

COMPUTING SERVICES IN THE CLOUD: A TUTORIAL

Harry Katzan, Jr.
Savannah State University

ABSTRACT

Cloud computing is a modality for providing computer facilities via the Internet. The cloud-computing concept incorporates single-function applications, such as those available as office suites, and the execution of comprehensive enterprise applications pieced together from components residing in varying Internet locations. The application software with cloud computing is useful for connecting people and organizations in various combinations across the Web and supports mobile computing. This paper gives a tutorial on cloud computing.

FORWARD

The discipline of service science serves as the basis of modern computer and Internet technology encompassing the subjects of Web services, service-oriented architecture, and, most recently, cloud computing. Most aspects of modern information systems are derived from service science, as well as are the pragmatic sides of business and economic theory. This paper seeks to investigate the underlying principles that govern the exchangeable value of cloud services. Throughout, we will attempt to show the real value of service, the different parts of which a cloud service is constructed, and the forces that govern the dynamics of service value.

One of the defining characteristics of cloud computing is the transfer of control from the client domain to the cloud service provider. Accordingly, it is particularly important that client requirements are delineated and analyzed, because without a clear understanding of exactly what client needs are in cloud computing, what constitutes a cloud service, what differentiates one form of cloud computing from another, and how cloud services operate and interoperate, continuous improvement will be a never-ending process of trial and error.

CLOUD COMPUTING CONCEPTS

Cloud computing is a means of accessing computer facilities via the Internet, where the adjective “cloud” reflects the diagrammatic use of a cloud as a metaphor for the Internet. Most of us have been using cloud-computing facilities in one form or another for years through ordinary email and the World Wide Web. Recently, the term has come to reflect the use of software and the running of computer applications via the Internet where the computer infrastructure and software are not “on premises.” Cloud computing, as a form of service provisioning, has given rise to several related concepts, such as mesh computing, cloud platforms, and software plus service.

A proper, but not necessarily definitive, conceptualization of cloud computing is to use office-class applications via your web browser over the Internet instead of having those applications reside on your “on premises” computer. In this instance, the service provider supplies the network access, security, application software, and data storage from a data center located somewhere on the Internet and implemented as a form of server farm with the requisite infrastructure. A service would have ubiquitous access through a web browser. In general, the cloud computing concept is not limited to single-function applications, such as those available with typical office suites, but could include comprehensive enterprise applications pieced together from components residing in varying Internet locations.

Every year, businesses spend millions of dollars on their IT infrastructure consisting of hardware, system software, applications, networks, people, and other organizational assets. With “on demand” computing, they can plug into the wall, figuratively speaking, and only pay for the IT services they use. The general concept is called *utility computing* that is accessed as most public utilities. When appropriate, a service utility is a viable option for obtaining computing services, the essence of which is in the packaging of computer services as a metered facility without up-front costs for IT infrastructure. In the current view of things, a services utility is network based and is dependant upon the Internet as a transport mechanism. In recent years, computing has become the operational medium for business, government, education, and a part of everyday life for most people, and as with electric utilities, computing utilities have evolved from being a luxury to an everyday necessity.

CLOUD SERVICE CHARACTERISTICS

Cloud service utilities are characterized by four key factors: necessity, reliability, usability, and scalability. *Necessity* refers to the idea that a preponderance of users depend on the utility to satisfy everyday needs. *Reliability* refers to the expectation that the utility will be available when the user requires it. *Usability* refers to the requirement that the utility is easy and convenient to use – regardless of the complexity of the underlying infrastructure. *Scalability* refers to the fact that the utility has sufficient capacity to allow the users to experience the benefits of an expandable utility that provides economy of scale. Certainly, modern Internet facilities for search operations that engage thousands of servers satisfy these characteristics.

The notion of “paying for what one uses” is a compelling argument for using the cloud for special or all computing needs. The proof of the pudding may be in the details. The key question is whether the service should be based on a metered model or a subscription model. With the *metered model*, the usage is easily measured, monitored, and verified and lends itself to managerial control on the part of the user. In addition, metering can be applied to differing levels of service. With the *subscription model*, usage is difficult to control and monitor and its adoption is favored by managers more concerned with convenience than with resource control.

The difference between application services and multi-tenant services may very well be the deciding factor in determining whether metered or subscriber service is the way to go. With *multi-tenant service*, several clients may share the same software with separate data – as in the case of office processing. With *application service*, the service provider supplies one instance of the software per client, thereby lending itself to a form of metered service. In the latter case, the notion of a client should be regarded as an environment comprised of several users.

HOSTING AND VIRTUALIZATION

A common example of utility computing is *hosting* wherein an application service provides “off premises” computer services on a subscription or pay-as-you-go basis. The practice is prevalent among relatively small software developers that require expensive computer facilities. A service provider usually supplies requisite services on a time-sharing basis through communications facilities, and the service provided is the access to and utilization of a computing platform comprised of a computer system, an operating system, and necessary utility facilities. This is the origin of the *Platform as a Service* (PaaS) concept, often sustained through virtualization.

Virtualization refers to the provisioning of a “not real but virtual” computing environment created through a software facility known as a *hypervisor* with the capability of managing several diverse computing platforms, executing concurrently, so that the client is given the operational advantage and illusion of having a unique copy of the selected platform. The hypervisor controls the underlying

computer hardware and software and passes control to a specific client instance on a demand basis. [Kat86]

BUSINESS ASPECTS OF CLOUD SERVICES

The long tail [Cho06] is a conceptualization of the unique business opportunities available through Internet access, exemplified by online book sellers and software services. A brick-and-mortar bookseller has a limited amount of self space and typically stocks only the most popular books. Online booksellers do not have the same limitation and are able to take advantage of the long tail, as suggested by Figure 1, to provide opportunities not available otherwise. The long-tail phenomenon also applies to line-of-business software, as depicted in Figure 2, and consumer-oriented services to provide a level of economy of scale not otherwise available with “on premises” software and the requisite computing platforms. The long-tail perspective provides a basis for the monetization of cloud computing. The defining characteristic of cloud computing is that services are accessible through a web browser. In general, the cloud computing concept is not limited to single-function applications, such as those available as office suites, but could include comprehensive enterprise applications pieced together from components residing in varying Internet locations. Because the application software with cloud computing is not executed on a local computer, it is useful for connecting people and organizations in various combinations across the Web and supporting mobile computing. Cloud computing should not be confused with outsourcing. With outsourcing, an existing function is moved out of the department, enterprise, or geographic jurisdiction. With cloud computing service, the home of an application originates in the cloud.

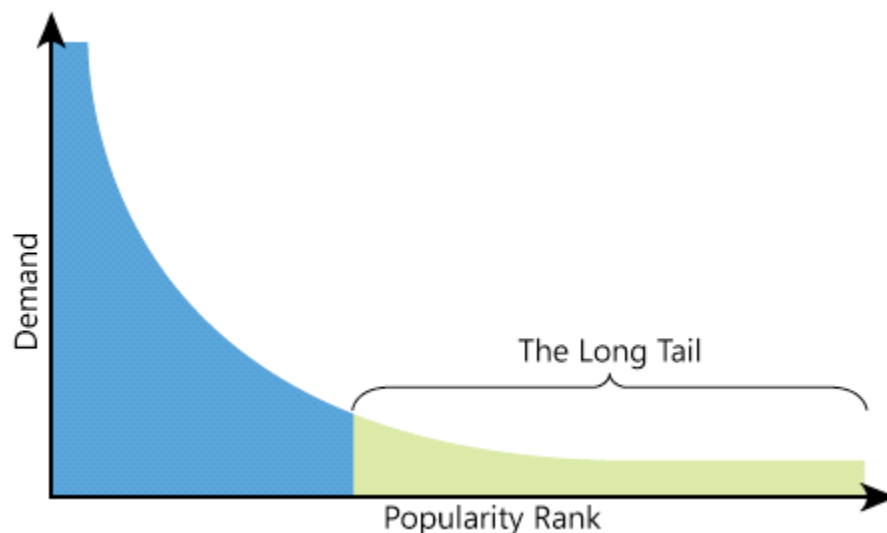


Figure 1. The Long-tail concept. [Cho06, p. 8]

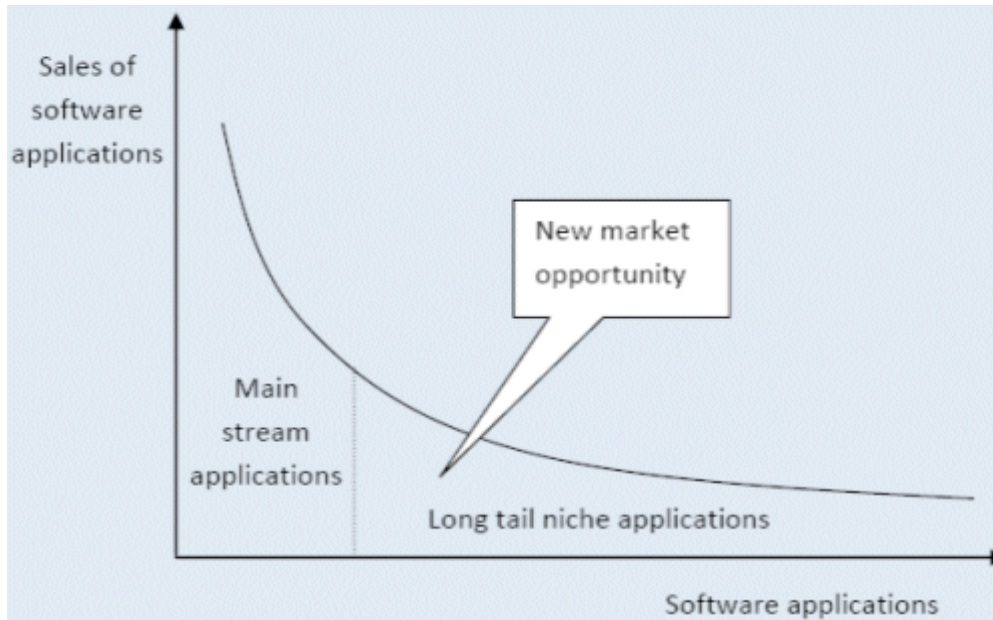


Figure 2 Long-tail software opportunities. [Cho08, p. 2]

It is difficult to state the difference between cloud computing and utility computing, because they both appear to refer to the same phenomena. Digging a little deeper, however, it would appear that utility computing is more of a business concept – perhaps a business model – providing “pay for what you get” services, where the operational framework could be traditional batch processing, local networks, enterprise networks, or the Internet. In fact, utility computing has the flavor of a “spin off” of a non-core service to another organizational entity – either internally or externally. In general, utility computing could use the Internet, but that is not a defining characteristic.

Cloud computing, on the other hand, is by definition an Internet-based facility – hence the “cloud” metaphor referring to the usual depiction of the Internet. With the Internet providing the accessibility component, service clients do not typically own the hardware and software infrastructure. So the chief advantage of cloud computing from the client’s perspective is the availability of software, storage, and computing resources without up-front infrastructure costs.

CLOUD TERMINOLOGY

As with other forms of service, cloud computing has its own terminology, reflecting the underlying concepts and components, such as the following: [Wik08]

Cloud application – the software service being offered.

Cloud client – the service participant using the cloud functionality.

Cloud platform – the computer infrastructure that supports the cloud computing service.

Cloud service – the application facilities provided by the cloud environment.

Cloud storage – the online storage for files and databases supported by the cloud platform.

Cloud architecture – the overall design of the cloud computing environment.

Cloud provider – the enterprise that owns and operates the cloud computing service.

Collectively, the cloud concept engenders two related topics: Software as a Service (SaaS) and Platform as a Service (PaaS). With SaaS, the burden of software installation, maintenance, and support is practically eliminated from the client domain. From the vendor’s viewpoint, SaaS establishes an ongoing revenue stream for the software developer. PaaS provisioning reflects a business model wherein one

enterprise supplies PaaS hosting to one or more SaaS enterprises. In the provisioning package, the PaaS provider would supply advanced security measures and other necessary infrastructure facilities never actually seen, per se, by the cloud client. In this scenario, the SaaS enterprise is then solely responsible for cloud software development.

BUSINESS AND CONSUMER SERVICES

Chong and Carraro [Cho06] define software as a service (SaaS) as software deployed as a hosted service and accessed over the Internet. The key features of SaaS are where the programs reside and how they are accessed. The two kinds of software in this category are business software and consumer software. Business software provides business services and emphasizes business solutions, such as CRM, SCM, ERP, and human resources. Consumer software provides publicly oriented personal solutions, such as office applications and are often provided at no cost – that is, in their cloud versions.

With business services, the most important consideration is whether the process is executed in-house or as a cloud service. When the process is handled in-house, total control over the operation is obtained along with limited opportunity for achieving economy-of-scale. As processes are distributed outward on the cloud, control is decreased but opportunities for achieving economy-of-scale are increased. The considerations are different with consumer services. Pure service, as with office applications, provides practically no control over the application to the client and a reasonably high-level of economy-of-scale to the provider. In many cases, consumer services are advertising-supported and are complimentary to the client through advertising. In addition to the metered and subscription models, the advertising-supported model is another means of monetizing cloud computing.

Business applications that reside “on premises” are governed by the traditional considerations of application acquisition and deployment. If an application resides on and is deployed from the cloud, then two options exist:

- (1) Build the software yourself (or have it built for you) and run it on the cloud as a hosted service – perhaps using a cloud platform.
- (2) Obtain the application software from an independent software vendor (ISV) and run it on the cloud in a standard or modified mode.

In the former case, all users access the same version of the software. In the latter case, a client gets a customized version achieved with a separate code base, or its equivalent, configuration options, or operational metadata. The subject of business services is covered in more detail in a subsequent section.

The primary advantage of a cloud consumer service is that it is typically free to the client, as well as being accessible from any location via the Internet, and it yields advertising-supported revenue for the provider. Consumer services have a near-zero marginal cost of distribution to clients, because of the long tail, and requires only a fraction of the number of clients to respond to advertising. This is the well-known *Freemium Business Model* [And04], characterized as follows: In the free sample product model, you give away 1% of your product to sell the additional 99%, whereas in the freemium model, you give away 99% to sell 1%. Because of the scale of the Internet with millions of users, you can reach a large market, so that the 1% is a huge amount.

Software plus Service (S+S) refers to a user-centric approach to service deployment by combining “on premises” computing (fat client) with enhanced services on the cloud. The enhanced services combine advanced functionality with the capability to scale up to meet peak computing demands for both business and consumer services. A related feature of S+S involves the distribution of service pack software

updates for both system and application software and the provisioning of automatic software downloading.

Clearly, the business model for the deployment of both SaaS and S+S changes with the adoption of cloud computing. The ownership of software shifts from the client to the provider, along with the responsibility for the technology infrastructure and its management. [Cho06] The marketing targets for SaaS and S+S clients are service consumers and small to medium-sized business, and economy of scale is achieved through specialization and the development of cloud platforms.

CLOUD SERVICE ARCHITECTURE

A comprehensive SaaS application structure includes a continuum of architectural levels, based on the capability of handling multiple clients and software configurability. Four levels are identified. The number of levels in any specific operational environment is based on the cloud platform and its characteristics.

Level One. At the first level, the users within a client domain address a single instance of an application running on a server. Each client/instance is totally independent of other client/instances running on the same server. This is the traditional hosted service operating in the cloud. Each software instance is individually customized for each client.

Level Two. At the second level, the server hosts a separate instance of the software for each client, but the instance is a configurable version of the same code base, reducing maintenance costs and contributing to increased economy-of-scale.

Level Three. At the third architectural level, the vendor runs a sole instance that is shared by multiple clients. The feature set for each client is determined by configurable metadata, and authorization/security policies insure the separation of user data.

Level Four. At the fourth level, the same “level three” instances are run on a server farm with fabric for load balancing.

The choice among architectural levels is determined by the provider/client’s business, architectural, and operational models.

CLOUD PLATFORMS

A *cloud platform* is an application service provider that runs in the cloud. More specifically, a cloud platform provides services to applications in the same manner that “software as a service” programs provide services to clients using the cloud as a transport medium. A cloud platform resides in a cloud data center and exists as a powerful computing facility, a storage system, an advanced operating system, support software, and the necessary fabric to sustain a server farm and scale up to support millions of Internet clients. A cloud platform is as much about operating in the cloud, as it is about developing applications for the cloud.

A cloud platform provides the facility for an application developer to create applications that run in the cloud or use cloud platform services that are available from the cloud. Chappell [Cha08a, Cha08b] lists three kinds of cloud services: SaaS user services, on-premises application development services (attached services), and cloud application development services. An *SaaS application* runs entirely in the cloud and is accessible through the Internet from an on-premises browser. *Attached services* provide functionality through the cloud to support service-oriented architecture (SOA) type component development that runs

on-premises. *Cloud application development services* support the development of applications that typically interact while running in the cloud and on-premises.

A cloud platform can be conceptualized as being comprised of three complementary groups of services: foundations, infrastructure services, and application services. The *foundation* refers to the operating system, storage system, file system, and database system. *Infrastructure services* include authorization/authentication/security facilities, integration between infrastructure and application services, and online storage facilities. *Application services* refer to ordinary business services that expose “functional” services as SOA components.

Cloud platforms are a lot like enterprise-level platforms, except that they are designed to scale up to support Internet-level operations.

APPLICATION SERVICES

“*Venus* is for application services and *Mars* is for infrastructure services.” Application services are designed to be used by people, and infrastructure services are designed to be used by applications. [Cha08, p. 11] The basic idea of cloud platforms is that SaaS applications will be created by developers to provide services used by people, and SaaS applications will use infrastructure services.

Software plus service (S+S) is an in-between point in the cloud service continuum, falling between the pure-play user-centric set of services and the large-scale enterprise application systems in which on-premises and cloud software interact to support comprehensive business services. In the S+S hierarchy, the cloud platform should consist of building block, attached, and finished services to complement application services, mentioned previously, and to support a flexible set of operational scenarios that include PCs, the Web, mobile devices, on-premises servers, and cloud-based services. [Fol08]

CLOUD COMPUTING REQUIREMENTS

In addition to the prototypical definitions, given earlier, cloud computing has been characterized in some creative ways, somewhat as follows: (1) It is a host in the cloud; (2) It is a hub in the tub; [Apparently, use of the word “tub” refers to the Internet.] (3) It is a means of monetizing Internet services; and (4) It is a complimentary means of achieving workflow and online collaboration. [Referring to the *Freemium Business Model*.] The list could go on-and-on. This section gives a set of basic client requirements for cloud computing from an “end user” perspective. The ideas are summarized in Table 1.

Class I. Single Tenant Model

The functionality inherent in this class is to host a software application and deploy it as a single instance for each user. The application software would use the computational and operating system facilities in the cloud, as well as services provided through the cloud, such as database management and communications support. This is a single identity environment, wherein each user supplies authentication and authorization information to the application via the operating system. If another user requires access to the same application, then a separate instance is deployed within the cloud environment. The cloud client is the individual user. In the latter case, variations in the application software are achieved through configuration options and metadata. Storage for user data is in the cloud using the fabric of the cloud service center. ISV monetization for standard applications is provided as metered, subscription, or advertising-supported services, and extra functions can be supported on a “fee for use” basis.

Class II. Client Based Model

The functionality inherent in this class is to host a software application and deploy it as a single instance for each client. The distinction is that a client may have several users accessing the same application, most probably at the same time, such that data can be exchanged between users. The model is commonly referred to as the “multiple tenant” model. Software is configured on a client basis and security is supplied by the cloud operating system. Storage for user data is provided in the cloud, logically segmented by client. Individual users may require extra cloud functionality that are supplied on a subscription or metered bases. Multi tenant operations require exceedingly complex cloud operating system features that are beyond the scope and intention of this paper.

Class III. Attached Service Model

In this model, software applications are hosted “on premises,” and extra functions are available as “attached services” from the cloud. This is a form of service-oriented architecture, wherein the extra functions are executed in the cloud.

Class IV. Cloud Application Platform

In this model, a cloud application platform is available on the cloud and can be viewed as an “operating system in the cloud,” comprised of foundation services, infrastructure services, and application services. [Cha08a] The cloud platform should contain computational, traditional operating system, storage, identity, packaged applications, and custom software applications.

Class V. Mesh Computing

A final class is termed a *mesh*, intended to connect a user’s PC, mobile device, and other cloud services via the Internet. The mesh software functions as a control hub providing unified services on a demand basis.

Clearly, the above list is a start in the delineation of end-user requirements for a cloud computing environment.

<i>Class</i>	<i>Name</i>	<i>Basis</i>	<i>Computation</i>	<i>Monetization</i>
I	Single instance	Single tenant	Cloud	A, B, C
II	Client based	Multiple tenant	Cloud	A, B
III	Attached services	Services	On premise	A
IV	Cloud platform	Services	Cloud	A, B
V	Mesh	Connection/hub	Cloud	C

(Legend: A-Metered, B-Subscription, C-Advertising-supported)

Table 1. Characterization of Cloud Computing Requirements.

FURTHER RESEARCH

Cloud computing has evolved into a huge research topic with each major player in the IT provisioning group supplying its own version of exactly what the subject matter should incorporate. The above materials are an attempt at finding a middle ground and providing a basis for further research. Two very

important areas have not been covered: authorization/authentication/security and cloud databases. Both are significant for exchanging data between applications within the cloud.

REFERENCES

- [And04] Anderson, C., "The Long Tail," *Wired Blog Network*, 2004.
- [Cha08a] Chappell, D., "A Short Introduction to Cloud Platforms," *Microsoft Corporation*, August 2008.
- [Cha08b] Chappell, D., "Introducing the Azure Services Platform," *Microsoft Corporation*, October 2008.
- [Cho06] Chong, F. and G. Carraro, "Architecture Strategies for Catching the Long Tail," *Microsoft Corporation*, April 2006.
- [Cho08] Chong, F., "Application Marketplaces and the Money Trail," *Microsoft Corporation*, March 2008.
- [Fol08] Foley, M., *Microsoft 2.0*, Indianapolis: Wiley Publishing, Inc., 2008.
- [Kat86] Katzan, H., *Operating Systems: A Pragmatic Approach (2e)*. Hoboken, NJ: John Wiley and Sons, 1986.
- [Kat08] Katzan, H., *Service Science: Concepts, Technology, Management*, New York: iUniverse, Inc., 2008.
- [Mil08] Miller, M., *Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online*, Indianapolis: Que Publishing, 2008.
- [Rap04] Rappa, M., "The utility business model and the future of computing services," *IBM Systems Journal*, Volume 43, Number 1, (2004).
- [Wik08] Wikipedia, "Software as a Service," www.wikipedia.com, 2008.