An Introduction to Cloud Computing

Nan Berman

First Edition, 2012

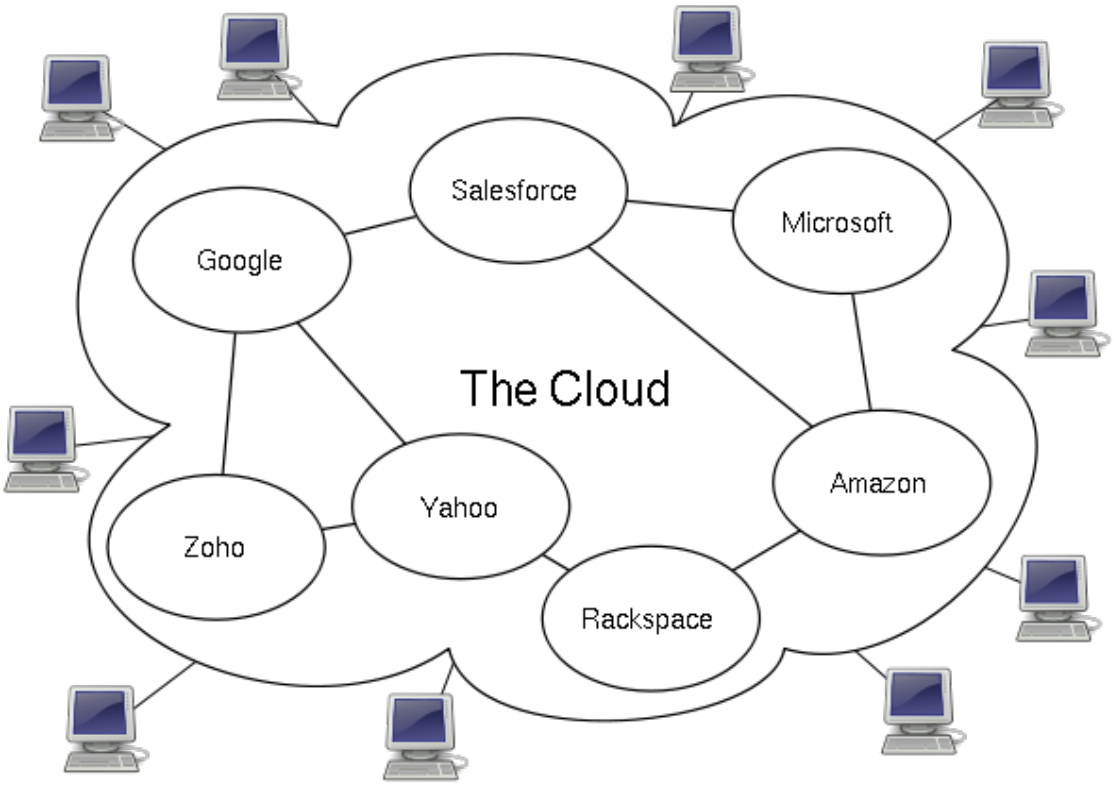
ISBN 978-81-323-4068-3

Published by:
White Word Publications
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter 1

Cloud Computing



Cloud computing conceptual diagram

Cloud computing refers to the provision of computational resources on demand via a computer network.

Cloud computing can be compared to the supply of electricity and gas, or the provision of telephone, television and postal services. All of these services are presented to the users in a simple way that is easy to understand without the users needing to know how the services are provided. This simplified view is called an abstraction. Similarly, cloud computing offers computer application developers and users an abstract view of services

that simplifies and ignores much of the details and inner workings. A provider's offering of abstracted Internet services is often called "The Cloud".

How it works

When a user accesses the cloud for a popular website, many things can happen. The user's IP address, for example, can be used to establish where the user is located (geolocation). DNS services can then direct the user to a cluster of servers that are close to the user so the site can be accessed rapidly and in the user's local language. Users do not log in to a server, but they log in to the service they are using by obtaining a session id or a cookie, which is stored in their browser.

What the user sees in the browser usually comes from a cluster of web servers. The web servers run user interface software which collects commands from the user (mouse clicks, key presses, uploads, etc.) and interprets them. Information is then stored on or retrieved from the database servers or file servers and an updated page is displayed to the user. The data across the multiple servers is synchronised around the world for rapid global access.

Technical description

Cloud computing is computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. Parallels to this concept can be drawn with the electricity grid where end-users consume power resources without any necessary understanding of the component devices in the grid required to provide the service.

Cloud computing describes a new supplement, consumption, and delivery model for IT services based on Internet protocols, and it typically involves provisioning of dynamically scalable and often virtualized resources. It is a byproduct and consequence of the ease-of-access to remote computing sites provided by the Internet. This frequently takes the form of web-based tools or applications that users can access and use through a web browser as if they were programs installed locally on their own computers.

The National Institute of Standards and Technology (NIST) provides a somewhat more objective and specific definition:

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Typical cloud computing providers deliver common business applications online that are accessed from another Web service or software like a Web browser, while the software and data are stored on servers.

Most cloud computing infrastructures consist of services delivered through common centers and built-on servers. Clouds often appear as single points of access for consumers' computing needs. Commercial offerings are generally expected to meet quality of service (QoS) requirements of customers, and typically include service level agreements (SLAs).

Overview

Comparisons

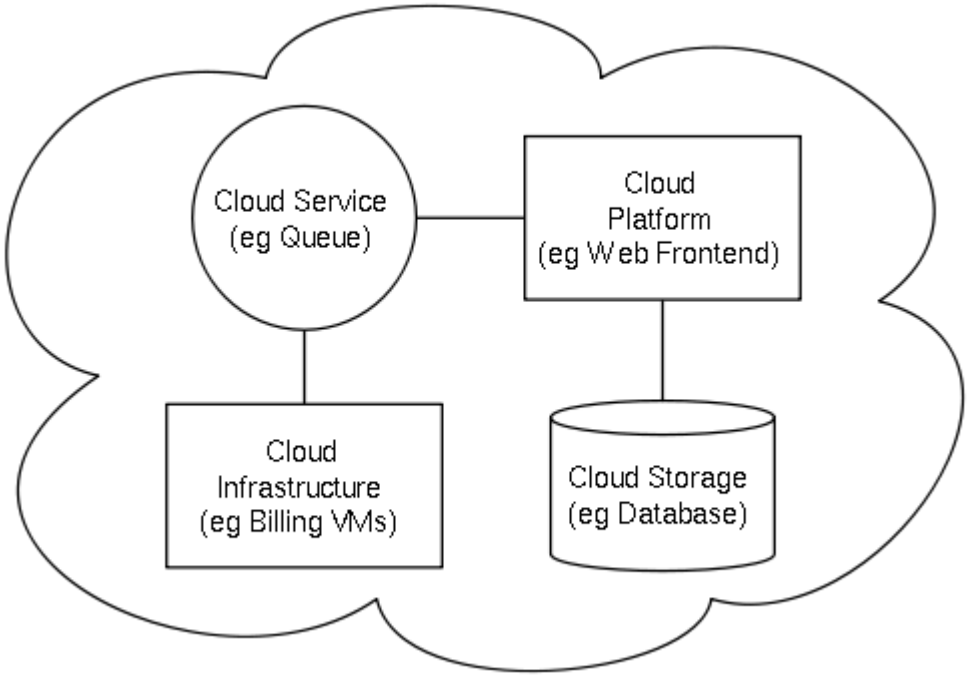
Cloud computing derives characteristics from, but should not be confused with:

1. Autonomic computing — "computer systems capable of self-management."
2. Client–server model – *client–server computing* refers broadly to any distributed application that distinguishes between service providers (servers) and service requesters (clients).
3. Grid computing — "a form of distributed computing and parallel computing, whereby a 'super and virtual computer' is composed of a cluster of networked, loosely coupled computers acting in concert to perform very large tasks."
4. Mainframe computer — powerful computers used mainly by large organizations for critical applications, typically bulk data processing such as census, industry and consumer statistics, enterprise resource planning, and financial transaction processing.
5. Utility computing — the "packaging of computing resources, such as computation and storage, as a metered service similar to a traditional public utility, such as electricity."
6. Peer-to-peer – distributed architecture without the need for central coordination, with participants being at the same time both suppliers and consumers of resources (in contrast to the traditional client–server model).
7. Service-oriented computing – Cloud computing provides services related to computing while, in a reciprocal manner, service-oriented computing consists of the computing techniques that operate on software-as-a-service.

Characteristics

The key characteristic of cloud computing is that the computing is "in the cloud"; that is, the processing (and the related data) is not in a specified, known or static place(s). This is in contrast to a model in which the processing takes place in one or more specific servers that are known. All the other concepts mentioned are supplementary or complementary to this concept.

Architecture



Cloud computing sample architecture

Cloud architecture, the systems architecture of the software systems involved in the delivery of cloud computing, typically involves multiple *cloud components* communicating with each other over application programming interfaces, usually web services and 3-tier architecture. This resembles the Unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces. Complexity is controlled and the resulting systems are more manageable than their monolithic counterparts.

The two most significant components of cloud computing architecture are known as the front end and the back end. The front end is the part seen by the client, i.e. the computer user. This includes the client's network (or computer) and the applications used to access the cloud via a user interface such as a web browser. The back end of the cloud computing architecture is the 'cloud' itself, comprising various computers, servers and data storage devices.

History

The term "cloud" is used as a metaphor for the Internet, based on the cloud drawing used in the past to represent the telephone network, and later to depict the Internet in computer network diagrams as an abstraction of the underlying infrastructure it represents.

Cloud computing is a natural evolution of the widespread adoption of virtualization, service-oriented architecture, autonomic and utility computing. Details are abstracted from end-users, who no longer have need for expertise in, or control over, the technology infrastructure "in the cloud" that supports them.

The underlying concept of cloud computing dates back to the 1960s, when John McCarthy opined that "computation may someday be organized as a public utility." Almost all the modern-day characteristics of cloud computing (elastic provision, provided as a utility, online, illusion of infinite supply), the comparison to the electricity industry and the use of public, private, government and community forms, were thoroughly explored in Douglas Parkhill's 1966 book, *The Challenge of the Computer Utility*.

The actual term "cloud" borrows from telephony in that telecommunications companies, who until the 1990s primarily offered dedicated point-to-point data circuits, began offering Virtual Private Network (VPN) services with comparable quality of service but at a much lower cost. By switching traffic to balance utilization as they saw fit, they were able to utilize their overall network bandwidth more effectively. The cloud symbol was used to denote the demarcation point between that which was the responsibility of the provider, and that which was the responsibility of the user. Cloud computing extends this boundary to cover servers as well as the network infrastructure. The first scholarly use of the term “cloud computing” was in a 1997 lecture by Ramnath Chellappa.

After the dot-com bubble, Amazon played a key role in the development of cloud computing by modernizing their data centers, which, like most computer networks, were using as little as 10% of their capacity at any one time, just to leave room for occasional spikes. Having found that the new cloud architecture resulted in significant internal efficiency improvements whereby small, fast-moving "two-pizza teams" could add new features faster and more easily, Amazon initiated a new product development effort to provide cloud computing to external customers, and launched Amazon Web Service (AWS) on a utility computing basis in 2006.

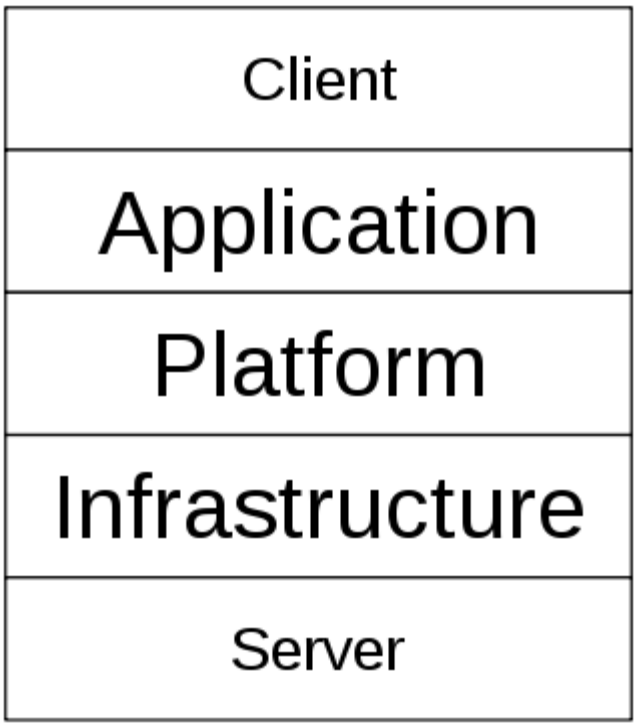
In 2007, Google, IBM and a number of universities embarked on a large-scale cloud computing research project. In early 2008, Eucalyptus became the first open-source, AWS API-compatible platform for deploying private clouds. In early 2008, OpenNebula, enhanced in the RESERVOIR European Commission-funded project, became the first open-source software for deploying private and hybrid clouds, and for the federation of clouds. In the same year, efforts were focused on providing QoS guarantees (as required by real-time interactive applications) to cloud-based infrastructures, in the framework of the IRMOS European Commission-funded project. By mid-2008, Gartner saw an opportunity for cloud computing "to shape the relationship among consumers of IT services, those who use IT services and those who sell them" and observed that "[o]rganisations are switching from company-owned hardware and software assets to per-use service-based models" so that the "projected shift to cloud computing ... will result in dramatic growth in IT products in some areas and significant reductions in other areas."

Key characteristics

- **Agility** improves with users' ability to rapidly and inexpensively re-provision technological infrastructure resources.
- **Application Programming Interface** (API) accessibility to software that enables machines to interact with cloud software in the same way the user interface facilitates interaction between humans and computers. Cloud computing systems typically use REST-based APIs.
- **Cost** is claimed to be greatly reduced and in a public cloud delivery model capital expenditure is converted to operational expenditure. This ostensibly lowers barriers to entry, as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained with usage-based options and fewer IT skills are required for implementation (in-house).
- **Device and location independence** enable users to access systems using a web browser regardless of their location or what device they are using (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.
- **Multi-tenancy** enables sharing of resources and costs across a large pool of users thus allowing for:
    - **Centralization** of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
    - **Peak-load capacity** increases (users need not engineer for highest possible load-levels)
    - **Utilization and efficiency** improvements for systems that are often only 10–20% utilized.
- **Reliability** is improved if multiple redundant sites are used, which makes well designed cloud computing suitable for business continuity and disaster recovery.
- **Scalability** via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis near real-time, without users having to engineer for peak loads. Performance is monitored, and consistent and loosely coupled architectures are constructed using web services as the system interface.
- **Security** could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than under traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems which are being shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.
- **Maintenance** of cloud computing applications is easier, because they do not need to be installed on each user's computer. They are easier to support and to improve, as the changes reach the clients instantly.

Layers

Once an Internet Protocol connection is established among several computers, it is possible to share services within any one of the following layers.



Client

A *cloud client* consists of computer hardware and/or computer software that relies on cloud computing for application delivery, or that is specifically designed for delivery of cloud services and that, in either case, is essentially useless without it. Examples include some computers, phones and other devices, operating systems and browsers.

Application

Cloud application services or "Software as a Service (SaaS)" deliver software as a service over the Internet, eliminating the need to install and run the application on the customer's own computers and simplifying maintenance and support. People tend to use the terms "SaaS" and "cloud" interchangeably, when in fact they are two different things. Key characteristics include:

- Network-based access to, and management of, commercially available (i.e., not custom) software
- Activities that are managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the Web

- Application delivery that typically is closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics
- Centralized feature updating, which obviates the need for downloadable patches and upgrades

Platform

Cloud platform services or "Platform as a Service (PaaS)" deliver a computing platform and/or solution stack as a service, often consuming cloud infrastructure and sustaining cloud applications. It facilitates deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers.

Infrastructure

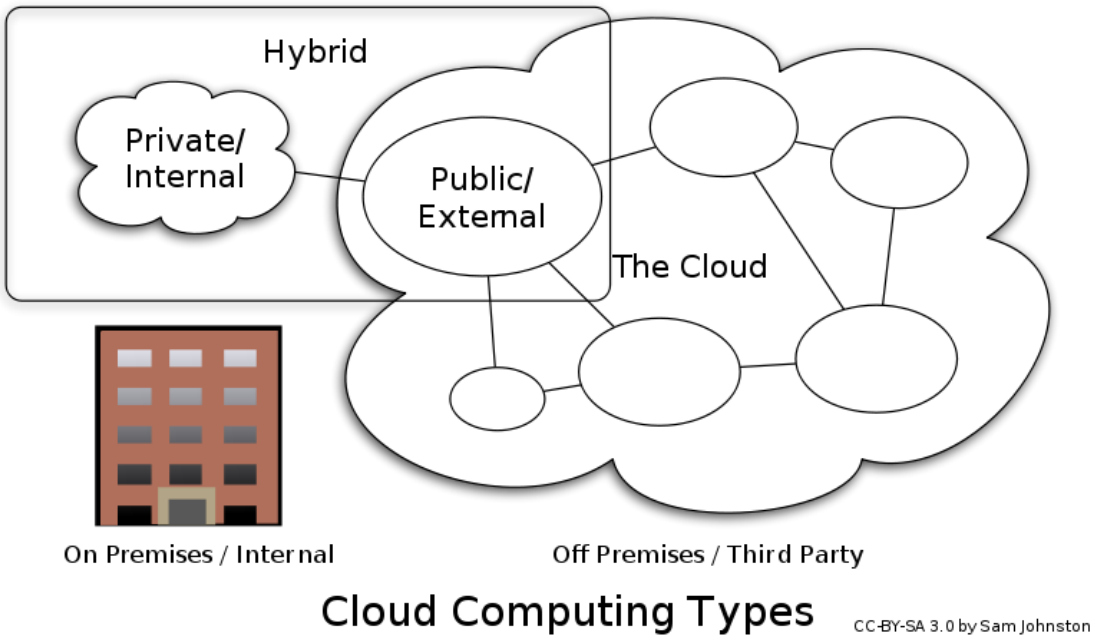
Cloud infrastructure services, also known as "Infrastructure as a Service (IaaS)", delivers computer infrastructure – typically a platform virtualization environment – as a service. Rather than purchasing servers, software, data-center space or network equipment, clients instead buy those resources as a fully outsourced service. Suppliers typically bill such services on a utility computing basis and amount of resources consumed (and therefore the cost) will typically reflect the level of activity. IaaS evolved from virtual private server offerings.

Cloud infrastructure often takes the form of a tier 3 data center with many tier 4 attributes, assembled from hundreds of virtual machines.

Server

The servers layer consists of computer hardware and/or computer software products that are specifically designed for the delivery of cloud services, including multi-core processors, cloud-specific operating systems and combined offerings.

Deployment models



Cloud computing types

Public cloud

Public cloud or external cloud describes cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web applications/web services, from an off-site third-party provider who bills on a fine-grained utility computing basis.

Community cloud

A community cloud may be established where several organizations have similar requirements and seek to share infrastructure so as to realize some of the benefits of cloud computing. With the costs spread over fewer users than a public cloud (but more than a single tenant) this option is more expensive but may offer a higher level of privacy, security and/or policy compliance. Examples of community clouds include Google's "Gov Cloud".

Hybrid cloud and hybrid IT delivery

The main responsibility of the IT department is to deliver services to the business. With the proliferation of cloud computing (both private and public) and the fact that IT departments must also deliver services via traditional, in-house methods, the newest catch-phrase has become “hybrid cloud computing.” Hybrid cloud is also called hybrid delivery by the major vendors including HP, IBM, Oracle and VMware who offer

technology to manage the complexity in managing the performance, security and privacy concerns that results from the mixed delivery methods of IT services.

A hybrid storage cloud uses a combination of public and private storage clouds. Hybrid storage clouds are often useful for archiving and backup functions, allowing local data to be replicated to a public cloud.

Another perspective on deploying a web application in the cloud is using Hybrid Web Hosting, where the hosting infrastructure is a mix between cloud hosting and managed dedicated servers – this is most commonly achieved as part of a web cluster in which some of the nodes are running on real physical hardware and some are running on cloud server instances.

Combined cloud

Two clouds that have been joined together are more correctly called a "combined cloud". A combined cloud environment consisting of multiple internal and/or external providers "will be typical for most enterprises". By integrating multiple cloud services users may be able to ease the transition to *public cloud* services while avoiding issues such as PCI compliance.

Private cloud

Douglas Parkhill first described the concept of a "private computer utility" in his 1966 book *The Challenge of the Computer Utility*. The idea was based upon direct comparison with other industries (e.g. the electricity industry) and the extensive use of hybrid supply models to balance and mitigate risks.

"Private cloud" and "internal cloud" have been described as neologisms, but the concepts themselves pre-date the term cloud by 40 years. Even within modern utility industries, hybrid models still exist despite the formation of reasonably well-functioning markets and the ability to combine multiple providers.

Some vendors have used the terms to describe offerings that emulate cloud computing on private networks. These (typically virtualization automation) products offer the ability to host applications or virtual machines in a company's own set of hosts. These provide the benefits of utility computing – shared hardware costs, the ability to recover from failure, and the ability to scale up or down depending upon demand.

Private clouds have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from lower up-front capital costs and less hands-on management, essentially "[lacking] the economic model that makes cloud computing such an intriguing concept". Enterprise IT organizations use their own private cloud(s) for mission critical and other operational systems to protect critical infrastructures.

Cloud engineering

Cloud engineering is the application of a systematic, disciplined, quantifiable, and interdisciplinary approach to the ideation, conceptualization, development, operation, and maintenance of cloud computing, as well as the study and applied research of the approach, i.e., the application of engineering to cloud. It is a maturing and evolving discipline to facilitate the adoption, strategization, operationalization, industrialization, standardization, productization, commoditization, and governance of cloud solutions, leading towards a cloud ecosystem. Cloud engineering is also known as cloud service engineering.

Cloud storage

Cloud storage is a model of networked computer data storage where data is stored on multiple virtual servers, generally hosted by third parties, rather than being hosted on dedicated servers. Hosting companies operate large data centers; and people who require their data to be hosted buy or lease storage capacity from them and use it for their storage needs. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as virtual servers, which the customers can themselves manage. Physically, the resource may span across multiple servers.

The Intercloud

The Intercloud is an interconnected global "cloud of clouds" and an extension of the Internet "network of networks" on which it is based. The term was first used in the context of cloud computing in 2007 when Kevin Kelly stated that "eventually we'll have the intercloud, the cloud of clouds. This Intercloud will have the dimensions of one machine comprising all servers and attendant cloudbooks on the planet.". It became popular in 2009 and has also been used to describe the datacenter of the future.

The Intercloud scenario is based on the key concept that each single cloud does not have infinite physical resources. If a cloud saturates the computational and storage resources of its virtualization infrastructure, it could not be able to satisfy further requests for service allocations sent from its clients. The Intercloud scenario aims to address such situation, and in theory, each cloud can use the computational and storage resources of the virtualization infrastructures of other clouds. Such form of pay-for-use may introduce new business opportunities among cloud providers if they manage to go beyond theoretical framework. Nevertheless, the Intercloud raises many more challenges than solutions concerning cloud federation, security, interoperability, quality of service, vendor's lock-ins, trust, legal issues, monitoring and billing.

The concept of a competitive utility computing market which combined many computer utilities together was originally described by Douglas Parkhill in his 1966 book, the "Challenge of the Computer Utility". This concept has been subsequently used many times over the last 40 years and is identical to the Intercloud.

Issues

Privacy

The cloud model has been criticized by privacy advocates for the greater ease in which the companies hosting the cloud services control, and thus, can monitor at will, lawfully or unlawfully, the communication and data stored between the user and the host company. Instances such as the secret NSA program, working with AT&T, and Verizon, which recorded over 10 million phone calls between American citizens, causes uncertainty among privacy advocates, and the greater powers it gives to telecommunication companies to monitor user activity. While there have been efforts (such as US-EU Safe Harbor) to "harmonize" the legal environment, providers such as Amazon still cater to major markets (typically the United States and the European Union) by deploying local infrastructure and allowing customers to select "availability zones."

Compliance

In order to obtain compliance with regulations including FISMA, HIPAA and SOX in the United States, the Data Protection Directive in the EU and the credit card industry's PCI DSS, users may have to adopt *community* or *hybrid* deployment modes which are typically more expensive and may offer restricted benefits. This is how Google is able to "manage and meet additional government policy requirements beyond FISMA" and Rackspace Cloud are able to claim PCI compliance. Customers in the EU contracting with cloud providers established outside the EU/EEA have to adhere to the EU regulations on export of personal data.

Many providers also obtain SAS 70 Type II certification (e.g. Amazon, Salesforce.com, Google and Microsoft), but this has been criticised on the grounds that the hand-picked set of goals and standards determined by the auditor and the auditee are often not disclosed and can vary widely. Providers typically make this information available on request, under non-disclosure agreement.

Legal

In March 2007, Dell applied to trademark the term "cloud computing" (U.S. Trademark 77,139,082) in the United States. The "Notice of Allowance" the company received in July 2008 was canceled in August, resulting in a formal rejection of the trademark application less than a week later. Since 2007, the number of trademark filings covering cloud computing brands, goods and services has increased at an almost exponential rate. As companies sought to better position themselves for cloud computing branding and marketing efforts, cloud computing trademark filings increased by 483% between 2008 and 2009. In 2009, 116 cloud computing trademarks were filed, and trademark analysts predict that over 500 such marks could be filed during 2010.

Other legal cases may shape the use of cloud computing by the public sector. On October 29, 2010, Google filed a lawsuit against the U.S. Department of Interior, which opened

up a bid for software that required that bidders use Microsoft's Business Productivity Online Suite. Google sued, calling the requirement "unduly restrictive of competition." Scholars have pointed out that, beginning in 2005, the prevalence of open standards and open source may have an impact on the way that public entities choose to select vendors.

Open source

Open source software has provided the foundation for many cloud computing implementations. In November 2007, the Free Software Foundation released the Affero General Public License, a version of GPLv3 intended to close a perceived legal loophole associated with free software designed to be run over a network.

Open standards

Most cloud providers expose APIs which are typically well-documented (often under a Creative Commons license) but also unique to their implementation and thus not interoperable. Some vendors have adopted others' APIs and there are a number of open standards under development, including the OGF's Open Cloud Computing Interface. The Open Cloud Consortium (OCC) is working to develop consensus on early cloud computing standards and practices.

Security

The relative security of cloud computing services is a contentious issue which may be delaying its adoption. Issues barring the adoption of cloud computing are due in large part to the private and public sectors unease surrounding the external management of security based services. It is the very nature of cloud computing based services, private or public, that promote external management of provided services. This delivers great incentive amongst cloud computing service providers in producing a priority in building and maintaining strong management of secure services.

Organizations have been formed in order to provide standards for a better future in cloud computing services. One organization in particular, the Cloud Security Alliance is a non-profit organization formed to promote the use of best practices for providing security assurance within cloud computing.

Availability and performance

In addition to concerns about security, businesses are also worried about acceptable levels of availability and performance of applications hosted in the cloud.

There are also concerns about a cloud provider shutting down for financial or legal reasons, which has happened in a number of cases.

Sustainability and siting

Although cloud computing is often assumed to be a form of "green computing", there is as of yet no published study to substantiate this assumption. Siting the servers affects the environmental effects of cloud computing. In areas where climate favors natural cooling and renewable electricity is readily available, the environmental effects will be more moderate. Thus countries with favorable conditions, such as Finland, Sweden and Switzerland, are trying to attract cloud computing data centers.

SmartBay, marine research infrastructure of sensors and computational technology, is being developed using cloud computing, an emerging approach to shared infrastructure in which large pools of systems are linked together to provide IT services.

Research

A number of universities, vendors and government organizations are investing in research around the topic of cloud computing. Academic institutions include University of Melbourne (Australia), Georgia Tech, Yale, Wayne State, Virginia Tech, University of Wisconsin–Madison, Carnegie Mellon, MIT, Indiana University, University of Massachusetts, University of Maryland, IIT Bombay, North Carolina State University, Purdue University, University of California, University of Washington, University of Virginia, University of Utah, University of Minnesota, among others.

Joint government, academic and vendor collaborative research projects include the IBM/Google Academic Cloud Computing Initiative (ACCI). In October 2007 IBM and Google announced the multi- university project designed to enhance students' technical knowledge to address the challenges of cloud computing. In April 2009, the National Science Foundation joined the ACCI and awarded approximately $5 million in grants to 14 academic institutions.

In July 2008, HP, Intel Corporation and Yahoo! announced the creation of a global, multi-data center, open source test bed, called Open Cirrus, designed to encourage research into all aspects of cloud computing, service and data center management. Open Cirrus partners include the NSF, the University of Illinois (UIUC), Karlsruhe Institute of Technology, the Infocomm Development Authority (IDA) of Singapore, the Electronics and Telecommunications Research Institute (ETRI) in Korea, the Malaysian Institute for Microelectronic Systems(MIMOS), and the Institute for System Programming at the Russian Academy of Sciences (ISPRAS). In Sept. 2010, more researchers joined the HP/Intel/Yahoo Open Cirrus project for cloud computing research. The new researchers are China Mobile Research Institute (CMRI), Spain's Supercomputing Center of Galicia (CESGA by its Spanish acronym), Georgia Tech's Center for Experimental Research in Computer Systems (CERCS) and China Telecom.

In July 2010, HP Labs India announced a new cloud-based technology designed to simplify taking content and making it mobile-enabled, even from low-end devices. Called SiteonMobile, the new technology is designed for emerging markets where people are

more likely to access the internet via mobile phones rather than computers. In Nov. 2010, HP formally opened its Government Cloud Theatre, located at the HP Labs site in Bristol, England. The demonstration facility highlights high-security, highly flexible cloud computing based on intellectual property developed at HP Labs. The aim of the facility is to lessen fears about the security of the cloud. HP Labs Bristol is HP's second-largest central research location and currently is responsible for researching cloud computing and security.

The IEEE Technical Committee on Services Computing in IEEE Computer Society sponsors the IEEE International Conference on Cloud Computing (CLOUD). CLOUD 2010 was held on July 5–10, 2010 in Miami, Florida

On March 23, 2011, Google, Microsoft, HP, Yahoo, Verizon, Deutsche Telecom and 17 other companies formed a nonprofit organization called Open Networking Foundation, focused on providing support for a new cloud initiative called Software-Defined Networking. The initiative is meant to speed innovation through simple software changes in telecommunications networks, wireless networks, data centers and other networking areas.

Criticism of the term

Some have come to criticize the term as being either too unspecific or even misleading. CEO Larry Ellison of Oracle Corporation asserts that cloud computing is "everything that we already do", claiming that the company could simply "change the wording on some of our ads" to deploy their cloud-based services. Forrester Research VP Frank Gillett questions the very nature of and motivation behind the push for cloud computing, describing what he calls "cloud washing" in the industry whereby companies relabel their products as cloud computing resulting in a lot of marketing innovation on top of real innovation. GNU's Richard Stallman insists that the industry will only use the model to deliver services at ever increasing rates over proprietary systems, otherwise likening it to a "marketing hype campaign".

Chapter 2

Cloud Storage and Cloud Storage Gateway

Cloud storage

Cloud storage is a model of networked online storage where data is stored on multiple virtual servers, generally hosted by third parties, rather than being hosted on dedicated servers. Hosting companies operate large data centers; and people who require their data to be hosted buy or lease storage capacity from them and use it for their storage needs. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers.

Cloud storage services may be accessed through a web service application programming interface (API), or through a Web-based user interface.

Cloud storage advantages

- Companies need only pay for the storage they actually use.
- Companies do not need to install physical storage devices in their own datacenter or offices, which reduces IT and hosting costs.
- Storage maintenance tasks, such as backup, data replication, and purchasing additional storage devices are offloaded to the responsibility of a service provider, allowing organizations to focus on their core business

Potential concerns

- Security of stored data and data in transit may be a concern when storing sensitive data at a cloud storage provider
- Performance may be lower than local storage
- Reliability and availability depends on wide area network availability and on the level of precautions taken by the service provider.

- Users with specific records-keeping requirements, such as public agencies that must retain electronic records according to statute, may encounter complications with using cloud computing and storage.

Cloud storage gateways

A cloud storage gateway can be optionally used at the customer premises, which expose cloud storage services as if they were local storage devices. Cloud storage gateways are network appliances or servers which translate standard cloud storage APIs such as SOAP or REST to either block-based data storage protocols such as iSCSI or Fibre Channel, or file-based network storage protocols such as NFS or CIFS (SMB).

Technology adoption

Public cloud storage services are fast becoming a more attractive option for the enterprise, according to a 2010 research. In 2010, service providers including Nirvanix, Nasuni, CTERA Networks and Petaera begun to use new generation object-oriented storage technology, which allows storing very large volumes of data at far lower cost than conventional enterprise storage systems. These companies offer enterprise customers a Cloud storage gateway that translates traditional file based protocols to cached object-oriented storage.

In this context, the EU-funded VISION Cloud Project VISION Cloud EU Project is developing a cloud-based infrastructure, built on open standards and new technologies, to provide a scalable, flexible and dependable framework for optimized delivery of data-intensive storage services.

According to the IDC storage analyst Brad Nisbet, solutions that focus on delivering "cost-effective and user-friendly" managed services are likely to encourage adoption of cloud storage within the small business market space.

Cloud storage gateway

A **cloud storage gateway** (or **Hybrid Cloud Gateway**) is a network appliance or server which resides at the customer premises and translates cloud storage APIs such as SOAP or REST to block-based storage protocols such as iSCSI or Fibre Channel or file-based interfaces such as NFS or CIFS.

Unlike the cloud storage services which they complement, cloud storage gateways use standard network protocols which provide a seamless integration with existing applications. Cloud storage gateways can also serve as intermediaries to multiple cloud storage providers. Some cloud storage gateways also include additional storage features

such as backup and recovery, encryption, deduplication and provisioning. According to analyst Terri McClure, cloud storage gateways help to address security concerns with deploying cloud storage by providing an onsite appliance that encrypts data while keeping the keys local, reduces latency by caching data or keeping local copies, and provides snapshot capabilities for data availability.

Cloud storage gateways are good targets for performing backups. LAN speed performance helps backups to complete within their allocated time windows, while the backup is later replicated asynchronously to offsite storage, after data deduplication and compression are performed.

Cloud gateways typically use local caching to alleviate latency issues.

Since some cloud providers charge for bandwidth and storage space, cloud gateways typically offer transparent data compression.

Vendors and product lines in the cloud gateway market include Gladinet, CTERA Networks Cloud Attached Storage, Nasuni Cloud Storage Gateway, TwinStrata, StorSimple, Cirtas, Riverbed Technology, Panzura , and Emulex.

According to a 2011 report by Gartner Group, Cloud gateways are expected to increase the use of cloud storage by lowering monthly charges and eliminating the concern of data security.

Chapter 3

Diskless Node

A **diskless node** (or **diskless workstation**) is a workstation or personal computer without disk drives, which employs network booting to load its operating system from a server. (A computer may also be said to *act as a diskless node*, if its disks are unused and network booting is used.)

Diskless nodes (or computers acting as such) are sometimes known as *network computers* or **hybrid clients**. *Hybrid client* may either just mean diskless node, or it may be used in a more particular sense to mean a diskless node which runs *some*, but not all, applications remotely, as in the thin client computing architecture.

Advantages of diskless nodes can include lower production cost, lower running costs, quieter operation, and manageability advantages (for example, centrally-managed software installation).

In many universities and in some large organizations, PCs are used in a similar configuration, with some or all applications stored remotely but executed locally—again, for manageability reasons. However, these are not diskless nodes if they still boot from a local hard drive.

Distinction between diskless nodes and centralized computing

Diskless nodes process data, thus using their own CPU and RAM to run software, but do not store data persistently—that task is handed off to a server. This is distinct from thin clients, in which all significant processing happens remotely, on the server—the only software that runs on a thin client is the "thin" (i.e. relatively small and simple) client software, which handles simple input/output tasks to communicate with the user, such as drawing a dialog box on the display or waiting for user input.

A collective term encompassing both thin client computing, and its technological predecessor, text terminals (which are text-only), is centralized computing. Thin clients and text terminals can both require powerful central processing facilities in the servers, in order to perform all significant processing tasks for all of the clients.

Diskless nodes can be seen as a compromise between fat clients (such as ordinary personal computers) and centralized computing, using central storage for efficiency, but not requiring centralized processing, and making efficient use of the powerful processing power of even the slowest of contemporary CPUs, which would tend to sit idle for much of the time under the centralized computing model.

	Centralized computing or Thin client	Diskless node	Fat client
Local hard drives used	No	No	Yes
Local general-purpose processing used	No	Yes	Yes

Principles of operation

The operating system (OS) for a diskless node is loaded from a server, using network booting. In some cases, removable storage may be used to initiate the bootstrap process, such as a USB flash drive, or other bootable media such as a floppy disk, CD or DVD. However, the firmware in many modern computers can be configured to locate a server and begin the bootup process automatically, without the need to insert bootable media.

After the bootstrapping process has been initiated, as described above, bootstrapping will take place according to one of three main approaches.

- In the first approach (used, for example, by the Linux Terminal Server Project), the kernel is loaded into memory and then the rest of the operating system is accessed via a network filesystem connection to the server. (A small RAM disk may be created to store temporary files locally.) This approach is sometimes called the "NFS root" technique when used with Linux or Unix client operating systems.

- In the second approach, the kernel of the OS is loaded, and part of the system's memory is configured as a large RAM disk, and then the remainder of the OS image is fetched and loaded into the RAM disk. This is the implementation that Microsoft has chosen for its Windows XP embedded remote boot feature.

- In the third approach, disk operations are virtualized and are actually translated into a network protocol. The data that are usually stored in a disk drive are then stored in virtual disks files homed on a server. The disk operations such as requests to read/write disk sectors are translated into corresponding network requests and processed by a service or daemon running on the server side. This is the implementation that is used by Neoware Image Manager, Ardence, VHD and various "boot over iSCSI" products. This third approach differs from the first approach because what is remote is not a file system but actually a disk device (or raw device) and that the client OS is not aware that it is not running off a hard

disk. This is why this approach is sometimes named "Virtual Hard Disk" or "Network Virtual Disk".

This third approach makes it easier to use client OS than having a complete disk image in RAM or using a read-only file system. And it is compatible with most of modern OSes (Linux and Windows, knowing that Windows cannot live on read-only media...). In this approach, the system uses some "write cache" that stores every data that a diskless node has written. This write cache is usually a file, stored on a server (or on the client storage if any). It can also be a portion of the client RAM. This write cache can be persistent or volatile. When volatile, all the data that has been written by a specific client to the virtual disk are dismissed when said client is rebooted, and yet, user data can remain persistent if recorded in user (roaming) profiles or home folders (that are stored on remote servers). The two major commercial products (the one from Hewlett-Packard, and the other one from Citrix Systems) that allow the deployment of Diskless Nodes that can boot Microsoft Windows or Linux client OS use such write caches. The Citrix product cannot use persistent write cache, but VHD and HP product can.

Diskless Windows nodes

Windows 3.x and Windows 95 OSR1 supported Remote Boot operations, from Netware servers, Windows NT Servers and even DEC Pathworks servers.

Third party software Vendors such as Qualystem (acquired by Neoware), LanWorks (acquired by 3Com), Ardence (acquired by Citrix),APCT and Xtreamining Technology have developed and marketed software products aimed to remote-boot newer versions of the Windows product line: Windows 95 OSR2 and Windows 98 were supported by Qualystem and Lanworks, Windows NT was supported by APCT and Ardence (called VenturCom at that time), and Windows 2000/XP/2003/Vista/Windows 7 are supported by Hewlett Packard (which acquired Neoware which had previously acquired Qualystem) and Citrix Systems (which acquired Ardence).

Comparison with fat clients

Software installation and maintenance

With essentially a single OS image for an array of machines (with perhaps some customizations for differences in hardware configurations among the nodes), installing software and maintaining installed software can be more efficient. Furthermore, any system changes made during operation (due to user action, worms, viruses, etc.) can be either wiped out when the power is removed (if the image is copied to a local RAM disk) such as Windows XP Embedded remote boot or prohibited entirely (if the image is a network filesystem). This allows use in public access areas (such as libraries) or in schools etc. where users might wish to experiment or attempt to "hack" the system.

However, it is not necessary to implement network booting to achieve either of the above advantages - ordinary PCs (with the help of appropriate software) can be configured to

download and reinstall their operating systems on (e.g.) a nightly basis, with extra work compared to using shared disk image that diskless nodes boot off.

Modern diskless nodes can share the very same disk image, using a 1:N relationship (1 disk image used simultaneously by N diskless nodes). This makes it very easy to install and maintain software applications: The administrator needs to install or maintain the application only once, and the clients can get the new application as soon as they boot off the updated image. Disk image sharing is made possible because they use the write cache: No client competes for any writing in a shared disk image, because each client writes to its own cache.

All the modern diskless nodes systems can also use a 1:1 Client-to-DiskImage relationship, where one client "owns" one disk image and writes directly into said disk image. No write cache is used then.

Making a modification in a shared disk image is usually made this way:

1. The administrator makes a copy of the shared disk image that he/she wants to update (this can be done easily because the disk image file is opened only for reading)

2. The administrator boots a diskless node in 1:1 mode (unshared mode) from the copy of the disk image he/she just made

3. The administrator makes any modification to the disk image (for instance install a new software application, apply patches or hotfixes...)

4. The administrator shutdowns the diskless node that was using the disk image in 1:1 mode

5. The administrator shares the modified disk image

6. The diskless nodes use the shared disk image (1:N) as soon as they are rebooted.

Centralized storage

The use of central disk storage also makes more efficient use of disk storage. This can cut storage costs, freeing up capital to invest in more reliable, modern storage technologies, such as RAID arrays which support redundant operation, and storage area networks which allow hot-adding of storage without any interruption. Further, it means that losses of disk drives to mechanical or electrical failure—which are statistically highly probable events over a timeframe of years, with a large number of disks involved—are often both less likely to happen (because there are typically less disk drives that can fail) and less likely to cause interruption (because they would likely be part of RAID arrays). This also means that the nodes *themselves* are less likely to have hardware failures than fat clients.

Diskless nodes share these advantages with thin clients.

Performance of centralized storage

However, this storage efficiency can come at a price. As often happens in computing, increased storage efficiency sometimes comes at the price of decreased performance.

Large numbers of nodes making demands on the same server simultaneously can slow down everyone's experience. However, this can be mitigated by installing large amounts of RAM on the server (which speeds up read operations by improving caching performance), by adding more servers (which distributes the I/O workload), or by adding more disks to a RAID array (which distributes the *physical* I/O workload). In any case this is also a problem which can affect *any* client-server network to some extent, since, of course, fat clients also use servers to store user data.

Indeed, user data may be much more significant in size and may be accessed far more frequently than operating systems and programs in some environments, so moving to a diskless model will not *necessarily* cause a noticeable degradation in performance.

Greater network bandwidth (i.e. capacity) will also be used in a diskless model, compared to a fat client model. This does not necessarily mean that a higher capacity network infrastructure will need to be installed—it could simply mean that a higher proportion of the existing network capacity will be used.

Finally, the combination of network data transfer latencies (physically transferring the data over the network) and contention latencies (waiting for the server to process other node's requests before yours) can lead to an unacceptable degradation in performance compared to using local drives, depending on the nature of the application and the capacity of the network infrastructure and the server.

Other advantages

Another example of a situation where a diskless node would be useful is in a possibly hazardous environment where computers are likely to be damaged or destroyed, thus making the need for inexpensive nodes, and minimal hardware a benefit. Again, thin clients can also be used here.

Diskless machines may also consume little power and make little noise, which implies potential environmental benefits and makes them ideal for some computer cluster applications.

Comparison with thin clients

Major corporations tend to instead implement thin clients (using Microsoft Windows Terminal Server or other such software), since much lower specification hardware can be used for the client (which essentially acts as a simple "window" into the central server which is actually running the users operating system as a login session). Of course, diskless nodes can also be used as thin clients. Moreover, thin client computers are

increasing in power to the point where they are becoming suitable as fully-fledged diskless workstations for some applications.

Both thin client and diskless node architectures employ diskless clients which have advantages over fat clients (see above), but differ with regard to the location of processing.

Advantages of diskless nodes over thin clients

- **Distributed load** The *processing* load of diskless nodes is distributed. Each user gets its own processing isolated environment, barely affecting other users in the network, as long as their workload is not filesystem-intensive. Thin clients rely on the central server for the processing and thus require a fast server. When the central server is busy and slow, both kinds of clients will be affected, but thin clients will be slowed down completely, whereas diskless nodes will only be slowed down when accessing data on the server.
- **Better multimedia performance**. Diskless nodes have advantages over thin clients in multimedia-rich applications that would be bandwidth intensive if fully served. For example, diskless nodes are well suited for video gaming.
- **Peripheral support** Diskless nodes are typically ordinary personal computers or workstations with no hard drives supplied, which means the usual large variety of peripherals can be added. By contrast, thin clients are typically very small, sealed boxes with no possibility for internal expansion, and limited or non-existent possibility for external expansion. Even if e.g. a USB device can be *physically* attached to a thin client, the thin client software might not support peripherals beyond the basic input and output devices - for example, it may not be compatible with graphics tablets, digital cameras or scanners.

Advantages of thin clients over diskless nodes

- The **hardware is cheaper** on thin clients, since processing requirements on the client are minimal, and 3D acceleration and elaborate audio support are not usually provided. Of course, a diskless node can also be purchased with a cheap CPU and minimal multimedia support, if suitable. Thus, cost savings may be smaller than they first appear for some organizations. However, many large organizations habitually buy hardware with a higher than necessary specification to meet the needs of particular applications and uses, or to ensure future proofing *(see below)*. There are also less "rational" reasons for overspecifying hardware which quite often come into play: departments wastefully using up budgets in order to retain their current budget levels for next year; and uncertainty about the future, or lack of technical knowledge, or lack of care and attention, when choosing PC specifications. Taking all these factors into account, thin clients may bring the most substantial savings, as only the servers are likely to be substantially "gold-plated" and/or "future-proofed" in the thin client model.

- **Future proofing** is not much of an issue for thin clients, which are likely to remain useful for the entirety of their replacement cycle - one to four years, or even longer - as the burden is on the servers. There are issues when it comes to diskless nodes, as the processing load is potentially much higher, thus meaning more consideration is required when purchasing. Thin client networks may require significantly more powerful servers in the future, whereas a diskless nodes network may in future need a server upgrade, a client upgrade, or both.

- Thin client networks have **less network bandwidth consumption** potentially, since much data is simply read by the server and processed there, and only transferred to the client in small pieces, as and when needed for display. Also, transferring graphical data to the display is usually more suited for efficient data compression and optimisation technologies than transferring arbitrary programs, or user data. In many typical application scenarios, both total bandwidth consumption and "burst" consumption would be expected to be less for an efficient thin client, than for a diskless node.

Chapter 4

Centralized Computing and Cloud Backup

Centralized computing

Centralized computing is computing done at a central location, using terminals that are attached to a central computer. The computer itself may control all the peripherals directly (if they are physically connected to the central computer), or they may be attached via a terminal server. Alternatively, if the terminals have the capability, they may be able to connect to the central computer over the network. The terminals may be text terminals or thin clients, for example.

It offers greater security over decentralized systems because all of the processing is controlled in a central location. In addition, if one terminal breaks down, the user can simply go to another terminal and log in again, and all of their files will still be accessible. Depending on the system, they may even be able to resume their session from the point they were at before, as if nothing had happened.

This type of arrangement does have some disadvantages. The central computer performs the computing functions and controls the remote terminals. This type of system relies totally on the central computer. Should the central computer crash, the entire system will "go down" (i.e. will be unavailable).

Another disadvantage is that central computing relies heavily on the quality of administration and resources provided to its users. Should the central computer be inadequately supported by any means (e.g. size of home directories, problems regarding administration), then your usage will suffer greatly. The reverse situation, however, (i.e., a system supported better than your needs) is one of the key advantages to centralized computing.

History

The very first computers did not have separate terminals as such; their primitive input/output devices were built in. However, soon it was found to be extremely useful for multiple people to be able to use a computer at the same time, for reasons of cost - early

computers were very expensive, both to produce and maintain, and occupied large amounts of floor space. The idea of centralized computing was born. Early text terminals used electro-mechanical teletypewriters, but these were replaced by cathode ray tube displays (as found in 20th century televisions and computers). The text terminal model dominated computing from the 1960s until the rise to dominance of home computers and personal computers in the 1980s.

Contemporary status

As of 2007, centralized computing is now coming back into fashion - to a certain extent. Thin clients have been used for many years by businesses to reduce total cost of ownership, while web applications are becoming more popular because they can potentially be used on many types of computing device without any need for software installation. Already, however, there are signs that the pendulum is swinging back again, away from pure centralization, as thin client devices become more like diskless workstations due to increased computing power, and web applications start to do more processing on the client side, with technologies such as AJAX and rich clients.

In addition, mainframes are still being used for some mission-critical applications, such as payroll, or for processing day-to-day account transactions in banks. These mainframes will typically be accessed either using terminal emulators (real terminal devices are not used much any more) or via modern front-ends such as web applications - or (in the case of automated access) protocols such as web services protocols.

Hybrid client model

Some organisations use a hybrid client model partway between centralized computing and conventional desktop computing, in which some applications (such as web browsers) are run locally, while other applications (such as critical business systems) are run on the terminal server. One way to implement this is simply by running remote desktop software on a standard desktop computer.

Hosted computing model

A relatively new method of centralized computing, **hosted computing**, solves many of the problems associated with traditional distributed computing systems. By centralizing processing and storage on powerful server hardware located in a data center, rather than in a local office, it relieves organizations of the many responsibilities in owning and maintaining an information technology system. These services are typically delivered on a subscription basis by an application service provider (ASP).

Cloud backup

Cloud Backup and Recovery (Cloud BUR) is a subset of Cloud Computing. This definition embraces the definitions of Cloud Computing as set forth by NIST and Gartner and takes it a step further to define Cloud BUR.

Cloud Backup and Recovery has the following characteristics:

Service-based

1. The assurance, guarantee, or validation that what was backed up is recoverable whenever it is required is critical. Data stored in the service provider's cloud must undergo regular integrity validation to ensure its recoverability.
2. Cloud BUR services need to provide a variety of granularity when it comes to RTOs. One size does not fit all either for the customers or the applications within a customer's environment.
3. The customer should never have to manage the back end storage repositories in order to back up and recover data.
4. The interface used by the customer needs to enable the selection of data to protect or recover, the establishment of retention times, destruction dates as well as scheduling.
5. Cloud backup needs to be an active process where data is collected from systems that store the original copy. This means that cloud backup will not require data to be copied into a specific appliance from where data is collected before being transmitted to and stored in the service provider's data centre.

Ubiquitous Access

1. Cloud BUR utilizes standard networking protocols (which today are primarily but not exclusively IP based) to transfer data between the customer and the service provider.
2. Vaults or repositories need to be always available to restore data to any location connected to the Service Provider’s Cloud via private or public networks.

Scalable and Elastic

1. Cloud BUR enables flexible allocation of storage capacity to customers without limit. Storage is allocated on demand and also de-allocated as customers delete backup sets as they age.
2. Cloud BUR enables a Service Provider to allocate storage capacity to a customer. If that customer later deletes their data or no longer needs that capacity, the Service Provider can then release and reallocate that same capacity to a different customer in an automated fashion.

Metered by Use

1. Cloud Backup allows customers to align the value of data with the cost of protecting it. It is procured on a per-gigabyte per month basis. Prices tend to vary based on the age of data, type of data (email, databases, files etc.), volume, number of backup copies and RTOs.

Shared and Secure

1. The underlying enabling technology for Cloud Backup is a full stack native cloud multitenant platform (shared everything).
2. Data mobility/portability prevents service provider lock-in and allows customers to move their data from one Service Provider to another, or entirely back into a dedicated Private Cloud (or a Hybrid Cloud).
3. Security in the cloud is critical. One customer can never have access to another's data. Additionally, even Service Providers must not be able to access their customer's data without the customer's permission.

Chapter 5

Cloud Computing Security and Elastic Computing

Cloud computing security

Cloud computing security (sometimes referred to simply as "cloud security") is an evolving sub-domain of computer security, network security, and, more broadly, information security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing.

Security issues associated with the cloud

There are a number of security issues/concerns associated with cloud computing but these issues fall into two broad categories: Security issues faced by cloud providers (organizations providing Software-, Platform-, or Infrastructure-as-a-Service via the cloud) and security issues faced by their customers. In most cases, the provider must ensure that their infrastructure is secure and that their clients' data and applications are protected while the customer must ensure that the provider has taken the proper security measures to protect their information.

Dimensions of cloud security

While cloud security concerns can be grouped into any number of dimensions (Gartner names seven while the Cloud Security Alliance identifies fifteen areas of concern) these dimensions have been aggregated into three general areas: Security and Privacy, Compliance, and Legal or Contractual Issues.

Security and privacy

In order to ensure that data is secure (that it cannot be accessed by unauthorized users or simply lost) and that data privacy is maintained, cloud providers attend to the following areas:

Data protection

To be considered protected, data from one customer must be properly segregated from that of another; it must be stored securely when "at rest" and it must be able to move securely from one location to another. Cloud providers have systems in place to prevent data leaks or access by third parties. Proper separation of duties should ensure that auditing and/or monitoring cannot be defeated, even by privileged users at the cloud provider.

Identity management

Every enterprise will have its own identity management system to control access to information and computing resources. Cloud providers either integrate the customer's identity management system into their own infrastructure, using federation or SSO technology, or provide an identity management solution of their own.

Physical and personnel security

Providers ensure that physical machines are adequately secure and that access to these machines as well as all relevant customer data is not only restricted but that access is documented.

Availability

Cloud providers assure customers that they will have regular and predictable access to their data and applications.

Application security

Cloud providers ensure that applications available as a service via the cloud are secure by implementing testing and acceptance procedures for outsourced or packaged application code. It also requires application security measures (application-level firewalls) be in place in the production environment.

Privacy

Finally, providers ensure that all critical data (credit card numbers, for example) are masked and that only authorized users have access to data in its entirety. Moreover, digital identities and credentials must be protected as should any data that the provider collects or produces about customer activity in the cloud.

Compliance

Numerous regulations pertain to the storage and use of data, including Payment Card Industry Data Security Standard (PCI DSS), the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act, among others. Many of these regulations require regular reporting and audit trails. Cloud providers must enable their customers to comply appropriately with these regulations.

Business continuity and data recovery

Cloud providers have business continuity and data recovery plans in place to ensure that service can be maintained in case of a disaster or an emergency and that any data lost will be recovered. These plans are shared with and reviewed by their customers.

Logs and audit trails

In addition to producing logs and audit trails, cloud providers work with their customers to ensure that these logs and audit trails are properly secured, maintained for as long as the customer requires, and are accessible for the purposes of forensic investigation (e.g., eDiscovery).

Unique compliance requirements

In addition to the requirements to which customers are subject, the data centers maintained by cloud providers may also be subject to compliance requirements.

Legal and contractual issues

Aside from the security and compliance issues enumerated above, cloud providers and their customers will negotiate terms around liability (stipulating how incidents involving data loss or compromise will be resolved, for example), intellectual property, and end-of-service (when data and applications are ultimately returned to the customer

Public records

Legal issues may also include records-keeping requirements in the public sector, where many agencies are required by law to retain and make available electronic records in a specific fashion. This may be determined by legislation, or law may require agencies to conform to the rules and practices set by a records-keeping agency. Public agencies using cloud computing and storage must take these concerns into account.

Elastic computing

Elastic computing is about harnessing the power of software and hardware computing resources available to today's information technology organizations. The creation of massive, scalable, internet-sized applications now becomes possible at a total cost of ownership that is dramatically lower than any traditional data center, or hardware-only cloud-based deployment can achieve. Elastic computing also aims to improve the application development lifecycle by proposing a methodic and disciplined approach to complex, heterogeneous system design and deployment. It allows applications to be developed more quickly and readily accommodate future change in policy and architecture. It engages the application developer to focus on building applications rather than spending his time specifying the underlying application platform.

Relationship to utility computing

Elastic computing not only leverages the value of utility computing, but also overcomes its shortfalls and then goes beyond. In essence, elastic computing is a synthesis of evolutionary concepts that have emerged on the enterprise computing scene over decades prior, adding the missing components that are essential to fulfill the promise of a true on-demand application computing stack at pay-per-use prices. The resulting value of elastic computing is enormous. Key benefits include:

- Up and down scalability of both infrastructure software and hardware resources to manage the peaks and troughs that are typical of real-world application usage patterns. In addition, automated system de- ployment, scaling and monitoring of this infrastructure allows application developers and supporting IT personnel to focus on higher value-add areas within the application life-cycle.
- Decoupling the application design specification from the deployment specification to (a) drive best practices in system architecture and design, (b) enable experimentation with system design and testing as a whole, (c) allow the application to leverage software improvements as they emerge, and (d) simplify support for versioning and application migration.
- On-demand utility pricing, which delivers a sensible approach to pricing that gives CPU, storage and bandwidth a normalized unit of measurement that incurs accumulative charges based on actual usage.

The net result is that elastic computing delivers a seamlessly scalable application infrastructure stack that allows organizations of all sizes to quickly and easily plug their applications in to available utility computing environments on a pay-per-use basis.

History

This history of elastic computing starts with the development of CPU and storage as the foundation for all computing technology, through to the rise of utility computing, a trend that has been heralded as a revolutionary change to how IT is planned and implemented.

The driving force behind each new developmental trend is the underlying need for greater computing efficiency achieved through optimized system utilization.

CPU and storage

The hardware layer provides the foundation for all computing technology. Dramatic increases in both CPU power and the capacity of the magnetic drives that feed data to the CPUs have made computers faster, and computing hardware increasingly inexpensive.

The result, however, led to over-planned, over-capacity-enabled data centers that consumed excessive power and cooling resources, creating the demand for greater system utilization and efficiency. This paved the way for virtualization.

Virtualization

Virtualization is broadly defined as the abstraction of computing resources. It involves taking a big server (or servers) or a big disk (or disk farm) and breaking down these large blocks of resources into smaller, independent, logical devices that can be accessed, administered, manipulated and consumed as independent entities. It addresses the under-utilization problem of a large machine that runs one set of applications configured in a specific way by letting fewer of that machine's resources remain idle. Virtualization has given the consumer choice in the way existing hardware is used, and has improved usage of computing resources. However, servers are not always too big. Rather than being split apart, in some instances they can be too small to handle a particularly large computing task. Thus, multiple servers need to be combined to process the large-scale computing needs of applications serving processes such as video transcoding or a Google search. This large-scale computing need led to the development of grid consolidation.

Grid consolidation

Grid consolidation applies the resources of many computers on a network simultaneously to a given computing task. Grids have been made to scale dynamically by allowing for the addition of resources to the running grid in a (sometimes) simple and transparent manner. Until recently, they were usually an "all or none" proposition requiring a large upfront investment in hardware and a re-architecture of existing applications. However, a way emerged of selling the pieces of large, monolithic, expensive grid-based systems in smaller parts by both selling access to multi-tenant applications and chunks of large, consolidated computer systems.

Utility computing

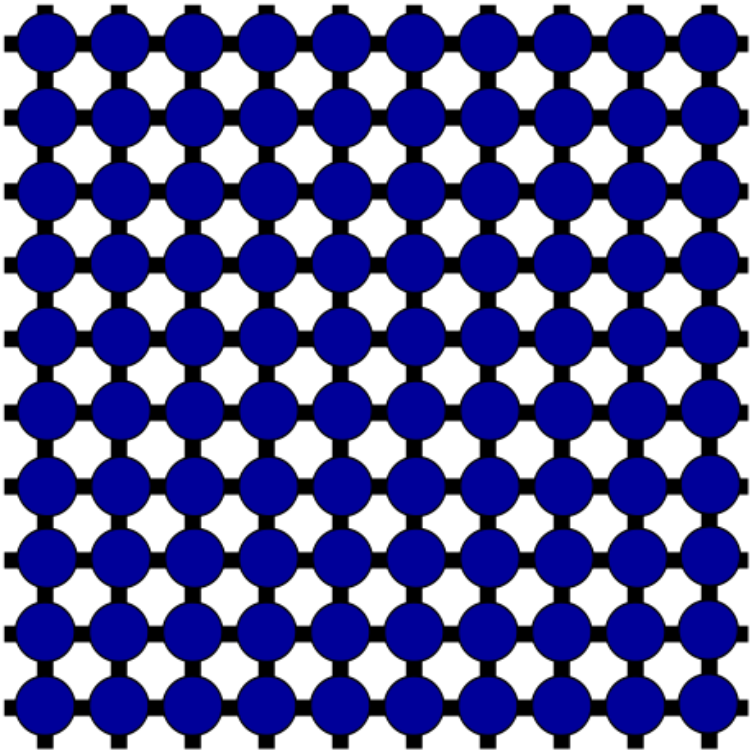
Utility computing leverages virtualized hosting environments and grid computing, with varying degrees of on-demand pricing, to allow users to purchase only the computing power and storage they require. It is heralded as a revolutionary change to how IT is planned and implemented. An important element of utility computing, the "on-demand" component has become a popular computing model that bears a close resemblance to the

way we use common, everyday products and services. Every time one “demands” that a light be turned on by flicking a switch or one picks up a telephone to hear dial tone, one is using a “non-computing” on- demand service. True on-demand computing follows the same pricing model. However, many hosted software and hardware services marketed as “on- demand” do not follow the utility-based pricing model. Instead, they employ a per-user, per-month or per-day(s) unit of measure and frequently require monthly or yearly purchase commitments. True utility computing should follow the model that if no computing power is being used, the consumer is not charged. When a demand for computing is fulfilled, the consumer is charged only for the resources he has consumed.

Chapter 6

Fabric Computing and Integrated Cloud Service Management

Fabric computing



Fabric computing or **unified computing** involves the creation of a **computing fabric** consisting of interconnected nodes that look like a 'weave' or a 'fabric' when viewed collectively from a distance.

Usually this refers to a consolidated high-performance computing system consisting of loosely coupled storage, networking and parallel processing functions linked by high

bandwidth interconnects (such as 10 Gigabit Ethernet and InfiniBand) but the term has also been used to describe platforms like the Azure Services Platform and grid computing in general (where the common theme is interconnected nodes that appear as a single logical unit).

The fundamental components of *fabrics* are "nodes" (processor(s), memory, and/or peripherals) and "links" (functional connection between nodes). While the term "fabric" has also been used in association with storage area networks and switched fabric networking, the introduction of compute resources provides a complete "unified" computing system. Other terms used to describe such fabrics include "unified fabric", "data center fabric" and "unified data center fabric".

According to Ian Foster, director of the Computation Institute at the Argonne National Laboratory and University of Chicago, "grid computing 'fabrics' are now poised to become the underpinning for next-generation enterprise IT architectures and be used by a much greater part of many organisations."

Brocade, Cisco and Egenera currently manufacture computing fabric equipment.

History

While the term has been in use since the mid to late 1990s the growth of cloud computing and Cisco's evangelism of *unified data center fabrics* followed by *unified computing* (an evolutionary data center architecture whereby blade servers are integrated or *unified* with supporting network and storage infrastructure) starting March 2009 has renewed interest in the technology. Other companies offering unified or fabric computing systems include Liquid Computing Corporation and Egenera.

There have been mixed reactions to Cisco's architecture, particularly from rivals who claim that these proprietary systems will lock out other vendors. Analysts claim that this "ambitious new direction" is "a big risk" as companies like IBM and HP who have previously partnered with Cisco on data center projects (accounting for $2-3bn of Cisco's annual revenue) are now competing with them.

Key characteristics

The main advantages of *fabrics* are that a massive concurrent processing combined with a huge, tightly-coupled address space makes it possible to solve huge computing problems (such as those presented by delivery of cloud computing services) and that they are both scalable and able to be dynamically reconfigured.

Challenges include a non-linearly degrading performance curve, whereby adding resources does not linearly increase performance which is a common problem with parallel computing and maintaining security.

Integrated Cloud Service Management

ICSM Summary

In order to take advantage of cloud computing benefits enterprise and government organizations must deploy **Integrated Cloud Service Management** solutions. ICSM is primarily concerned with reducing costs associated with switching to a new cloud-based infrastructure.

ICSM reduces cost of operations in the cloud and enables smoother migration into the cloud by:

- Extending asset management systems to track and provision cloud resources
- Providing tools that set, enforce new and existing IT governance policies in the cloud
- Establishing project and role based access control to cloud-based resources
- Facilitating secure application deployments through enhanced data and network security
- Addressing new and cloud specific security risks
- Automating infrastructure and application management tasks
- Enabling transparency in cloud infrastructure and applications lifecycle
- Providing tools to enforce and track compliance with standards such as PCI and HIPAA
- Integrating cloud specific monitoring instrumentation with existing ESM infrastructure
- Enabling secure, transparent and governed self-service for infrastructure consumers
- Integrating existing billing codes and other financial metadata with cloud-based assets
- Reducing instances of abandoned and mis-used computing infrastructure

ICSM Roles

In order to accomplish its objectives, ICSM recognizes various organizational roles that are involved in the process of managing and consuming infrastructure.

Infrastructure Managers

In a traditional non-cloud environment IMs are responsible for managing:

- Servers, storage and networking gear
- Datacenter floorspace, power and cooling
- External and internal network connectivity
- Physical and network security
- Datacenter capacity

- Contracts with hardware and network providers

In a cloud based environment IMs are responsible for:

Setting and enforcing standards that apply to individual cloud providers Negotiating contracts and service level agreements with cloud providers Deploying asset management solutions to track cloud based resources Deploying reporting tools that allow infrastructure consumers receive cloud usage statements Deploying infrastructure and application portals that enable self-service wherever possible Implementing strategy for cloud independence Implementing auditing mechanisms to track and control infrastructure management activities Evaluating and selecting cloud providers Ensuring infrastructure consumers and staff use provider neutral management tools Maintaining network connectivity and data replication between corporate and cloud data centers

ICSM recognizes this set of activities under term Cloud Provider Management and Cloud Governance.

System Administrators

System Administrators are responsible for administration of: Operating Systems Firewalls and VPN Backup Systems Middleware and Databases ERP Software Collaboration Monitoring

Under ICSM, in addition to existing responsibilities, System Administrators are responsible for maintaining their software specific virtual images and templates. The ensures that infrastructure consumers can quickly provision standardized images containing specific software such as Application Server or Sharepoint environment. System Administrators must deploy tools that help them to manage both images as well as templates where service consumers can specify application specific configuration parameters. In a cloud based environment, System Administrators need to learn how to manage environments using images and templates rather than by performing administrating on specific server instances. ICSM generally refers to these set of activities as Configuration Management.

Network and Firewall administrators must participate in cloud management solution selection process to ensure that the chosen solution can implement organizational network security policy. For example, corporate network security policy may state that by default there should be no connectivity from development to production environments. Network and Firewall administrators must then ensure that the cloud management solution implements this policy by default when virtual infrastructure is provisioned into cloud network perimeters. ICSM recognizes these activities under a term Cloud Network Policy Enforcement and Cloud Governance.

Backup Administrators must also participate in cloud provider and cloud management selection process to ensure that providers and management solutions provide adequate facilities to perform data backup that is in line with corporate standards. Backup

Administrators must find ways to extend existing backup management solutions into the cloud or deploy similar cloud-specific implementations. ICSM recognizes these activities under a term Cloud Backup.

Enterprise Monitoring group is responsible for blending cloud provider instrumentation data with existing monitoring infrastructure. ESM should select could management tools that facilitate rollup of instrumentation data from multiple cloud providers. ESM must ensure consistent deployment of application and OS specific monitoring packages to images deployed across all cloud providers. ICSM recognizes these activities under a term Cloud Monitoring.

Architecture and Information Security

Architecture and Information Security group is responsible for setting cloud provider specific default settings and security policies. Under principles of ICSM, Information Security group is responsible for setting default data and network security settings. For example, Information Security group must be able to enforce disk or network encryption for certain types of applications. IS group must also select cloud management platform where chosen security policies can be enforced. Architecture group is responsible for setting policies and usage of cloud SaaS offerings (This will be discussed in a separate white paper). ICSM recognizes these activities under a term Cloud Security and Cloud Governance.

Infrastructure Consumers

Infrastructure Consumers and consume and leverage computing infrastructure. ICSM assumes infrastructure consumers perform following key activities. Request and decommission computing, storage and network infrastructure Coordinate application new releases and QA efforts Promote application releases to new environments Restart and troubleshoot applications Modify computing, storage and network allocated to application Request access control changes to application resources Viewing cloud resource usage reports Responding to instrumentation events Request that data and network security controls enabled or disabled

Chapter 7

Message Queuing Service and Pidoco

Message queuing service

A **message queueing service** is a message-oriented middleware or MOM deployed in a compute cloud using software as a service model. Service subscribers access queues and or topics to exchange data using point-to-point or publish and subscribe patterns.

Goals

A message queueing service aims to eliminate the traditional overhead associated with operating in-house messaging infrastructures. These operating overheads include:

- Unused capacity installed to meet peak demands
- Human resources that are necessary to maintain messaging infrastructure
- Projects idle time waiting for resource provisioning
- Need to isolate messaging resources

Besides reducing cost, a message queueing service seeks to simplify access to messaging resources and therefore facilitate integration efforts within organizations and between them.

Benefits

A message queueing service also creates new value by providing reduced costs, enhanced performance and reliability. In order to provide those benefits, a message queueing service leverages cloud computing resources such as storage, network, memory and processing capacity. By using virtually unlimited cloud computing resources, a message queueing service provides an internet scale messaging platform.

Accessibility

A message queueing service is accessible through a variety of protocols such as Java Message Service, AMQP, REST-style APIs and web services.

Usage Examples

- Patient gets admitted into a hospital out of her provider's network. Producer hospital can start sending real time events about the treatment of the patient to her physician's hospital using a message queueing service platform. The cost of integration between hospitals is marginal since they do not need to configure messaging protocols, VPNs and other details.

- Information processing organization that processes events from thousands of different sources, can ask its information providers to simply place messages onto queue services and reduce integration costs.

- A Call Centre can carry on servicing requests for bills to be present when the billing system is unavailable

- Embedded telemetry devices in vehicles can securely communicate with an application that number crunches statistics in near-real time; Round-Robin messaging lets the vehicle supplier add computing resources as his sales increase.

- Security trading application can post updates to P&L application that might be unavailable at the moment.

- Technician submits an x-ray while consuming application instances in London, Chicago and Sao Palo compete who gets the message first by listening on the same queue.

Vendors

- Amazon Simple Queue Service - supports messages up to 8k; does not guarantee order or that message will be delivered once only. Supports REST API only. Utilizes Amazon Compute Cloud. Proprietary platform.
- OpenMQ - proprietary open source platform.
- Solace MQ from Solace Systems
- StormMQ - Open platform supports messages up to 50Mb. Uses AMQP to avoid vendor lock-in and provide language neutrality. Locate-It Option allows customers to audit the location of their data at all times and satisfy data protection principles.

Pidoco

The **Pidoco Usability Suite** is a cloud-based collaboration software by Pidoco GmbH for creating, sharing and testing wireframes, mockups and prototypes of websites, mobile apps and enterprise software applications.

Pidoco Usability Suite

The Pidoco Usability Suite is a cloud-based collaboration software for planning, designing and testing websites, web applications, mobile apps and enterprise software. The software tool enables users to quickly create clickable wireframes, mockups and interactive low-fidelity prototypes of GUIs without programming by using drag-and-drop placement of pre-fabricated elements. Pidoco allows users to share projects with team members and other project stakeholders for real-time online collaboration, reviewing and user testing. The web application is used to visualize requirements, collaborate in the design phases of software development, involve end users, and generate optimized specifications for traditional or agile development processes. Pidoco is browser-based and thus compatible with Windows, Mac OS and Linux.

Functionality

Pidoco is structured into several views that give access to various functions. The functions include:

- Editor – Creating, sharing and exporting wireframes, mockups and prototypes
- Simulator – Reviewing and discussing projects and requirements
- Tester – Testing prototypes for usability with end users online

Versions

The Pidoco Usability Suite is marketed in three editions:

- Basic – The Basic edition is the entry-level version which includes the editing functions.
- Classic – The Classic version includes all Basic functionality and additional sharing options as well as real-time collaborative editing capability for teams.
- Expert – The Expert version is the top-of-line version and includes all Classic functions as well as testing functionality and additional collaboration options like screensharing, VoIP connection, chat and session recording.

Compatibility

The Pidoco Usability Suite is a browser-based application that runs on various operating systems, including Windows, Mac OS and Linux. Supported browsers include Internet Explorer 7+, Firefox 3.0+, Chrome 4+, and Safari 4+ for the simulation and testing

functions that allow teams to collaborate on prototypes by gathering feedback. Creating and editing prototypes is supported in Firefox 3.0+, Chrome 4+, and Safari 4+.

Requirements

Pidoco is a JavaScript/SVG based tool that uses open web standards. It is offered on the software-as-a-service (SaaS) model and does not require download or installation. The software can be accessed online via the application homepage, so a broad-band internet connection and a web browser software are necessary to use the Pidoco software.

Software-as-a-Service

Pidoco operates on the Software-as-a-Service (SaaS) model, and is available as standard hosted SaaS option, hosted dedicated server option or as an in-house installation. Attributes of SaaS include flexibility and scalability, multi-tenant capability, recurring subscription fees rather than upfront capital cost, and require no internal hardware.

Chapter 8

Platform as a Service and Nimbus Plug Computer

Platform as a service

Platform as a Service (PaaS) is the delivery of a computing platform and solution stack as a service. PaaS offerings facilitate deployment of applications without the cost and complexity of buying and managing the underlying hardware and software and provisioning hosting capabilities, providing all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely available from the Internet.

PaaS offerings may include facilities for application design, application development, testing, deployment and hosting as well as application services such as team collaboration, web service integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation and developer community facilitation. These services may be provisioned as an integrated solution over the web.

Types

Add-on development facilities

These facilities allow customization of existing SaaS applications, and in some ways are the equivalent of macro language customization facilities provided with packaged software applications such as Lotus Notes, or Microsoft Word. Often these require PaaS developers and their users to purchase subscriptions to the co-resident SaaS application.

Stand alone development environments

Stand-alone PaaS environments do not include technical, licensing or financial dependencies on specific SaaS applications or web services, and are intended to provide a generalized development environment.

Application delivery-only environments

Some PaaS offerings lack development, debugging and test capabilities, and provide only hosting-level services such as security and on-demand scalability.

Open Platform as a Service

Lets the developer use any programming language, any database, any operating system, any server, etc.

Key characteristics

Services to develop, test, deploy, host and maintain applications in the same integrated development environment

Different PaaS offerings provide different combinations of services to support the application development life-cycle. Comprehensive PaaS should provide all service options in an integrated development environment within the actual target delivery platform, with source code control, version control, dynamic (interactive) multiple user testing, roll out and roll back with the ability to audit and track who made what changes when to accomplish what purpose

Web based user interface creation tools

PaaS offerings typically provide some level of support to ease the creation of user interfaces, either based on standards such as HTML and JavaScript or other Rich Internet Application technologies like Adobe Flex, Flash and AIR. Rich, interactive, multi-user environments and scenarios can be defined, tried out by real people (non-programmers), with tools that make it easy to log/single out features that annoy or frustrate either novices or experts. Creation tools allow interfaces to be defined for different user profiles by function or expertise. PaaS offers improved user experience by incorporating channels for real people feedback throughout creation, design, development, testing, roll-out, production...the entire life-cycle through to 'end-of-life" "reincarnation" or "next generation evolution" of the application.

Multi-tenant architecture

PaaS offerings typically attempt to support use of the application by many concurrent users, by providing concurrency management, scalability, fail-over and security. The architecture enables defining the "trust relationship" between users in security, access, distribution of source code, navigation history, user (people and device) profiles, interaction history, and application usage.

Integration with web services and databases

Support for SOAP and REST interfaces allow PaaS offerings to create compositions of multiple web services, sometimes called "mashups" as well as access databases and re-use services maintained inside private networks. Support for keeping the

user/relationships (if multiple users)/device context and profile through the mashup across web services, databases and networks.

Support for development team collaboration

The ability to form and share code with ad-hoc or pre-defined or distributed teams greatly enhances the productivity of PaaS offerings. Schedules, objectives, teams, action items, owners of different areas of responsibilities, roles (designers, developers, tester, QC) can be defined, updated and tracked based on access rights.

Utility-grade instrumentation

PaaS offerings provide developers insight into the inner workings of their applications, and the behavior of their users. Some PaaS offerings use information about user behaviour to enable pay-per-use billing. Historical/usage evidence may help:

- determine whether services are of value to users/customers,
- compare the value of different services, and
- track activity based costs and revenues.

Visualization tools could show usage patterns, exposing functional or correlational relationships between:

- services and/or user interactions,
- the value to the user or users, and
- the cost of alternative service paths such as web and cell phone

just to name a couple.

Financial data collection and, possibly, forecasting, are required to determine who pays what to whom and when (how often).

Nimbus Plug Computer

Nimbus Plug Computer

Nimbus Plug Computer(with iPhone for size comparison)

Manufacturer	Ionics EMS, Inc.
Type	Plug computer
Release date	September 2009
Operating system	Linux, Android
Power	2.3 W idle no attached devices, 7.0 W running at 100% CPU utilization
CPU	1.2 GHz ARM Marvell Kirkwood 88F6281
Storage capacity	External hard drive/flash disk
Memory	512 MB SDRAM, 512 MB Flash
Display	none (headless Server)
Connectivity	USB 2.0, Gigabit Network, *(+ SD slot & JTAG mini USB for the Development Kit Version)*
Dimensions	83mm (L) x 52mm (W) x 43mm

The **Nimbus Plug Computer** is a 2nd Generation re-design of the of the original reference design of the plug computers on the market. It features the same 1.2 GHz

Marvell Kirkwood 88F6281 ARM CPU) and has features literally the same as the reference design, except the Nimbus comes supplied with Debian 5 (Lenny) - ARM build.

What is very different is the *Ultra Compact Form Factor* and is only HALF the size of the reference design. Those who were asked to comment on it at CES2010 could not believe that it is a Computer.

The development kit version (to assist in the development of software for the platform) adds an external JTAG Kit and cable, in additional to the GCC cross-compiler for ARM. The JTAG device has a mini USB connector wired to an FTDI FT2232 chip which provides the developer's computer with access to two ports, a JTAG port connected to the internal JTAG bus, and an RS232 port connected to the Kirkwood processor's serial port through which the bootstrap and kernel console can be accessed. This debug console can be accessed from any computer with support for the FTDI bus translator (FreeBSD, Linux, Mac OS X, Windows).

Applications

Some of the applications that have been dreceloped on the Nimbus.

'**SOHO Office Communications Server'** Created on a larger PC Server in 2004 by a small software company in Singapore for its business needs. Now the same applications (and even more more) can be installed on a tiny plug computer a fraction of the size of a PC and consuming a fraction of the power.

- Postfix Email Server (with Dovecot IMAP/POP3 and Webmail Support)
- Apache Web server and PHP
- MySQL Database system
- FTP Server (and remote acess Serer)
- Samba Server
- Print Server
- Groupware Server & more

- **Low Cost Video Surveillance**

  - Use of up to 3 USB Webcams
  - Motion Software application for Motion detection and video recording
  - Saving video footage unto USB HDD
  - Alerts

- **Low Cost & Low Operating Cost Torrent Downloader**

  - Storage in USB HDD
  - Web Browser Interface

Chapter 9

Plug Computer and ProActive

Plug computer

Marvell Technology Group's SheevaPlug plug computer with iPhone (approx 115 mm x 60 mm) for scale

A **plug computer** is a small form factor server for use in the home or office. Compared to their PC-based counterparts, plug computers are lower cost, consume less power, often do not have a video card, and are intended to be powered up at all times. Although plug

computers are often enclosed in an AC power plug or AC adapter, the term "plug" also refers to "plug and play" appliance-like devices which may be in any form factor.

Suitable for running a media server, back-up services, file sharing and remote access functions such devices can be used as a bridge between in home protocols such as Digital Living Network Alliance (DLNA) & Server Message Block (SMB) and cloud based services.

History

The NSLU2 (Network Storage Link for USB 2.0 Disk Drives) was a small Network-attached storage (NAS) device manufactured by Linksys between 2004 and 2008.

A number of other devices in this form factor began to appear at the 2009 Consumer Electronics Show.

CloudPlug, a plug computer developed by CTERA Networks

- On January 6, 2009 CTERA Networks launched a device called *CloudPlug* that provides online backup at local disk speeds and overlays a file sharing service. The device also transforms any external USB hard drive into a network-attached storage device.

Seagate Dockstar, a plug computer similar to the SheevaPlug

- On January 7, 2009, Cloud Engines unveiled *Pogoplug* network access server.
- On January 8, 2009, Axentra announced availability of their *HipServ* platform.
- On February 23, 2009, Marvell Technology Group announced its plans to build a mini-industry around plug computers.
- On August 19, 2009, CodeLathe announced availability of their *TonidoPlug* network access server.
- On November 13, 2009 QuadAxis launched its plug computing device product line and development platform, featuring the *QuadPlug* and *QuadPC* and running QuadMix, a modified Linux.

- On 5 January, 2010 Iomega announced their *iConnect* network access server.
- On January 7, 2010 Pbxnsip launched its plug computing device the *sipJack* running pbxnsip: an IP Communications platform.

ProActive

	ProActive
Developer(s)	OW2 Consortium
Written in	Java
Operating system	Cross-platform
Type	Grid Computing
License	GNU General Public License

ProActive is Java grid middleware for parallel, distributed, and multi-threaded computing. It is developed by the OW2 Consortium, including INRIA, CNRS, University of Nice Sophia Antipolis, and ActiveEon. It is open-source software released under the GPL license.

ProActive provides a comprehensive framework and parallel programming model to simplify the programming and execution of parallel applications running on multi-core processors, distributed on Local Area Network (LAN), on clusters and data centers, on intranets, and on Internet grids.

The ProActive programming model combines the active object design pattern with futures objects.

Programming model

The model was created by Denis Caromel, professor at University of Nice Sophia Antipolis. Several extensions of the model were made later on by members of the OASIS team at INRIA. The book *A Theory of Distributed Objects* presents the ASP calculus that formalizes ProActive features, and provides formal semantics to the calculus, together with properties of ProActive program execution.

Active objects

Active objects are the basic units of activity and distribution used for building concurrent applications using ProActive. An active object runs with its own thread. This thread only

executes the methods invoked on this active object by other active objects, and those of the passive objects of the subsystem that belongs to this active object. With ProActive, the programmer does not have to explicitly manipulate Thread objects, unlike in standard Java.

Active objects can be created on any of the hosts involved in the computation. Once an active object is created, its activity (the fact that it runs with its own thread) and its location (local or remote) are perfectly transparent. Any active object can be manipulated as if it were a passive instance of the same class.

An *active object* is composed of two objects: a *body*, and a standard Java object. The body is not visible from the outside of the active object.

The body is responsible for receiving calls (or *requests*) on the active object and storing them in a queue of pending calls. It executes these calls in an order specified by a synchronization policy. If a synchronization policy is not specified, calls are managed in a "First in, first out" (FIFO) manner.

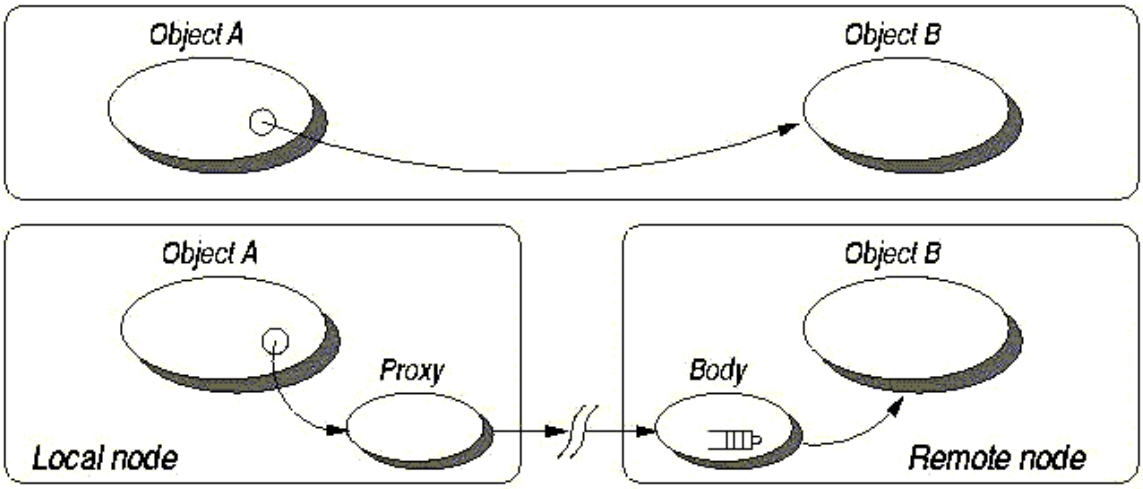
The thread of an active object then chooses a method in the queue of pending requests and executes it. No parallelism is provided inside an active object; this is an important decision in ProActive's design, enabling the use of "pre-post" conditions and class invariants.

On the side of the subsystem that sends a call to an active object, the active object is represented by a *proxy*. The proxy generates future objects for representing future values, transforms calls into Request objects (in terms of metaobject, this is a reification) and performs deep copies of passive objects passed as parameters.

Active object basis

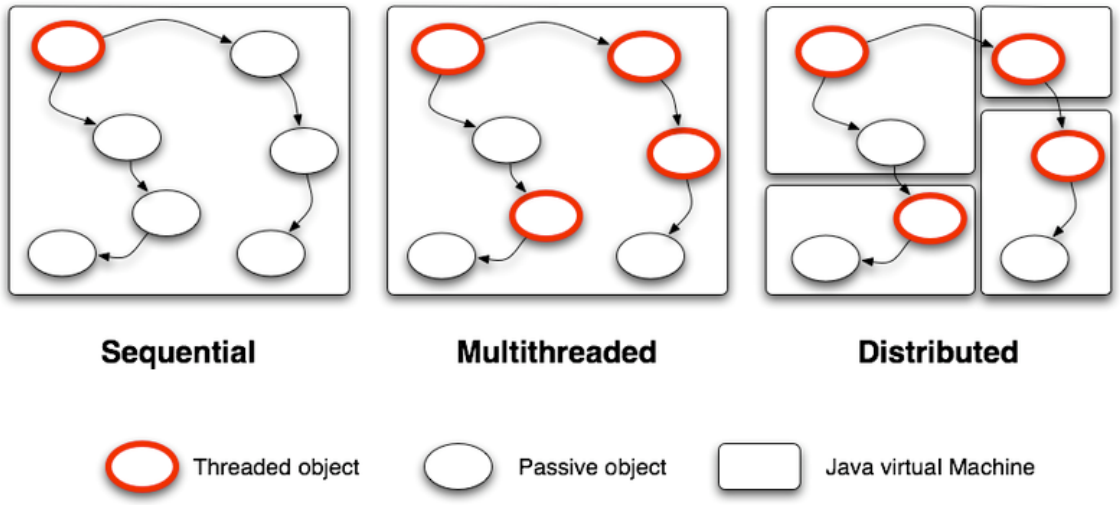
ProActive is a library designed for developing applications in the model introduced by Eiffel//, a parallel extension of the Eiffel programming language.

In this model, the application is structured in *subsystems*. There is one active object (and therefore one thread) for each subsystem, and one subsystem for each active object (or thread). Each subsystem is thus composed of one active object and any number of passive objects—possibly no passive objects. The thread of one subsystem only executes methods in the objects of this subsystem. There are no "shared passive objects" between subsystems.



A call onto an active object, as opposed to a call onto passive one

These features impact the application's topology. Of all the objects that make up a subsystem—the active object and the passive objects—only the active object is known to objects outside of the subsystem. All objects, both active and passive, may have references onto active objects. If an object *o1* has a reference onto a passive object *o2*, then *o1* and *o2* are part of the same subsystem.



The model: Sequential, multithreaded, distributed

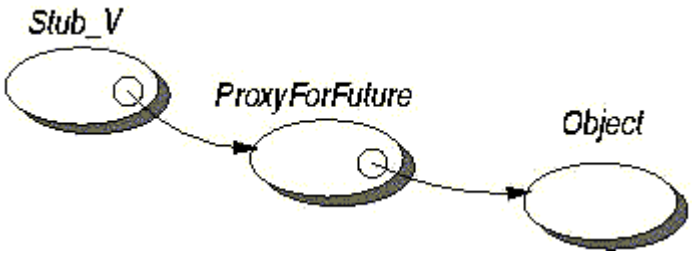
This has also consequences on the semantics of message-passing between subsystems. When an object in a subsystem calls a method on an active object, the parameters of the call may be references on passive objects of the subsystem, which would lead to shared passive objects. This is why passive objects passed as parameters of calls on active objects are always passed by deep-copy. Active objects, on the other hand, are always

passed by reference. Symmetrically, this also applies to objects returned from methods called on active objects.

Thanks to the concepts of asynchronous calls, futures, and no data sharing, an application written with ProActive doesn't need any structural change—actually, hardly any change at all—whether it runs in a sequential, multi-threaded, or distributed environment.

Asynchronous calls and futures

Whenever possible, a method call on an active object is reified as an asynchronous request. If not possible, the call is synchronous, and blocks until the reply is received. If the request is asynchronous, it immediately returns a *future object*.



A future object

The future object acts as a placeholder for the result of the not-yet-performed method invocation. As a consequence, the calling thread can go on with executing its code, as long as it doesn't need to invoke methods on the returned object. If the need arises, the calling thread is automatically blocked if the result of the method invocation is not yet available. Although a future object has structure similar to that of an active object, a future object is not active. It only has a Stub and a Proxy.

A simple example

The code excerpt below highlights the notion of future objects. Suppose a user calls a method `foo` and a method `bar` from an active object `a`; the `foo` method returns void and the `bar` method returns an object of class `V`:

```
// a one way typed asynchronous communication towards the (remote) AO a
// a request is sent to a
a.foo (param);

// a typed asynchronous communication with result.
// v is first an awaited Future, to be transparently filled up after
// service of the request, and reply
V v = a.bar (param);
...
// use of the result of an asynchronous call.
// if v is still an awaited future, it triggers an automatic
// wait: Wait-by-necessity
```

```
v.gee (param);
```

When `foo` is called on an active object `a`, it returns immediately (as the current thread cannot execute methods in the other subsystem). Similarly, when `bar` is called on `a`, it returns immediately but the result `v` can't be computed yet. A future object, which is a placeholder for the result of the method invocation, is returned. From the point of view of the caller subsystem, there is no difference between the future object and the object that would have been returned if the same call had been issued onto a passive object.

After both methods have returned, the calling thread continues executing its code as if the call had been effectively performed. The role of the future mechanism is to block the caller thread when the `gee` method is called on `v` and the result has not yet been set : this inter-object synchronization policy is known as *wait-by-necessity*.

Chapter 10

Rich Internet Application and Sector/Sphere

Rich Internet application

A **Rich Internet Application** (**RIA**) is a Web application that has many of the characteristics of desktop applications, typically delivered either by way of a site-specific browser, via a browser plug-in, independent sandboxes, or virtual machines. Adobe Flash, Java, and Microsoft Silverlight are currently the three most common platforms, with penetration rates around 99%, 80%, and 54% respectively (as of July 2010). Although new Web standards have emerged, they still use the principles behind RIAs.

Users generally need to install a software framework using the computer's operating system before launching the application, which typically downloads, updates, verifies and executes the RIA. This is the main differentiator from JavaScript-based alternatives like Ajax that use built-in browser functionality to implement comparable interfaces. While some consider such interfaces to be RIAs, some consider them competitors to RIAs and others, including Gartner, treat them as similar but separate technologies.

RIAs dominate in online gaming as well as applications that require access to video capture (with the notable exception of Gmail, which uses its own task-specific browser plug-in). Web standards such as HTML5 have developed and the compliance of Web browsers with those standards has somewhat improved. However, the need for plug-in based RIAs for accessing video capture and distribution has not diminished, even with the emergence of HTML5 and JavaScript-based desktop-like widget sets that provide alternative solutions for mobile Web browsing.

History

The term "rich Internet application" was introduced in a white paper of March 2002 by Macromedia (now merged into Adobe), though the concept had existed for a number of years earlier under names such as:

- Remote Scripting, by Microsoft, circa 1999
- X Internet, by Forrester Research in October 2000
- Rich (Web) clients
- Rich Web application

Design, measurement, performance

Rich Internet applications use a distributed-function model rather than the simple thin-client–server model.

Flash, Silverlight and Java enrich user experiences in part due to their reduced reliance on network/server communications. Since 80% of the time is spent to download all the components of the page, simplifying a page's design is also a way to reduce response time. Another way to tackle this is to reduce HTTP requests by combining all HTTP requests and CSS style sheets.

Characteristics

RIAs present indexing challenges to search engines, but Adobe Flash content is now at least partially indexable.

Security can improve over that of application software (for example through use of sandboxes and automatic updates), but the extensions themselves remain subject to vulnerabilities and access is often much greater than that of native Web applications. For security purposes, most RIAs run their client portions within a special isolated area of the client desktop called a sandbox. The sandbox limits visibility and access to the file-system and to the operating system on the client to the application server on the other side of the connection. This approach allows the client system to handle local activities, calculations, reformatting and so forth, thereby lowering the amount and frequency of client-server traffic, especially as compared to the client-server implementations built around so-called thin clients.

Sector/Sphere

	Sector/Sphere
Developer(s)	The Sector Alliance
Stable release	2.5 / September 29, 2010; 6 months ago
Development status	Active

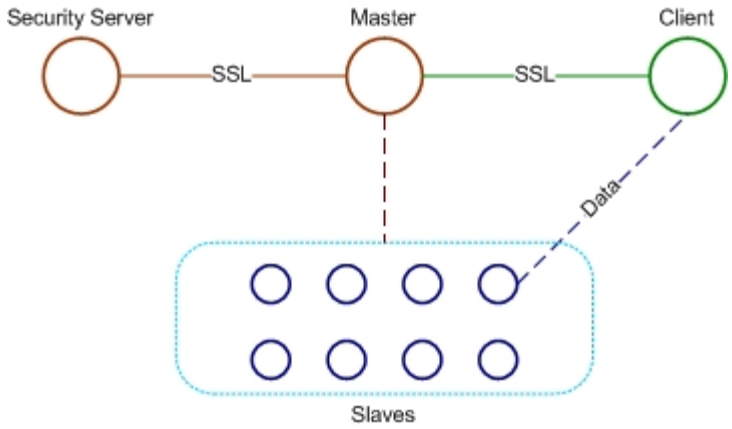
Written in	C++
Operating system	Linux / Windows
Type	Distributed File System
License	Apache License 2.0

Sector/Sphere is an open source software suite for high-performance distributed data storage and processing. It can be broadly compared to Google's GFS/MapReduce stack. Sector is a distributed file system targeting data storage over a large number of commodity computers. Sphere is the programming framework that supports massive in-storage parallel data processing for data stored in Sector. Additionally, Sector/Sphere is unique in its ability to operate in a wide area network (WAN) setting.

The system was created by Dr. Yunhong Gu (the author of UDT) in 2006 and it is now maintained by a group of open source developers.

Architecture

Sector/Sphere consists of four components. The security server maintains the system security policies such as user accounts and the IP access control list. One or more master servers control operations of the overall system in addition to responding to various user requests. The slave nodes store the data files and process them upon request. The clients are the users' computers from which system access and data processing requests are issued.



Sector Distributed File System

Sector is a user space file system which relies on the local/native file system of each node for storing uploaded files. Sector provides file system-level fault tolerance by replication, thus it does not require hardware fault tolerance such as RAID, which is usually very expensive.

Sector does not split user files into blocks; instead, a user file is stored intact on the local file system of one or more slave nodes. This means that Sector has a file size limitation that is application specific. The advantages, however, are that the Sector file system is very simple, and it leads to better performance in Sphere parallel data processing due to reduced data transfer between nodes. It also allows uploaded data to be accessible from outside the Sector system.

Sector provides many unique features compared to traditional file systems. Sector is topology aware. Users can define rules on how files are located and replicated in the system, according to network topology. For example, data from a certain user can be located on a specific cluster and will not be replicated to other racks. For another example, some files can have more replicas than others. Such rules can be applied at per-file level.

The topology awareness and the use of UDT as data transfer protocol allows Sector to support high performance data IO across geographically distributed locations, while most file systems can only be deployed within a local area network. For this reason, Sector is often deployed as a content distribution network for very large datasets.

Sector integrates data storage and processing into one system. Every storage node can also be used to process the data, thus it can support massive in-storage parallel data processing. Sector is application aware, meaning that it can provide data location information to applications and also allow applications to specify data location, whenever necessary.

As a simple example of the benefits of Sphere, Sector can return the results from such commands as "grep" and "md5sum" without reading the data out of the file system. Moreover, it can compute the results of multiple files in parallel.

The Sector client provides an API for application development which allows user applications to interact directly with Sector. The software also comes prepackaged with a set of command-line tools for accessing the file system. Finally, Sector supports the FUSE interface; presenting a mountable file system that is accessible via standard command-line tools.

Sphere Parallel Data Processing Engine

Sphere is a parallel data processing engine integrated in Sector and it can be used to process data stored in Sector in parallel. It can broadly compared to MapReduce, but it uses generic User Defined Functions (UDFs) instead of the map and reduce functions. A UDF can be either a map function or a reduce function, or even others.

Benefiting from the underlying Sector file system and the flexibility of the UDF model, Sphere can manipulate the locality of both input data and output data, thus it can effectively supports multiple input datasets, combinative and iterative operations, and even legacy application executable.

Due to the fact that Sector does not split user files, Sphere can simply wrap up many existing applications that accepts files or directories as input, without rewriting them. Thus it can provide greater compatibility to legacy applications.

Chapter 11

Software as a Service

Software as a service sometimes referred to as "on-demand software," is a software delivery model in which software and its associated data are hosted centrally (typically in the (Internet) cloud) and is accessed by users using a thin client, typically a web browser over the Internet.

While practically every Internet service (such as Web search engine or web-based Email) is driven by some underlying software, the term *software-as-a-service* is often used in the context of business software, and in some cases even more narrowly as software in a category which has on-premise software; i.e., equivalent software that is installed in businesses' computer networks or personal computers.

SaaS has become a common delivery model for most business applications, including accounting, collaboration, customer relationship management (CRM), enterprise resource planning (ERP), invoicing, human resource management (HRM), content management (CM) and service desk management.

History

Centralized hosting of business application dates back to the 1960s. Starting at that decade, IBM and other mainframe providers conducted a service bureau business, often referred to as time-sharing or utility computing. Such services included offering computing power and database storage to banks and other large organizations from their worldwide data centers.

The expansion of the Internet during the 1990s brought about a new class of centralized computing, called Application Service Providers (ASP). Application service providers provided businesses with the service of hosting and managing specialized business applications, with the goal of reducing cost by central administration and through the solution provider's specialization in a particular business application.

Software-as-a-service is essentially an extension of the idea of the ASP model. The term software-as-a-service, however, is commonly used in more specific settings: whereas most initial application service providers focused on managing and hosting third-party

independent software vendors' software, contemporary software-as-a-service vendors typically develop and manage their own software; whereas many initial application service providers offered more traditional client-server applications, which require installation of software on users' personal computers, contemporary software-as-a-service solutions are predominantly web-based and only require an internet browser to use; and, whereas the software architecture used by most initial application service providers mandated maintaining a separate instance of the application for each business, contemporary software-as-a-service solutions normally utilize a multi-tenant architecture, in which the application is designed to serve multiple businesses and partitions its data accordingly.

The concept of software-as-a-service has been popularized by Salesforce.com, which coined the term "The End of Software" to differentiate its (then new) approach from traditional on-premise software.

The software-as-a-service acronym, *SaaS*, first appeared in an article called "Strategic Backgrounder: Software as a Service", published in February 2001 by the Software & Information Industry's (SIIA) eBusiness Division.

Key characteristics

Software and business professionals generally associate the term SaaS with business software, and as a possibly lower-cost way for businesses to use software as needed rather than license every application on every device. With a well-designed implementation and properly priced licenses, on-demand SaaS provides license benefits without the associated complexity and the potential high cost to equip devices with applications they may not need.

SaaS characteristics include:

- Network-based access to, and management of, commercially available software
- Activities managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the Web
- Application delivery typically closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering and management characteristics
- Centralized feature updating, which obviates the need for end-users to download patches and upgrades.
- Frequent integration into a larger network of communicating software—either as part of a mashup or a plugin to a platform as a service

(Service oriented architecture is naturally more complex than traditional models of software deployment.)

SaaS providers generally price applications on a per-user basis and/or per business basis, sometimes with a relatively small minimum number of users and often with additional

fees for extra bandwidth and storage. SaaS revenue streams to the vendor are therefore lower initially than traditional software license fees, but are also recurring, and therefore viewed as more predictable, much like maintenance fees for licensed software.

Some SaaS applications are free to the user, with revenue being derived from alternate sources such as advertising, or upgrade fees for enhanced functionality (often referred to as "freemium"). Examples of free SaaS applications include large players such as Gmail, Google Docs and Zoho, as well as smaller providers like Wave Accounting (free accounting), Freshbooks (freemium time tracking and invoicing) and Load Impact (freemium load testing). Examples of paid SaaS applications are Salesforce.com for CRM, Netsuite for accounting and Locus Technologies for environmental and GHG information management.

In addition to characteristics mentioned above, SaaS sometimes provides:

- More feature requests from users, since there is frequently no marginal cost for requesting new features
- Faster new feature releases, since the entire community of users provides feedback and suggestions for improvement and derives benefits from upgrades
- Embodiment of recognized best practices, since the user community drives the software publisher to support best practice. The publisher benefits from the "wisdom of the crowd".

Benefits

Some of the claimed benefits for SaaS are:

- Capital expenditure is reduced by not having to purchase servers or full copies of software. This is counterbalanced by the increased revenue cost of paying for the use of the SaaS
- Faster implementation. In some cases the customer can deploy SaaS more quickly as no local installation is required.
- Depending on the user, it may remove a non-core activity (deployment and support of the software and its associated infrastructure) freeing up time to focus on core business activities
- Reduced need to predict scale of demand and infrastructure investment up front
- Possible improvements to reliability if the SaaS provider's infrastructure is more redundant or has higher availability than the user would otherwise have

Advantages

- Accessible from anywhere with an internet connection
- No local server installation
- Pay per use or subscription based payment methods
- Rapid scalability
- System maintenance (backup, updates, security, etc) often included in service

- Possible security improvements, although users with high security requirements (e.g., large corporations) may find SaaS a security concern
- Reliability
- Reduced time to market

Implementation

One of the companies that sells that kind of service classifies SaaS into four "maturity levels," whose key attributes are configurability, multi-tenant efficiency, and scalability. Each level is distinguished from the previous one by the addition of one of those three attributes:

- Level 1 – Ad-hoc/custom: Each customer has a customized version of the hosted application that runs as its own instance on the host's servers. Migrating a traditional non-networked or client–server application to this level of SaaS typically requires the least development effort, and reduces operating costs by consolidating server hardware and administration.
- Level 2 – Configurable: This adds greater program flexibility through configurable metadata, so many customers use separate instances of the same application code. This lets the vendor meet different customer needs through detailed configuration options, while simplifying common code base maintenance and updating.
- Level 3 – Configurable, multi-tenant-efficient: This adds multi-tenancy to the second level, so a single program instance serves all customers. This enables more efficient server resource use without apparent difference to the end user, but ultimately faces scalability limits.
- Level 4 – Scalable, configurable, multi-tenant-efficient: The fourth and final SaaS maturity level adds scalability through a multitier architecture that supports a load-balanced farm of identical application instances that run on a variable number of servers. The provider can adjust system capacity to match demand by adding or removing servers without further altering the software architecture.

SaaS architectures may also use virtualization, either in addition to multi-tenancy, or in place of it. A principal virtualization benefit is that it can increase system capacity without additional programming. On the other hand, much programming may be required to construct a more efficient multi-tenant application. Combining multi-tenancy and virtualization provides still greater flexibility to tune the system for optimal performance. In addition to full operating system-level virtualization, other virtualization techniques applied to SaaS include application virtualization and virtual appliances.

SaaS application development may use various types of software components and frameworks. These tools can reduce time-to-market and the cost of converting a traditional on-premise software product or building and deploying a new SaaS solution. Examples include components for subscription management, grid computing software, web application frameworks, and complete SaaS platform products.

SaaS and SOA

Much like other software, SaaS can also take advantage of Service Oriented Architecture to let software applications communicate with each other. Each software service can act as a service provider, exposing its functionality to other applications via public brokers, and can also act as a service requester, incorporating data and functionality from other services. Enterprise Resource Planning (ERP) Software providers leverage SOA in building their SaaS offerings.

Software as a secure service (SaSS) is a variation of SaaS that provides security in the link to the service, content storage on the service, and non-proprietary format data backups and restores of data stored on the service.

Adoption

Drivers

A traditional rationale for outsourcing IT systems involves applying economies of scale to application operation, i.e., an outside service provider can offer better, cheaper, more reliable applications. SaaS-based application use has grown dramatically. A Gartner survey in July 2009 found that customers are "somewhat satisfied". Several important changes to the way people work have facilitated this rapid acceptance:

- Fast, low-cost broadband is available.
- Computers have become widespread—most information workers have at least basic computer skills.
- Computing has become a commodity. In the past, corporate mainframes were jealously guarded as strategic advantages. More recently, applications were viewed as strategic. Today, people know it's the business processes and the data itself (customer records, workflows, pricing information) that matters. Computing and application licenses are cost centers, and as such, they're suitable for cost reduction and outsourcing. The adoption of SaaS could also drive Internet-scale to become a commodity.
- Insourcing IT systems requires expensive overhead including salaries, health care, liability, and physical building space.
- Applications have tended to standardize. With notable, industry-specific exceptions, most people spend most of their time using standardized applications. An expense-reporting page, an applicant screening tool, a spreadsheet, or an e-mail system are all sufficiently ubiquitous and well understood that most users can switch from one system to another easily. This is evident from the number of web-based calendaring, spreadsheet, and e-mail systems that have emerged in recent years.
- Parametric applications are usable. In older applications, one could often only change a workflow by modifying the code. In more recent applications, particularly web-based ones, significantly new applications can be created from parameters and macros. This allows organizations to create different kinds of

business logic on a common application platform. Many SaaS providers allow a wide range of customization within a basic set of functions.

- A specialized software provider can target global markets. A company that made software for human resource management at boutique hotels might once have had a hard time finding enough of a market to sell its applications. But a hosted application can instantly reach the entire market, making specialization within a vertical market not only possible, but preferable. This in turn means SaaS providers can often deliver products that meet specific market needs better than traditional "shrinkwrap" applications.
- Web systems demonstrate reliability. Despite sporadic outages and slow-downs, most people are willing to use the public Internet, the Hypertext Transfer Protocol and the TCP/IP stack to deliver business functions to end users.
- Security is sufficiently well trusted and transparent. With the broad adoption of SSL, organizations have a way of reaching their applications without the complexity and burden of end-user configurations or VPNs.
- Enablement technology (tools, libraries, etc,) is available. According to IDC, organizations developing enablement technology that allow other vendors to quickly build SaaS applications will play an important role in driving the adoption of SaaS. Because of SaaS' relative infancy, many companies have either built enablement tools or platforms or are in the process of engineering enablement tools or platforms. A Saugatuck study shows the industry will most likely converge to three or four enablers that will act as SaaS Integration Platforms (SIPs).
- Wide-area network bandwidth has grown drastically, following Moore's Law (more than 100% increase each 24 months), and is about to reach slow local networks bandwidths. Added to network quality improvement, this has driven people and companies to trustfully access remote locations and applications with low latencies and acceptable speeds.
- SaaS has "democratized" software, allowing small and medium businesses to access functionality formerly the domain of large enterprises. Many analytical software tools have been released as SaaS applications on a monthly subscription basis.
- SaaS facilitates data aggregation. Instead of collecting data from multiple data sources with different database schemas, all data for all customers is stored in a single database schema (i.e., multi-tenant). This simplifies running queries across customers, mining data, and looking for trends.
- The rise of third-party SaaS data escrow services has reduced some security concerns by allowing application data to be held with an independent third party.

Adoption challenges

Although it has numerous significant benefits that make it compelling, SaaS still needs to overcome few significant obstacles to widespread adoption.

- Data being stored on the vendor’s servers, security is a major concern for SaaS users.

- Given SaaS applications are hosted over the Internet, performance and load management are critical to SaaS success. Failure of an application may result in loss of data, which in turn may lead to considerable recovery costs or loss in business.
- It is important to consider latency parameters before committing to a migration to SaaS model. Hosting applications far away from the servers at a cloud storage site introduces additional latency into the environment. This is a primary reason why the SaaS model is not suitable for applications that demand sub-second response times.

Sales channels

With products focussed towards the mid market, direct selling can become an expensive undertaking. SaaS companies seek alternatives by selling through value-added resellers (VARs), Managed Service Providers (MSPs), Master Managed Service Providers (MMSPs), and similar alliance partners. However, since SaaS is not only a different delivery mechanism, but a different business model and different technology, selling through channels has its own challenges.

Pricing models

SaaS applications provide the opportunity to implement pricing models that establish and maintain recurring revenue streams. Most SaaS vendors charge a monthly hosting or subscription fee. Opportunities also exist to charge per transaction, event, or other unit of value. These alternative pricing models exist because customers "lease" the software from the vendors and the vendors can view all transactional activity.

User satisfaction

Gartner's 2008 survey of 333 enterprises in the US and UK found a low level of approval from customers, describing overall satisfaction levels as "lukewarm." Respondents who decided against SaaS cited high service cost, integration difficulty, and technical requirements. A recent report from Forrester, “The ROI of Software-As-A-Service,” examined a range of companies that chose SaaS solutions and found that SaaS does result in long-term value. Companies interviewed for the report cited several reasons for their ROI of SaaS:

- Rapid deployment
- Increased user adoption
- Reduced support needs
- Lower implementation and upgrade costs

Criticism

Richard Stallman strongly criticizes SaaS. According to Stallman, using SaaS can cause as much harm as proprietary software, since users cannot study nor modify the particular software they use, thus, they cannot control their own computing.

Chapter 12

Steam (Software)

Steam is a digital distribution, digital rights management, multiplayer and communications platform developed by Valve Corporation. It is used to distribute games and related media online, from small independent developers to larger software houses. Steam also has community features, automated game updates, and in-game voice and chat functionality.

As of March 2011, there are over 1,250 games available through Steam, and over 30 million active user accounts. Although Valve never releases sales figures, Steam is estimated to have a 70% share of the digital distribution market for video games.

Many major publishers have large catalogues available on Steam, including: Bethesda Softworks, Electronic Arts, Activision, and Rockstar Games.

Client functionality

Steam includes a digital storefront called the Steam Store, through which users can purchase computer games digitally. Once purchased, software is permanently attached to the user's Steam account (however "gifting" of games to other accounts is possible in some limited circumstances). Content is delivered using a proprietary file transfer protocol from an international network of servers. While logged in, games can be downloaded to any computer that has the Steam client installed. Steam includes a server browser for users to search, filter, bookmark, and join Internet and LAN servers for games that integrate with it. It can be accessed from the desktop or from a games menu system, and queries friends to show a list of servers to which a user's contacts are connected. Currently accounts cannot be merged or deleted by their creators.

Once patches for a user's installed software have been uploaded to Steam, they are applied the next time the user logs in or launches a game. Once a patch has been applied, it cannot be removed unless a patch is released which reverses the previous one. Individual games can be set to only update when requested by the user. Steam requires games to be fully patched before they can be played online, however.

Valve Anti-Cheat (VAC), Valve's proprietary anti-cheat system, is available for online game servers to use. VAC-secured servers automatically detect users who are using cheats.

Localization

Steam is currently available in the following languages: Czech, Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Romanian, Russian, Simplified Chinese, Traditional Chinese, Spanish, Swedish, Turkish and Thai.

Steam games are typically purchased through the integrated Steam Store via a web-based basket/checkout system. Steam sells its products in US dollars, Euros and pounds sterling, based on the user's location.

Mods

The Steam interface allows for user-defined shortcuts to be added. In this way third-party mods, and games not purchased through the Steam Store, can use Steam features. Valve sponsors and distributes some mods for free, and mods that use Steamworks can also use VAC, Friends, the server browser, and any Steam features supported by their parent game.

Multiplayer lobbies and matchmaking

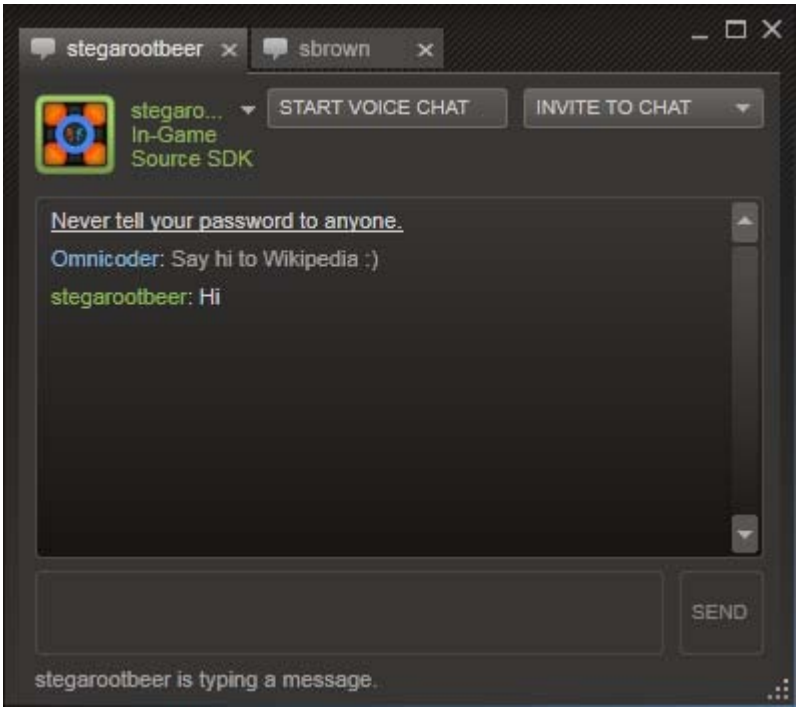
Introduced in *Left 4 Dead* and made available through Steamworks, a lobby system allows for players to organize and agree on game settings before joining a server and a matchmaking system can automatically group players together based on a certain criteria.

Steam Community

The Steam Community's homepage

On September 12, 2007, Valve released the Steam Community, a social network service that allows Steam users to communicate with each other on a many-to-many scale. It is accessible from both the desktop (in a web browser or the Steam client) and through an "overlay" program that can be viewed on top of 3D-accelerated games. In January 2010, Valve reported that 10 million of the 25 million active Steam accounts had signed up to Steam Community.

Notwithstanding privacy settings, a user's page includes some brief personal information, links to any friends' user pages, details of any games owned, the number of hours of playtime during the past two weeks, a 0–10 "Steam Rating" of activity, and links to any groups of which the user is a member. Users can also receive a feed of their friends' actions, including groups joined, games purchased, and Steam Achievements earned.



An IM conversation in Steam

Friends, Steam's instant messaging tool, supports both one-to-one and group conversations, held publicly or privately, and Peer-to-Peer VoIP. It provides extended information about what games each user is playing, allowing others to join their contacts in Steam-integrated multiplayer games.

On July 1, 2010, Valve made an announcement about Steam Community being able to act as an OpenID provider, thus enabling authentication of user's SteamID without requiring them to enter their Steam username or password on any third-party site.

Steam Cloud

Valve provides a service called Steam Cloud. For supported games, Steam Cloud stores various amounts of game data, such as keyboard shortcut settings and single-player game saves, on a central server. Any changes to relevant game files are uploaded to the main server, and newer files are automatically downloaded and used when a game is started. Valve launched the service simultaneously with the release of *Left 4 Dead*, and the service now supports most newer Valve games. It is one of the services offered to game developers through Steamworks. Users can individually disable Steam Cloud on a per-game and per-account basis.

Downloadable content

Steam offers a framework for selling and distributing downloadable content (DLC) for games. DLC, if available, is listed on the game's Store page. New DLC releases are listed alongside full games in the "New Releases" section on the storefront.

File system

Steam-integrated games are stored as single non-compressed archive files with the extension `.gcf` (an initialism for **G**ame **C**ache **F**ile). Steam allocates space on the user's hard disk for `.gcf` files before downloading in order to reduce fragmentation which may occur when downloading large files and performing disk access. Game Cache Files help to make games more portable, stop users from accidentally overwriting important files, allow for easy modification of resources, and allows for validation of content for errors. For games that do not integrate, a 'No Cache File' system is provided. Here, a `.ncf` index file points to a directory of loose files somewhere else on the system.

Retail boxed games

Users who buy selected boxed games sold at retail stores are required to register the game on Steam with the included CD key, which will then be attached to the user's Steam account in the same way as a digital purchase. The game will then act in every way like a digital copy. All of Valve's post-Steam games are sold in this fashion, as are a growing number of third party titles (most notably the *Call of Duty* series, from *Modern Warfare 2* onward).

Statistics collection

Steam collects and reports anonymous metrics of its usage, stability, and performance. With the exception of Valve's hardware survey, most collection occurs without notifying the user or offering an opt-out. Some of these metrics are available publicly, such as what games are being played or statistics on player progress in certain games. Valve has also used information from these statistics to justify implementing new features in Steam, such as the addition of a defragmentation option for game caches. Announced on July 15, Steam will soon start offering users to allow Steam to collect the details of what programs are installed on their system which are listed within the Windows' Programs and Features control panel.

Browser

Steam includes a tabbed Webkit based browser which can be accessed in-game.

Steam Guard

Steam Guard functionality was added in March 2011 to the Steam client. Steam Guard takes advantage of the identity protection provided with Intel's second generation Core

processors and compatible motherboard hardware to allow the user to lock their account to a specific computer. Once locked, activity by that account on other computers must first be approved of by the user on the locked computer. This feature is aimed to prevent Steam accounts from being hijacked by malicious users. Support APIs for Steam Guard are available to third-party developers through Steamworks.

An alternative option available to users interested in using Steam Guard is two-factor authentication through the use of a one-time verification code sent to a verified email address associated with the Steam account. Many of Steam Guard's features will work the same with the only real difference being the method of authentication.

Steamworks

On January 28, 2008, Valve released Steamworks, a free development and publishing suite (granted at Valve's discretion or with an Unreal Engine 3 license) that gives developers access to every component of Steam. Specifically, Steamworks provides means of games to integrate with the Steam client, including networking and player authentication tools for both server and peer-to-peer multiplayer games, matchmaking services, support for Steam community friends and groups, Steam statistics and achievements, integrated voice communications, and Steam Cloud support; the API also provides for anti-cheating devices and digital copy management. Steamworks can be combined with a standard Steam distribution agreement, the latter of which gives it advertising space in the Steam store but also provides Valve with a share of revenue; *Audiosurf* became the first game to be released in this way in February 2008. Several major games have since implemented Steamworks, including *Aliens vs. Predator*, *Call of Duty: Modern Warfare 2*, *Fallout: New Vegas*, *Unreal Tournament 3*, and *Warhammer 40,000: Dawn of War II*. Most games using the Steamworks API also opt for a presence in the Steam store. The only known exceptions (since Valve does not make announcements about such games) are *NBA 2K9*, and *Supreme Commander 2* for Mac OS X.

PlayStation 3

Portal 2's PlayStation 3 release will include several Steamworks features, including cross-platform instant messaging, Steam Cloud for saved games, and the ability for PS3 owners to download Portal 2 from Steam (Windows or Mac) at no extra cost. Cross-platform play is also supported. Valve said they "hope to expand upon this foundation with more Steam features and functionality in DLC and future content releases".

Market share and impact

Sales figures for Steam are hard to come by. Despite claiming that Steam is "wildly more profitable" than retail, Valve never reveals overall sales. *Forbes* has stated, however, that Steam sales contribute 50 to 70% of the $4 billion market for downloaded PC games, and that Steam offers gross margins of 70%, compared with 30% at retail.

Also, some independent developers have discussed the benefits of their games being available through Steam and its promotions. *Recettear: An Item Shop's Tale* broke over 100,000 units, far exceeding the expected 10,000 units, which its localization distributor, Carpe Fulgur, attribute in part to Steam and its sales. *Magicka* sold 30,000 copies on its day of release in January 2011, placing it at the top of Steam's "top sellers" list, and went on to sell 200,000 in 17 days. *Garry's Mod* sold 5,729 copies on its day of release in November 2006, and 312,541 in its first two years (with yearly sales growth of 33%).

Promotions

Steam regularly offers a variety of promotions, which include Guest Passes, Free Weekends, and weekend-long sales.

"Guest Passes" are allocated to a user when he or she purchases an applicable game. The user can then share the passes with others who have not purchased the game, allowing the new user to play the game for a limited time (which varies depending on the game). Once an activated guest pass expires, the recipient will be prompted to purchase the game in order to continue playing. The number of guest passes available to a game purchaser is determined on a game-by-game basis, and they expire one month after being granted if not used.

"Free Weekends" are multi-player promotions in which a game becomes free to play on Steam for a weekend. When the promotion ends, the game files remain on the participating users' computers, but they are barred from playing further until they purchase the game.

Valve runs the subscription-based Valve Cyber Café Program, which is the only legal way for a cyber café to offer Steam-based games.

From December 6 to December 20, 2010, Steam had an ongoing "multi-game game" called "The Great Steam Treasure Hunt", where various objectives, both in Steam itself and in numerous games, where presented for a day, and the completion of ten or more gave the user a chance to win 100 free games at the end of it, as well as ongoing chances for users to win the top five games in their wish lists.

History

Beginnings

Prior to Steam, Valve had problems releasing updates for their online games, such as *Counter-Strike*, wherein a patch would result in the disconnection of the larger part of the online user base for several days. They decided to make a platform which would update games automatically, and implement better anti-piracy and anti-cheat measures. Valve originally approached several companies—including Microsoft, Yahoo!, and RealNetworks—to build a client with these features, but all turned them down.

Steam's development began at an uncertain date prior to 2002. Working titles included "Grid" and "Gazelle". It was revealed to the public on 22 March 2002 at the Game Developers Conference, and was presented purely as a distribution network. To demonstrate the ease of integrating Steam to a game, Relic Entertainment had created a special version of *Impossible Creatures*. The game was ultimately not released on Steam, however.

Valve partnered with a number of companies including AT&T, Acer and GameSpy Industries. The first mod on the system was *Day of Defeat*.

The Steam client was first made available for download in 2002 during the beta period for *Counter-Strike* 1.6. At that time, its primary function was streamlining the patch process common in online computer games. Installation and use of Steam was mandatory for *Counter-Strike* 1.6 beta testers, but Steam remained an optional component. 80,000–300,000 gamers tested the system when it was in its beta period. The system and web site choked under the strain of thousands of users simultaneously attempting to play the latest version of Counter-Strike. In 2004, the World Opponent Network was shut down and replaced by Steam.

Around this time, Valve began negotiating contracts with several publishers and independent developers to release their products on Steam. *Rag Doll Kung Fu* and *Darwinia* are two examples, and Canadian publisher Strategy First announced in December 2005 that it would be partnering with Valve for digital distribution of current and future titles. In 2002, Gabe Newell the head of Valve said he was offering mod teams a game engine license and distribution over Steam for $995.

Half-Life 2 release

Profitability

Todd Hollenshead, CEO of id Software, at QuakeCon 2007 presenting the release of all id Software games on Steam

In 2005, the first third-party games began to appear on Steam. Valve also announced that Steam was starting to be profitable, if only due to some highly successful Valve games. Although digital distribution was still no match to retail in terms of sales volume, profit margins for Valve and developers were far bigger on Steam than at retail.

In 2007, big developer-publishers such as id Software, Eidos Interactive and Capcom started to distribute their games on Steam. In May 2007, 13 million accounts had been

created on Steam, and 150 games were for sale on the platform. In October 2007, the release of *The Orange Box*, and the distribution of high-profile games such as *BioShock*, *Call of Duty 4: Modern Warfare*, and *S.T.A.L.K.E.R.: Shadow of Chernobyl*, helped increase Steam's popularity.

Interface update and Mac OS X release

On 23 February 2010, Valve released a public beta for a major update to the Steam client. The update added a new "Library" panel and the rendering engine for the Store and Community pages—including the in-game overlay engine—was changed from Internet Explorer's Trident engine to a WebKit implementation. It was launched on 26 April.

On March 8, 2010, Valve announced that Steam was in development for Mac OS X. Prior to this announcement, it teased the release through several images emailed to Mac community and gaming web sites featuring Valve game characters with Apple logos or featured in parodies of old Macintosh advertisements. In one case, Valve developed a full video homage to the *1984* Apple Macintosh commercial to announce the availability of *Half-Life 2* and its episodes to the service, with some concept images for the video previously used to tease the Mac Steam client.

Prior to Steam's Mac OS X release, Valve released teaser ads that paid homage to famous Apple advertisements, such as the *1984* commercial.

Originally planned for release in April 2010, Steam for Mac OS X was launched worldwide on May 12, 2010, following a successful beta period. In addition to the Steam client, several features were made available to developers to take advantage of a cross-platform Source engine and platform and network capabilities using Steamworks. Through *SteamPlay*, the Mac OS X client will allow players who have already purchased compatible products in the Windows version to re-download the Mac versions at no cost, allowing them to continue to play the game on the other platform; however some third party titles may require the user to purchase again to gain the cross-platform functionality. The Steam Cloud will be cross-platform compatible. Multiplayer games can also be cross-compatible, allowing Windows and Mac players to play with each other.

Upon launch, over 50 games, most supporting the *SteamPlay* feature, were available for the client. As part of the launch, Valve offered both PC and Mac users to download *Portal* for free during the first two weeks of launch. For some weeks after the Mac client launch, Valve will expand the catalog of offerings for the service on a weekly basis, with

each week highlighting a new feature of the service. Valve has released the native Mac OS X; OpenGL versions of *Left 4 Dead*, *Left 4 Dead 2*, *Team Fortress 2*, *Counter-Strike: Source*, *Portal*, and *Half-Life 2* and its episodes following Steam's release. *Portal 2*, due in 2011, will be the first Valve title simultaneously released on both the Windows and Mac versions of Steam.

Linux

Based on Valve's website "There are no plans to create a native Linux Steam Client at this time." Following the announcement of Steam for Mac OS X, Linux benchmark and news website *Phoronix* found Linux-related references in a beta release of the Mac Steam client. This was followed later by the discovery of files for an incomplete Linux client available for download on Valve's own web servers. On May 12, 2010, *The Daily Telegraph* claimed that Valve had confirmed that a Linux client was planned to be released "in the coming months", but gave no other details. This claim was then repeated on other websites as hearsay evidence. Valve has since denied working on a Linux version of Steam.

PlayStation 3

At E3 2010, Gabe Newell, the managing director of Valve, announced that Steamworks would arrive on the PlayStation 3 with *Portal 2*. It will provide automatic updates, community support, downloadable content and other unannounced features.

Steam Translation Server

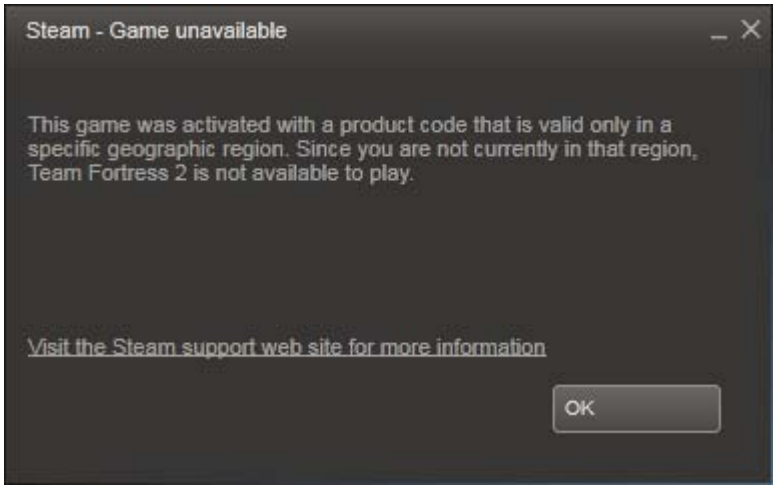
In March 2010 Valve launched the Steam Translation Server, which is a service that allows Steam users to help in the translation of Steam and a selected library of Steam games. The service was undergoing a closed beta until October 2010. Since then, every Steam user has the ability to apply as a translator.

Webmoney support

On 18 December 2010, Valve officially launched support for Webmoney payment system. It can be used from the following countries: Azerbaijan, Belarus, Czech Republic, Estonia, Finland, France, Germany, Israel, Kazakhstan, Kyrgyzstan, Latvia, Lithuania, Poland, Russian Federation, Singapore, Spain, Switzerland, Thailand, Turkey, Ukraine, and the United Arab Emirates.

Criticism

Regional restrictions and pricing



Some games purchased in one region become unplayable if user moves to another.

Steam allows developers and publishers to change prices and restrict game availability depending on the user's location. This can cause some games to cost more than retail prices, despite digital distribution removing the costs of manufacturing, packaging, design, and logistics.

Valve also restricts game registration and playability to the buyer's country of residence. One example of this regional restriction can be seen where Valve uses Steam's authentication to prevent boxed versions of their games sold in Russia and Thailand, which are priced significantly lower than elsewhere, from being used outside those territories.

Steam offers products in three currencies: US Dollar, Euro and Pound Sterling. The currency is selected automatically based on the user's location, and cannot be changed. Steam has been heavily criticized by European users for converting dollar amounts equally between US Dollars and Euros, despite Euros being worth more.

Authentication

It is necessary to authenticate every Steam game online, whether purchased via Steam itself or installed via a retail disc, the first time it is played. After the initial authentication, an offline mode allows games to be run without being connected to your Steam account.

According to the Steam Subscriber Agreement, Steam's availability is not guaranteed and Valve is under no legal obligation to release an update disabling the authentication system in the event that Steam becomes permanently unavailable.

Chapter 13

Sun Cloud and WaveMaker

Sun Cloud

Sun Cloud is an on-demand Cloud computing service operated by Sun Microsystems, a subsidiary of Oracle Corporation. The Sun Cloud Compute Utility provides access to a substantial computing resource over the Internet for US$1 per CPU-hour. It is based on and supports open source technologies such as Solaris 10, Sun Grid Engine, and the Java platform.

Sun Cloud delivers enterprise computing power and resources over the Internet, enabling developers, researchers, scientists and businesses to optimize performance, speed time to results, and accelerate innovation without investment in IT infrastructure.

The Sun Cloud is available worldwide. Since Sunday, March 7, 2010, the network.com web site has been inaccessible.

Suitable applications

A typical application that can run on the Compute Utility fits the following parameters:

- must be self-contained
- runs on the Solaris 10 Operating System (OS)
- is implemented with standard object libraries included with the Solaris 10 OS or user libraries packaged with the executable
    - all executable code must be available on the Compute Utility at time of execution
- runs to completion under control of shell scripts (no requirement for interactive access)
- has a total maximum size of applications and data that does not exceed 10 gigabytes
- can be packaged for upload to Sun Cloud as one or more ZIP files of 300 megabytes or smaller

Resources, jobs and runs

Resources are collections of files that contain the user's data and executable.

Jobs are a Compute Utility concept that define the elements of the unit of work that is submitted to the Sun Cloud Compute Utility. The major elements of a job include the name of the shell script controlling program execution, required arguments to the shell script, and a list of resources that must be in place for the job to run.

A run is a specific instantiation of a Job description submitted to the Sun Cloud Compute Utility. Runs occur when the job is submitted to the Compute Utility for execution.

CPU-hour

For each job one submits and runs on the Cloud, the Sun Cloud CPU usage is aggregated and then rounded up to the nearest whole hour. For example, if a job used 1000 CPUs for one minute, it would be aggregated as 1000 CPU minutes or 16.67 CPU hours. The software rounds this up to 17 hours and the job would be billed as US $17.

The Application Catalog

On March 13, 2007, Sun announced the launch of Application Catalog, an online service that allows developers and ISVs to develop and publish their applications, enabling communities of scientists and academics in life sciences, education, engineering, and other fields to accelerate innovation and complete research projects quickly and less expensively.

The Network.com Application Catalog gives users immediate online access to popular ISV and open source applications through an easy-to-use Web portal with no contractual obligation. Users can upload and run their own applications and create a personal library of favorites or take advantage of the pre-installed and configured applications giving them instant productivity. The portal gives them everything they need to conduct analysis and complete complex computational tasks to help speed scientific discovery and shorten the time to market for new products. They simply select the application, upload their data, and get results fast.

Network.com enables anyone to publish applications to the Application Catalog and take advantage of the powerful Solaris 10-based Cloud platform. Users can publish their own applications to a private library and access them whenever they want; they can also share their applications with others while retaining their data securely in their private space.

Available Applications

Applications available on the Catalog include(by category):

- General - Blender, FDS

- Computer Aided Engineering - Calculix, deal.II, Elmer Solver, Impact, FreeFEM, OFELI
- Life Sciences - BLAST, FASTA, GROMACS, Clustalw, eHITS, T-Coffee, fastDNAml, READSEQ

Examples of types of suitable applications include:

- Bio informatics
- Financial domain applications, like Monte Carlo method, Black-Scholes option pricing models
- Computer Arts, like Fractal landscape generation
- Speech synthesis applications, like Festival
- Scientific applications, like Computer simulation

WaveMaker

	WaveMaker
Original author(s)	WaveMaker
Initial release	December 16, 2007; 3 years ago
Stable release	6.2.5GA / December 14, 2010; 3 months ago
Written in	Java and Javascript
Operating system	Windows, Mac OS X, Linux
Type	Rapid application development
License	Apache License 2.0

WaveMaker (formerly known as ActiveGrid) is an open source software development platform that automates much of the process for creating Java web and cloud applications. WaveMaker provides a visual rapid application development platform and is available as a free open source software download or as a hosted cloud development environment (aka Platform as a Service) running on Amazon EC2.

WaveMaker allows web developers to create Ajax applications. The WaveMaker framework itself integrates Spring, ACEGI, Dojo Toolkit 1.0, authentication, LDAP,

ActiveDirectory and POJOs and their products include Visual Ajax Studio for RIAs development and WaveMaker Rapid Deployment Server for Java application.

Applications generated by WaveMaker community edition are licensed under the Apache license. A commercial license is also available with additional enterprise security capabilities.

WaveMaker applications are created using the WaveMaker studio, a WYSIWYG development studio that runs in a browser and enables drag and drop development of web applications following a Model-view-controller architecture. WaveMaker supports RAD for the web, similar to what products like MS Access, PowerBuilder and Lotus Notes provided for client server computing.

WaveMaker applications run in a standard Java server based on Tomcat, the Dojo Toolkit, Spring and Hibernate. Currently, it is supported on Microsoft Windows, Linux and Macintosh.

WaveMaker has also released a cloud computing version of its 4GL that runs on Amazon EC2.

As an example of the level of complexity of applications that can be built using a WYSIWYG development approach for Ajax applications, the WaveMaker Studio was built using WaveMaker. WaveMaker is meant for use by web developers who prefer visual tools.

Features

- Automatic generation of Hibernate mapping, Hibernate queries from database schema import.
- Automatic creation of Enterprise Data Widgets based on schema import. Each widget can display data from a database table as a grid or edit form. Edit form implements create, update, delete functions automatically.
- Visual, drag & drop assembly of web applications.
- WYSIWYG Ajax development studio runs in a browser.
- Developer sees live application data within the studio (LiveLayout).
- Simplified (one-touch) deployment to Tomcat, Websphere, Weblogic, JBoss.
- Data schema aware forms to edit, update, delete data (LiveForms).
- Mashup Tool for assembling web applications based on SOAP, REST and RSS web services, Java Services and databases.
- Leverage existing CSS, HTML and Java.
- Deploys a standard Java .war file.
- Browser-based WaveMaker studio can be bundled by ISVs with their web-based application to enable user customizaton.