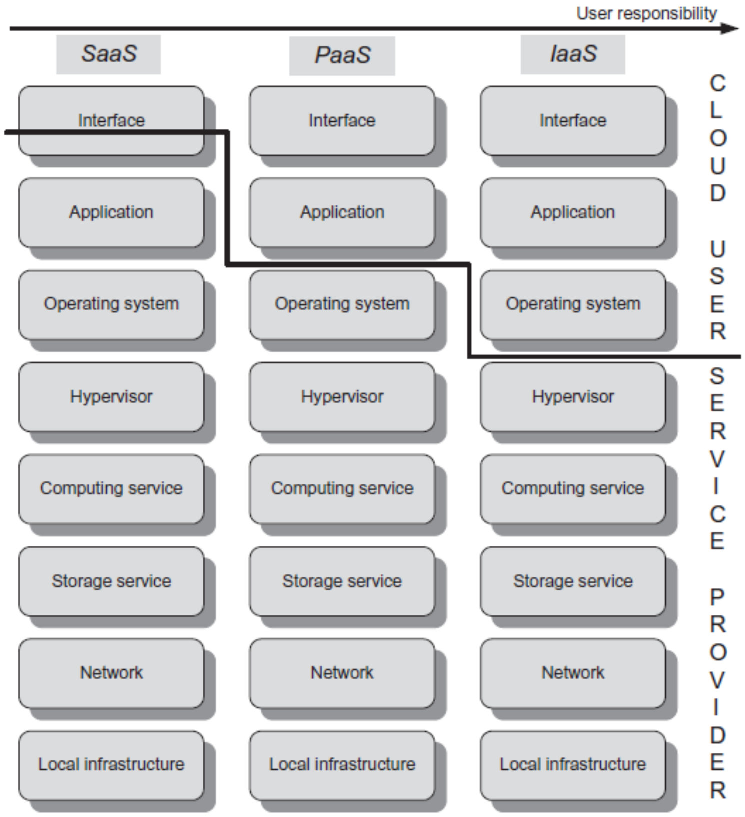
- The service provider supplies both the hardware and the application software, and the user has direct access to these services through a Web interface.

- In the case of IaaS, the service provider supplies the hardware (servers, storage, networks) and system software (operating systems, databases); in addition, the provider ensures system attributes such as security, fault tolerance, and load balancing. The representative of IaaS is Amazon AWS.

- PaaS provides only a platform, including the hardware and system software, such as operating systems and databases. Typical examples are Google App Engine, Microsoft Azure, and Force.com, provided by Salesforce.com.

- Saas, the service provider supplies both the hardware and the application software, and the user has direct access to these services through aWeb interface and has no control over cloud resources. Typical examples are Google with Gmail, Google Docs, Google Calendar, Google Groups, and Picasa and Microsoft with the Online Services.

The level of user control over the system in IaaS is different form PaaS. IaaS provides total control, whereas PaaS typically provides no control. Consequently, IaaS incurs administration costs similar to a traditional computing infrastructure, whereas the administrative costs are virtually zero for PaaS.



User experience

- User experience based on a large population of cloud computing users.
- The main user concerns are security threats, the dependence on fast Internet connections that forced version updates, data ownership, and user behavior monitoring.
- (i) abuse and villainous use of the cloud; (ii) APIs that are not fully secure; (iii) malicious insiders; (iv) account hijacking; (iv) data leaks; and (v) issues related to shared resources.

The suggested solutions to these problems are as follows:

- SLAs and tools to monitor usage should be deployed to prevent abuse of the cloud;
- Data encryption and security testing should enhance the API security;
- an independent security layer should be added to prevent threats caused by malicious insiders;
- Strong authentication and authorization should be enforced to prevent account hijacking;
- Data decryption in a secure environment should be implemented to prevent data leakage;

A broad set of concerns identified by the NIST working group on cloud security includes:

- Potential loss of control/ownership of data.
- Data integration, privacy enforcement, data encryption.
- Data remanence after deprovisioning.
- Multitenant data isolation.
- Data location requirements within national borders.
- Hypervisor security.
- Audit data integrity protection.
- Verification of subscriber policies through provider controls.
- Certification/accreditation requirements for a given cloud service.

The top workloads mentioned by the users involved in this study are data mining and other analytics (83%), application streaming (83%), help desk services (80%), industry-specific applications (80%), and development environments (80%).

The study also identified workloads that are not good candidates for migration to a public cloud environment: 1) Sensitive data such as employee and health care records.2) Multiple codependent services (e.g., online transaction processing). 3)Third-party software without cloud licensing. 4) Workloads requiring auditability and accountability. 5) Workloads requiring customization.

Software Licensing

Software licensing for cloud computing is an enduring problem without a universally accepted solution at this time. The license management technology is based on the old model of computing centers with licenses given on the basis of named users or as site licenses.

IBM reached an agreement allowing some of its software products to be used on EC2. Math Works developed a business model for the use of MATLAB in grid environments. The Software-as-a-Service (SaaS) deployment model is gaining acceptance because it allows users to pay only for the services they use.

The increased negotiating power of users, coupled with the increase in software piracy, has renewed interest in alternative schemes such as those proposed by the SmartLM research project (www.smartlm.eu). SmartLM license management requires a complex software infrastructure involving SLA, negotiation protocols, authentication, and other management functions.

- When a user requests a license from the license service, the terms of the license usage are negotiated and they are part of an SLA document.
- The negotiation is based on application-specific templates and the license cost becomes part of the SLA.
- The SLA describes all aspects of resource usage, including the ID of application, duration, number of processors, and guarantees, such as the maximum cost and deadlines.

Cloud Computing: Applications and Paradigms

The development of efficient cloud applications inherits the challenges posed by the natural imbalance among computing, I/O, and communication bandwidths of physical systems. These challenges are greatly amplified due to the scale of the system, its distributed nature, and the fact that virtually all applications are data-intensive.

- **Performance isolation** - nearly impossible to reach in a real system, especially when the system is heavily loaded.
- **Reliability** - major concern; server failures expected when a large number of servers cooperate for the computations.
- Cloud infrastructure exhibits latency and bandwidth fluctuations which affect the application performance.
- Performance considerations limit the amount of data logging; the ability to identify the source of unexpected results and errors is helped by frequent logging.

Cloud computing is very attractive to the users:

- ➢ Economic reasons.

- low infrastructure investment.
- low cost - customers are only billed for resources used.

Convenience and performance.

- Application developers enjoy the advantages of a just-in-time infrastructure; they are free to design an application without being concerned with the system where the application will run.
- The execution time of compute-intensive and data-intensive applications can, potentially, be reduced through parallelization. If an application can partition the workload in n segments and spawn n instances of itself, then the execution time could be reduced by a factor close to n.

Existing cloud applications and new application opportunities

Existing cloud applications can be divided into several broad categories: (i) processing pipelines; (ii) batch processing systems; and (iii) Web applications

Processing pipelines are data-intensive and sometimes compute-intensive applications and represent a fairly large segment of applications currently running on the cloud.

Several types of data processing applications can be identified:

- **Indexing.** The processing pipeline supports indexing of large datasets created by Web crawler engines.
- **Data mining**. The processing pipeline supports searching very large collections of records to locate items of interests.
- **Image processing.** A number of companies allow users to store their images on the cloud (e.g., Flickr (www.flickr.com) and Google (http://picasa.google.com/)). The image-processing pipelines support image conversion (e.g., enlarging an image or creating thumbnails). They can also be used to compress or encrypt images.
- **Video transcoding.** The processing pipeline transcodes from one video format to another (e.g., from AVI to MPEG).
- **Document processing**. The processing pipeline converts very large collections of documents from one format to another (e.g., from Word to PDF), or encrypts the documents. It could also use optical character recognition (OCR) to produce digital images of documents.

Batch processing systems also cover a broad spectrum of data-intensive applications in enterprise computing. Such applications typically have deadlines, and the failure to meet these deadlines could have serious economic.

Security is also a critical aspect for many applications of batch processing. A nonexhaustive list of batch processing applications includes:

- Generation of daily, weekly, monthly, and annual activity reports for organizations in retail, manufacturing, and other economic sectors.
- Processing, aggregation, and summaries of daily transactions for financial institutions, insurance companies, and healthcare organizations.
- Inventory management for large corporations.
- Processing billing and payroll records.
- Management of the software development (e.g., nightly updates of software repositories).
- Automatic testing and verification of software and hardware systems.

Science and engineering could greatly benefit from cloud computing because many applications in these areas are compute- and data-intensive. Similarly, a cloud dedicated to education would be extremely useful. Mathematical software such as MATLAB and Mathematica could also run on the cloud.

Architectural styles for cloud applications

Cloud computing is based on the client-server paradigm. The vast majority of cloud applications take advantage of request/response communication between clients and stateless servers.

A stateless server does not require a client to first establish a connection to the server. Instead, it views a client request as an independent transaction and responds to it.

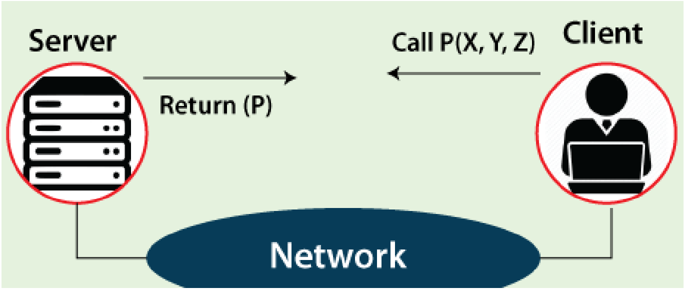
The advantages of stateless servers are obvious. Recovering from a server failure requires considerable overhead for a server that maintains the state of all its connections, whereas in the case of a stateless server a client is not affected while a server goes down and then comes back up between two consecutive requests.

A stateless system is simpler, more robust, and scalable. A client does not have to be concerned with the state of the server. If the client receives a response to a request, that means that the server is up and running; if not, it should resend the request later.

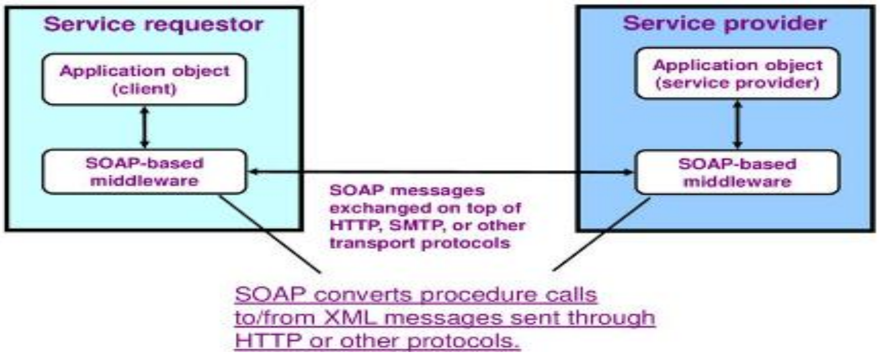
For example, a basic Web server is stateless; it responds to an HTTP request without maintaining a history of past interactions with the client. The client, a browser, is also stateless since it sends requests and waits for responses.

The Hypertext Transfer Protocol (HTTP) used by a browser to communicate with the Web server is a request/response application protocol. HTTP uses the Transport Control Protocol (TCP), a connection-oriented and reliable transport protocol.

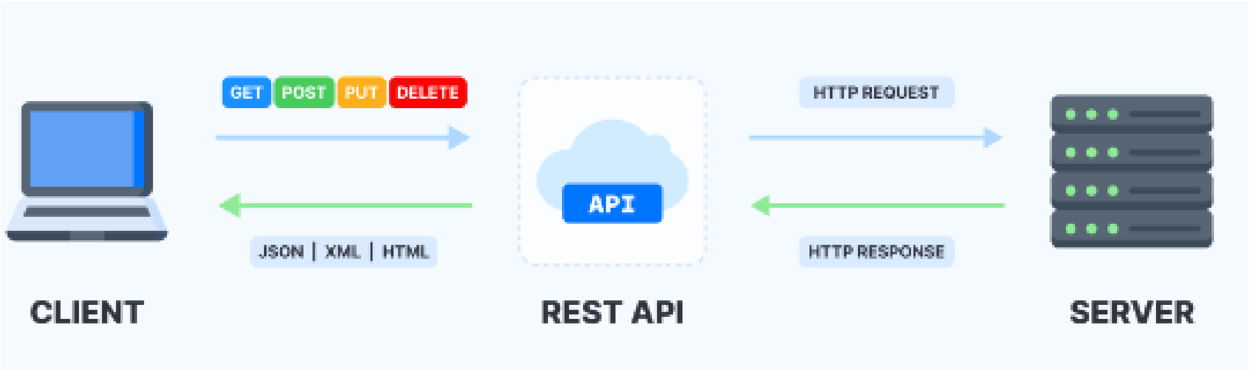
- Clients and servers communicate using Remote Procedure Calls (RPCs).
- Remote Procedure Call is a software communication protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details.



- Simple Object Access Protocol (SOAP) - application protocol for web applications; message format based on the XML. Uses TCP or UDP transport protocols.
- Simple Object Access Protocol (SOAP) is **a lightweight XML-based protocol that is used for the exchange of information.**


Service requestor
Application object (client)
SOAP-based middleware
Service provider
Application object (service provider)
SOAP-based middleware
SOAP messages exchanged on top of HTTP, SMTP, or other transport protocols
SOAP converts procedure calls to/from XML messages sent through HTTP or other protocols.

- Representational State Transfer (REST) - software architecture for distributed hypermedia systems. REST almost always uses HTTP to support all four Create/Read/Update/Delete (CRUD) operations. It uses GET, PUT, and DELETE to read, write, and delete the data, respectively. REST is a much easier-to-use alternative to RPC, CORBA, or Web Services such as SOAP or WSDL.



Workflows: Coordination of multiple activities

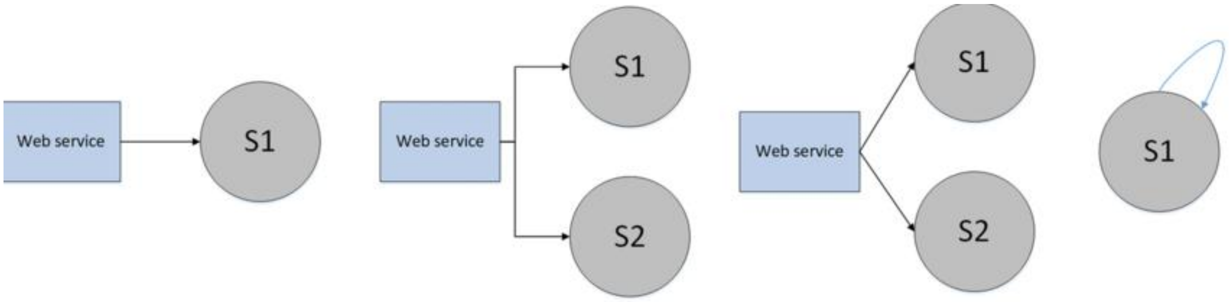
Many cloud applications require the completion of multiple interdependent tasks; the description of a complex activity involving such an ensemble of tasks is known as a workflow.

Workflow models are abstractions revealing the most important properties of the entities participating in a workflow management system. Task is the central concept in workflow modeling; a task is a unit of work to be performed on the cloud, and it is characterized by several attributes, such as:

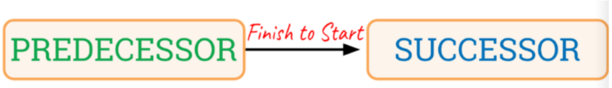
- **Name.** A string of characters uniquely identifying the task.
- **Description.** A natural language description of the task.
- **Actions.** Modifications of the environment caused by the execution of the task.
- **Preconditions.** Boolean expressions that must be true before the action(s) of the task can take place.
- **Post-conditions.** Boolean expressions that must be true after the action(s) of the task take place.
- **Attributes.** Provide indications of the type and quantity of resources necessary for the execution of the task, the actors in charge of the tasks, the security requirements, whether the task is reversible, and other task characteristics.
- **Exceptions.** Provide information on how to handle abnormal events. The exceptions supported by a task consist of a list of <event, action> pairs. The exceptions included in the task exception list are called anticipated exceptions, as opposed to unanticipated exceptions.

- A *composite task* is a structure describing a subset of tasks and the order of their execution.
- A *primitive task* is one that cannot be decomposed into simpler tasks.

A composite task inherits some properties from workflows; it consists of tasks and has one start symbol and possibly several end symbols. At the same time, a composite task inherits some properties from tasks; it has a name, preconditions, and post-conditions.



A **routing task** is a special-purpose task connecting two tasks in a workflow description. The task that has just completed execution is called the predecessor task; the one to be initiated next is called the **successor task**.



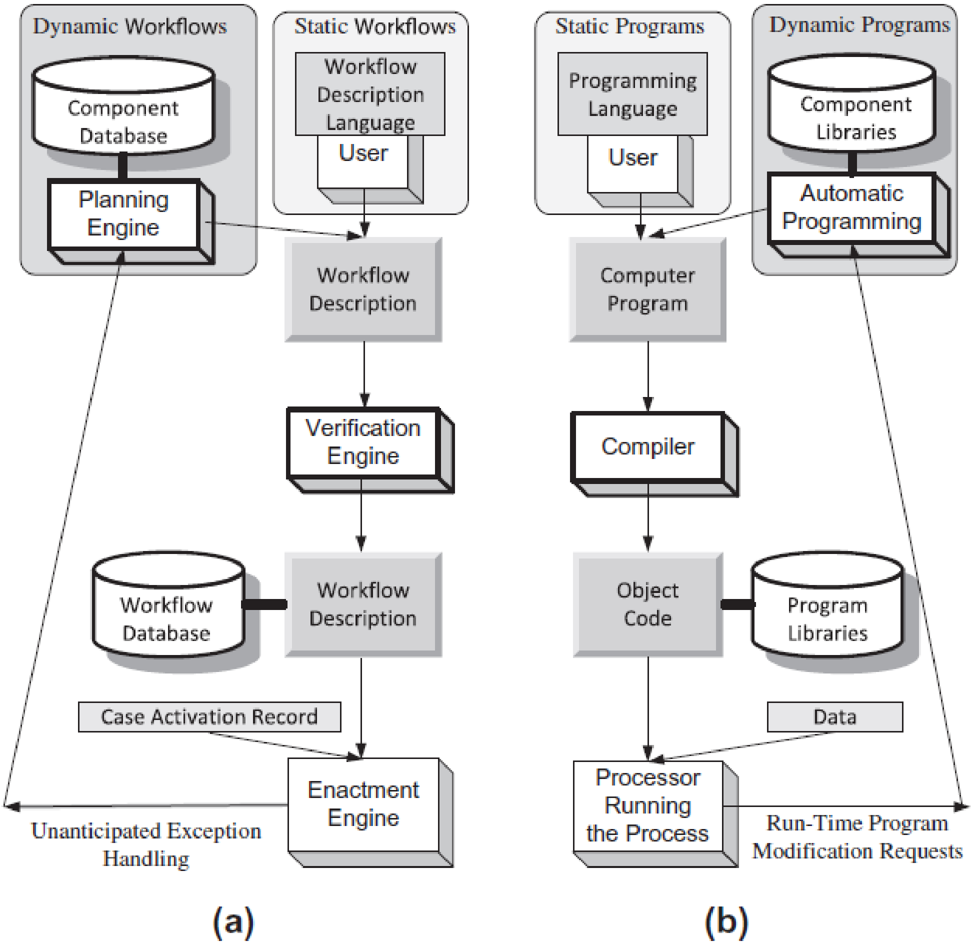
A fork routing task **triggers execution of several successor tasks**.

- Several semantics for this construct are possible: All successor tasks are enabled.
- Each successor task is associated with a condition.

A join routing task **waits for completion of its predecessor tasks**.

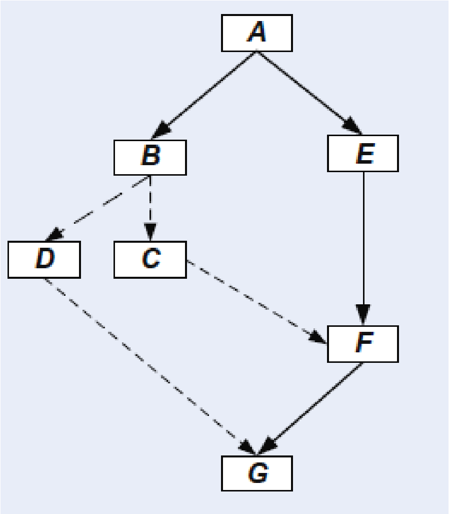
- There are several semantics for the join routing task: The successor is enabled after all predecessors end.
- The successor is enabled after out of predecessors end.
- Iterative: The tasks between the fork and the join are executed repeatedly.

A process description, also called a workflow schema, is a structure describing the tasks or activities to be executed and the order of their execution. A process description contains one start symbol and one end symbol. A process description can be provided in a workflow definition language (WFDL), supporting constructs for choice, concurrent execution, the classical fork, join constructs, and iterative execution.

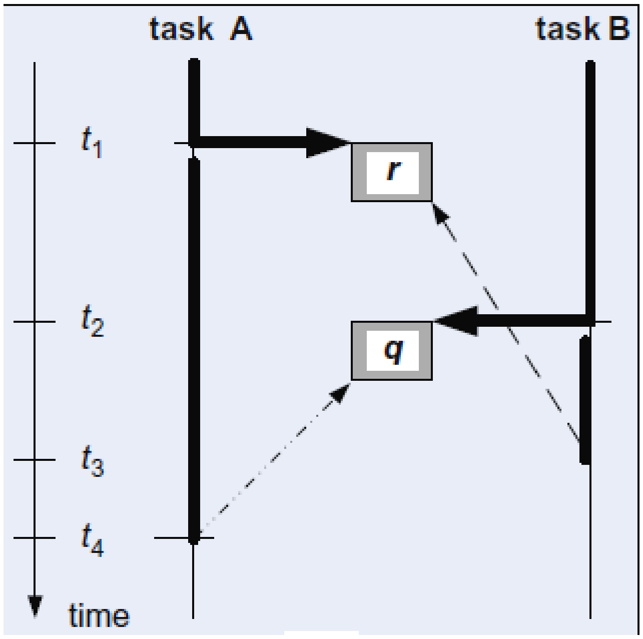


A parallel between workflows and programs. (a) The life cycle of a workflow. (b) The life cycle of a computer program.

Not all processes are safe and live. For example, the process description violates the liveness requirement. As long as task C is chosen after completion of B, the process will terminate. However, if D is chosen, then F will never be instantiated, because it requires the completion of both C and E. The process will never terminate, because G requires completion of both D and F.



Although the original description of a process could be live, the actual enactment of a case may be affected by deadlocks due to resource allocation. To illustrate this situation, consider two tasks, A and B, running concurrently. Each of them needs exclusive access to resources r and q for a period of time.

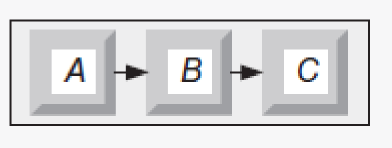


Tasks A and B need exclusive access to two resources r and q, and a deadlock may take place if the following sequence of events occurs. At time t1 task A acquires r, at time t2 task B acquires q and continues to run; then at time t3 task B attempts to acquire r and it blocks because r is under the control of A. Task A continues to run and at time t4 attempts to acquire q and it blocks because q is under the control of B.

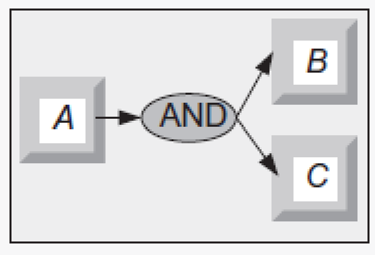
Workflow pattern:

Workflow pattern refers to the temporal relationship among the tasks of a process. The workflow description languages and the mechanisms to control the enactment of a case must have provisions to support these temporal relationships.

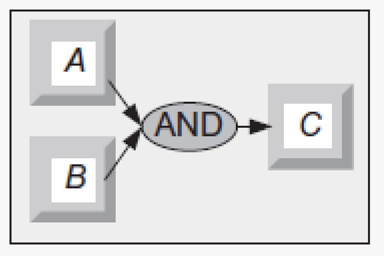
- The **sequence pattern** occurs when several tasks have to be scheduled one after the completion of the other:



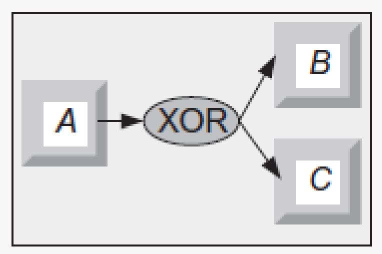
- The AND split pattern requires several tasks to be executed concurrently. Both tasks B and C are activated when task A terminates. In case of an explicit AND split, the activity graph has a routing node and all activities connected to the routing node are activated as soon as the flow of control reaches the routing node.



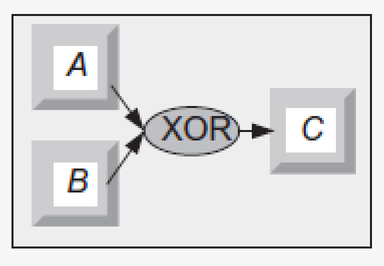
- The synchronization pattern requires several concurrent activities to terminate before an activity can start. In our example, task C can only start after both tasks A and B terminate.



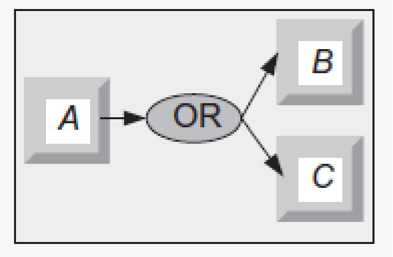
- The XOR split requires a decision; after the completion of task A, either B or C can be activated



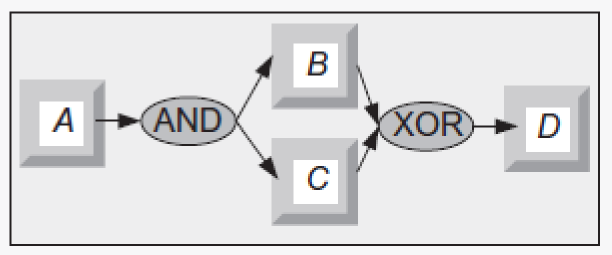
- In the XOR join, several alternatives are merged into one. In our example, task C is enabled when either A or B terminates.



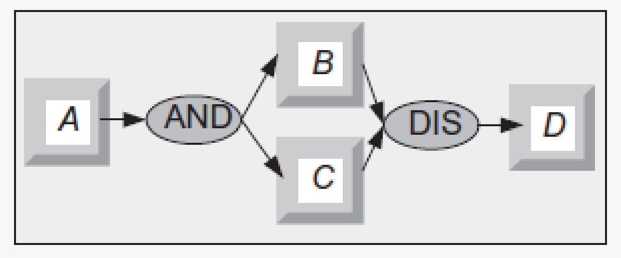
- The OR split pattern is a construct to choose multiple alternatives out of a set. In our example, after completion of task A, one could activate either B or C, or both.



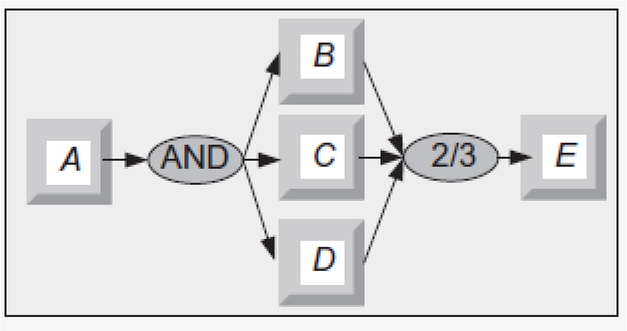
- The multiple merge construct allows multiple activations of a task and does not require synchronization after the execution of concurrent tasks. Once A terminates, tasks B and C execute concurrently



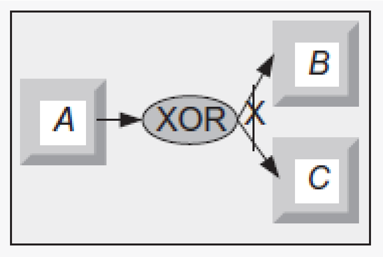
- The discriminator pattern waits for a number of incoming branches to complete before activating the subsequent activity ; then it waits for the remaining branches to finish without taking any action until all of them have terminated. Next, it resets itself.



- The N out of M join construct provides a barrier synchronization. Assuming that M > N tasks run concurrently, N of them have to reach the barrier before the next task is enabled. In our example, any two out of the three tasks A, B, and C have to finish before E is enabled.



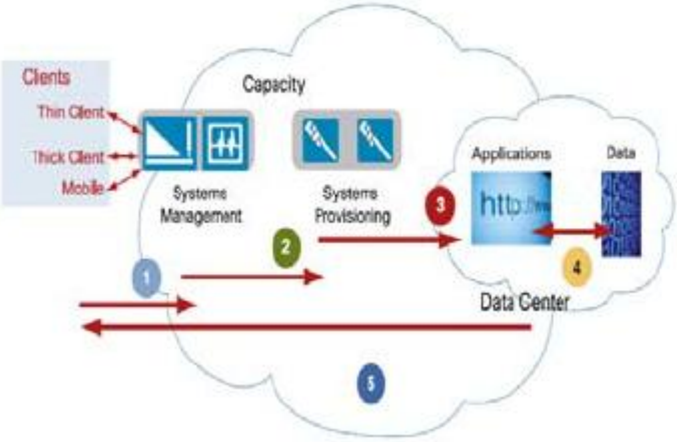
- The deferred choice pattern is similar to the XOR split, but this time the choice is not made explicitly and the run-time environment decides what branch to take.



Some workflows are static. The activity graph does not change during the enactment of a case. Dynamic workflows are those that allow the activity graph to be modified during the enactment of a case.

Two basic models for the mechanics of workflow enactment:

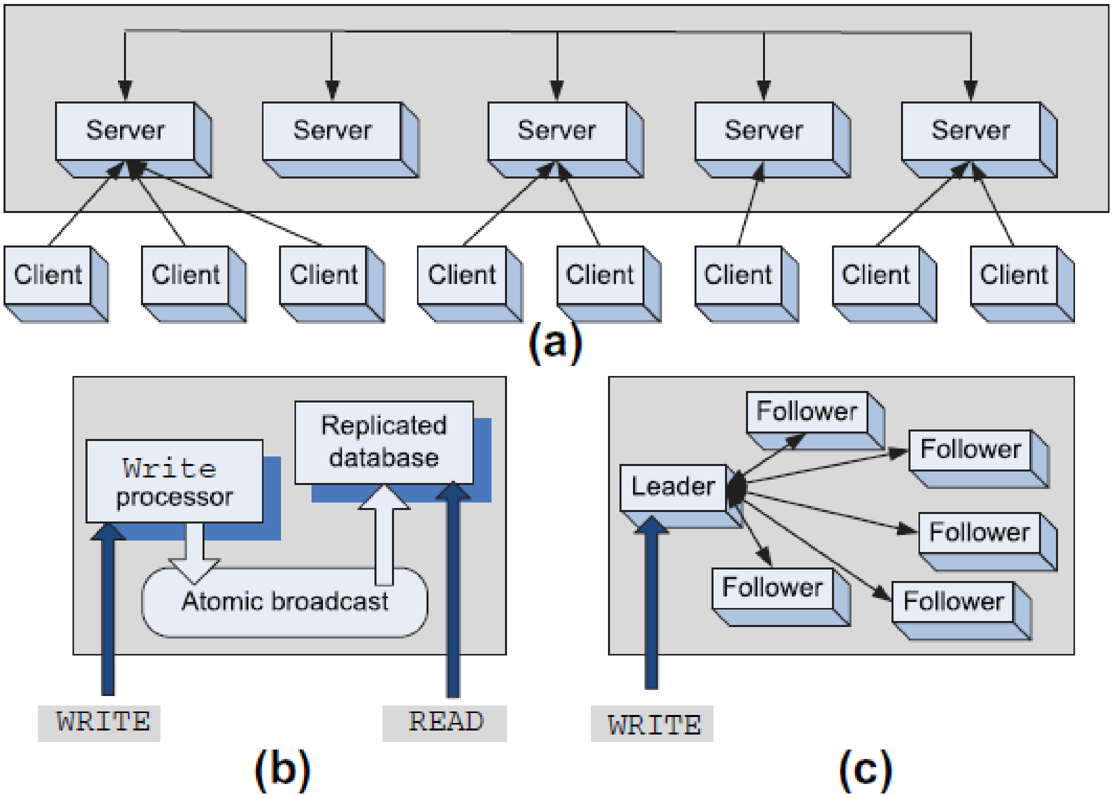
1. Strong coordination models, whereby the process group P executes under the supervision of a coordinator process or processes. A coordinator process acts as an enactment engine and ensures seamless transition from one process to another in the activity graph.
2. Weak coordination models, whereby there is no supervisory process.



Coordination based on a state machine model: The *ZooKeeper*

Cloud computing elasticity requires the ability to distribute computations and data across multiple systems. Coordination among these systems is one of the critical functions to be exercised in a distributed environment.

- ZooKeeper is a distributed co-ordination service to manage large set of hosts. Co-ordinating and managing a service in a distributed environment is a complicated process.
- ZooKeeper solves this issue with its simple architecture and API. ZooKeeper allows developers to focus on core application logic without worrying about the distributed nature of the application.
- A group of systems in which a distributed application is running is called a **Cluster** and each machine running in a cluster is called a **Node**.
- A distributed application has two parts, Server and Client application.
- Server applications are actually distributed and have a common interface so that clients can connect to any server in the cluster and get the same result.



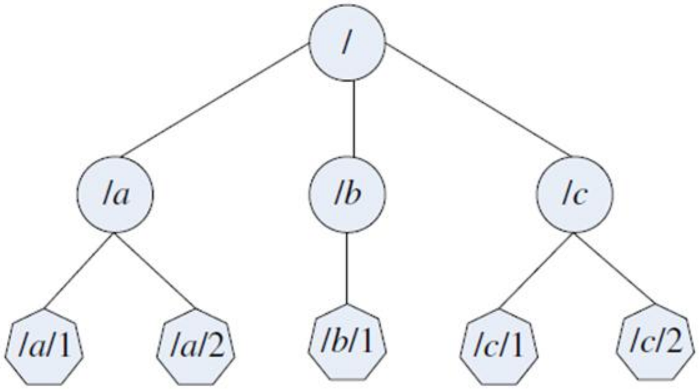
The *ZooKeeper* coordination service. (a) The service provides a single system image. Clients can connect to any server in the pack. (b) Functional model of the *ZooKeeper* service. The replicated database is accessed directly by `read` commands; `write` commands involve more intricate processing based on atomic broadcast. (c) Processing a `write` command: (1) A server receiving the command from a client forwards the command to the *leader*; (2) the *leader* uses atomic broadcast to reach consensus among all *followers*.

Figures (b) and (c) show that a read operation directed to any server in the pack returns the same result, whereas the processing of a write operation is more involved; the servers elect a leader, and any follower receiving a request from one of the clients connected to it forwards it to the leader.

The leader uses atomic broadcast to reach consensus. When the leader fails, the servers elect a new leader. The system is organized as a shared hierarchical namespace similar to the organization of a file system. A name is a sequence of path elements separated by a backslash. Every name in Zookeper's namespace is identified by a unique path.

Hierarchical Namespace:

The tree structure of ZooKeeper file system used for memory representation. ZooKeeper node is referred as **znode**. Every znode is identified by a name and separated by a sequence of path (/).



The data stored in each node is read and written atomically. A read returns all the data stored in a *znode*, whereas a write replaces all the data in the *znode*. Unlike in a file system, *Zookeeper* data, the image of the state, is stored in the server memory. Updates are logged to disk for recoverability, and *writes* are serialized to disk before they are applied to the in-memory database that contains the entire tree.

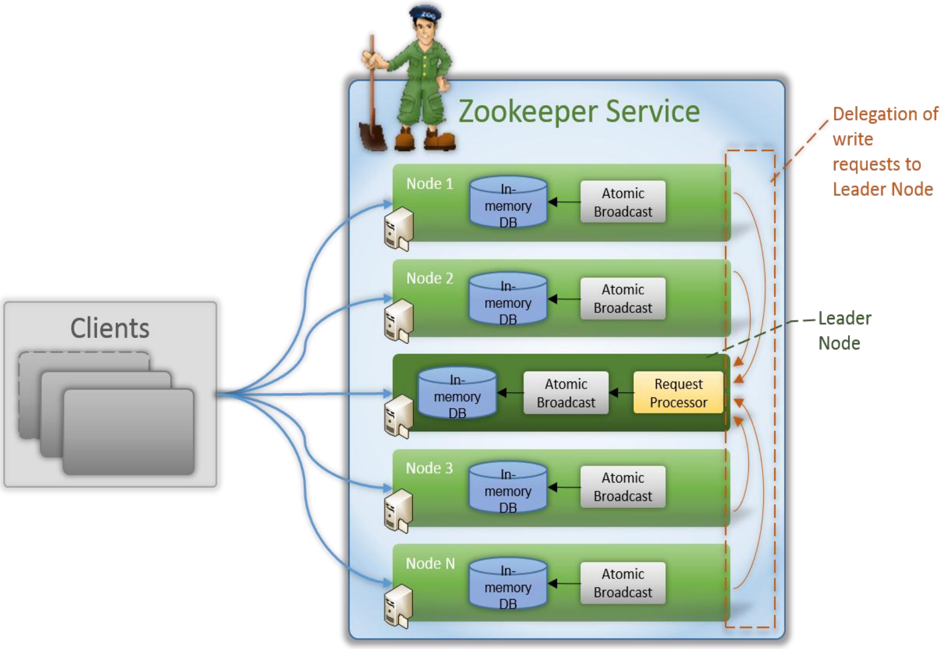
The *ZooKeeper* service guarantees:

1. **Atomicity.** A transaction either completes or fails.
2. **Sequential consistency of updates.** Updates are applied strictly in the order in which they are received.
3. **Single system image for the clients.** A client receives the same response regardless of the server it connects to.
4. **Persistence of updates.** Once applied, an update persists until it is overwritten by a client.
5. **Reliability.** The system is guaranteed to function correctly as long as the majority of servers function correctly.

The application programming interface (API) to the ZooKeeper service is very simple and consists of seven operations:

- create – add a node at a given location on the tree.
- delete – delete a node.
- get data – read data from a node.
- set data – write data to a node.
- get children – retrieve a list of the children of the node.
- synch – wait for the data to propagate.

ZooKeeper is a distributed coordination service based on this model. The high-throughput and low-latency service is used for coordination in large-scale distributed systems. The open-source software is written in Java and has bindings for Java and C. Information about the project is available at http://zookeeper.apache.org/.



High-performance computing on a cloud

- HPC, or supercomputing, is like everyday computing, only more powerful.
- It is a way of processing huge volumes of data at very high speeds using multiple computers and storage devices as a cohesive fabric.
- HPC makes it possible to explore and find answers to some of the world's biggest problems in science, engineering, and business.

Community AtmosphereMode (CAM), the atmospheric component of Community Climate System Model (CCSM), is used for weather and climate modeling. The code developed at NCAR uses two two-dimensional domain decompositions – one for the dynamics and the other for remapping.

General Atomic and Molecular Electronic Structure System (GAMESS) is used for ab initio quantum chemistry calculations. The code, developed by the Gordon Research Group at the U.S.Department of Energy's Ames Lab at Iowa State University, has its own communication library, the Distributed Data Interface (DDI), and is based on the same program multiple data (SPMD) execution model.

Gyrokinetic (GTC) is a code for fusion research.15 It is a self-consistent, gyrokinetic tridimensional particle-in-cell (PIC) code with a non spectral Poisson solver. It uses a grid that follows the field lines as they twist around a toroidal geometry representing a magnetically confined toroidal fusion plasma. The version of GTC used at NERSC uses a fixed, one-dimensional domain decomposition with 64 domains and 64MPItasks.

IntegratedMapandParticle Accelerator Tracking Time (IMPACT-T) is a code for the prediction and performance enhancement of accelerators. It models the arbitrary overlap of fields from beamline elements and uses a parallel, relativistic PIC method with a spectral integrated Green function solver.

This object-oriented Fortran90 code uses a two-dimensional domain decomposition in the y−z directions and dynamic load balancing based on the domains. Hockney's Fast Fourier Transform (FFT) algorithm is used to solve Poisson's equation with open boundary conditions.

MAESTRO is a low Mach number hydrodynamics code for simulating astrophysical flows. Its integration scheme is embedded in an adaptive mesh refinement algorithm based on a hierarchical system of rectangular.

MIMD Lattice Computation (MILC) is a Quantum Chromo Dynamics (QCD) code used to study "strong" interactions binding quarks into protons and neutrons and holding them together in the nucleus.

PARAllel Total Energy Code (PARATEC) is a quantum mechanics code that performs ab initio total energy calculations using pseudo-potentials, a plane wave basis set, and an all-band (unconstrained) Conjugate Gradient (CG) approach. Parallel three-dimensional FFTs transform the wave functions between real and Fourier space.