Cloud Computing Economics: Democratization And Monetization Of Services

Harry Katzan, Jr., Savannah State University, USA

ABSTRACT

Cloud computing is a modality for providing computer services via the Internet, by incorporating ubiquitous access through a web browser for the execution of single-function applications, such as those available as office suites, and comprehensive enterprise line-of-business applications pieced together from components residing in varying Internet locations. This paper covers the democratization and monetization of software services and uses cloud computing as the primary delivery vehicle. Cloud computing represents a contextual shift in how computers are provisioned and accessed. There is opportunity and value in cloud computing for providers, ISVs, and customers. The subject is covered from an economic perspective for each of the groups.

Keywords: Service science, cloud computing, cloud economics, service democratization, service monetization.

INTRODUCTION

The discipline of modern information systems is based on service science, and within that domain, this paper seeks to analyze the underlying principles that govern the exchangeable value of cloud computer services. Throughout, we will attempt to show the real value of cloud service, the different parts of which a cloud service is constructed, and the forces that govern the dynamics of service value. In fact, if one replaces the concept of labor with that of service, the principles of service science can be derived from the essential work of Adam Smith [20], most pointedly in his notions of "value in use" and "value in exchange." One of the defining characteristics of cloud computing is the transfer of control from the client domain to the cloud service provider. Another is that the client benefits from economy of scale on the part of the provider. There are differences of opinion on exactly what cloud computing is, and this paper attempts to mediate the various economic positions.

BASIC 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.

Conceptualization

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 or mobile device. 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.

Utility Computing

Every year organizations 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 [12].

Business Aspects of Cloud Services

The long tail [8] 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 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 democratization and monetization of cloud computing. 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.

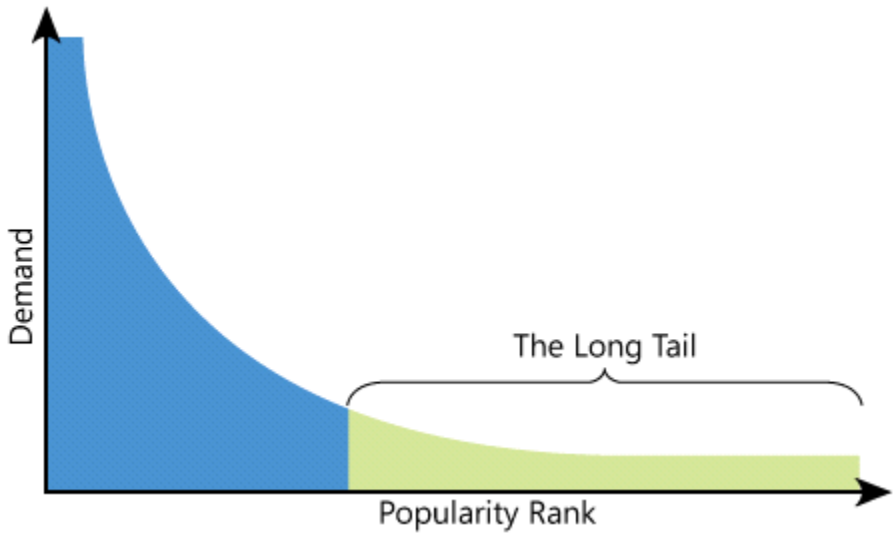

Figure 1. The long-tail concept. (Chong and Carraro [8], p. 8)

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, with the Internet providing the accessibility component. Service clients do not typically own the hardware and software infrastructure, so a major advantage of cloud computing from the client's perspective is the availability of software, storage, and computing resources without up-front infrastructure costs.

BUSINESS AND CONSUMER SERVICES

Chong and Carraro [8] 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 personal solutions, such as office applications, that are often available at no cost 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 require only a fraction of the number of clients to respond to advertising. This is the well-known *Freemium Business Model* [1], 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 [8]. The marketing targets for SaaS and S+S clients are service consumers and small to medium-sized businesses, 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 lead 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 operating system 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. Both use the Internet 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; and in so doing, the application developer uses services that are available from the cloud. Chappell [6, 7] 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 or mobile device. *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 Internet-level operations supporting millions of clients.

Application Services

Application services are designed to be used by people, and infrastructure services are designed to be used by applications. The basic idea of cloud platforms is that SaaS applications will be created by developers to provide services used by people, and will use cloud 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, Web access, mobile devices, on-premises servers, and cloud-based services [11].

DEMOCRATIZATION OF CLOUD SERVICES

The origin of democratization from a cloud computing perspective lies in the fact that people can have better access to software, computing facilities, and data through network effects, by employing cloud computing facilities rather than strictly using "on premises" software and hosting. Essentially this means a person or an organization can obtain results via the Internet previously available only to the privileged few. The topics of business intelligence (BI) and knowledge management (KM) represent good examples. In banking, for example, a snapshot of a customer's profile from various data sources would heretofore require the development of special computer programs. With cloud computing and democratization, that same information could be obtained dynamically and presented through an adaptable dashboard, enhancing collaborative efforts and business agility.

Overview

Cloud service democratization refers to either of three distinct but related forms. In the first instance, known as the availability model, it is the process of making a premium cloud service available for general use, rather than through proprietary services. In the second instance, known as the sharing model, it is the capability of sharing data, infrastructure, and storage that would not be otherwise accessible with on-premises facilities. The final instance, known as the voting model, is the phenomena of giving power to the end user by providing access to facilities that are implicitly more preferable than other cloud resources by virtue of the fact that they are used or referenced more frequently by other end users.

Availability Model

The *availability model* reflects the ability of having access to information, software, and computing resource infrastructure without necessarily having to own it. In many cases, the cost and time elements are too great for many organizations, because the up-front costs and time to develop the information, software, and on-premises resources is simply too great for many potential clients and ISVs. Figure 2 represents the availability situation for business users. The cost of providing computing infrastructure and software by traditional ISVs is such that it is affordable only to larger businesses. This situation leaves out the long tail of small to medium-sized businesses that could benefit from the solution if the cost were lower.

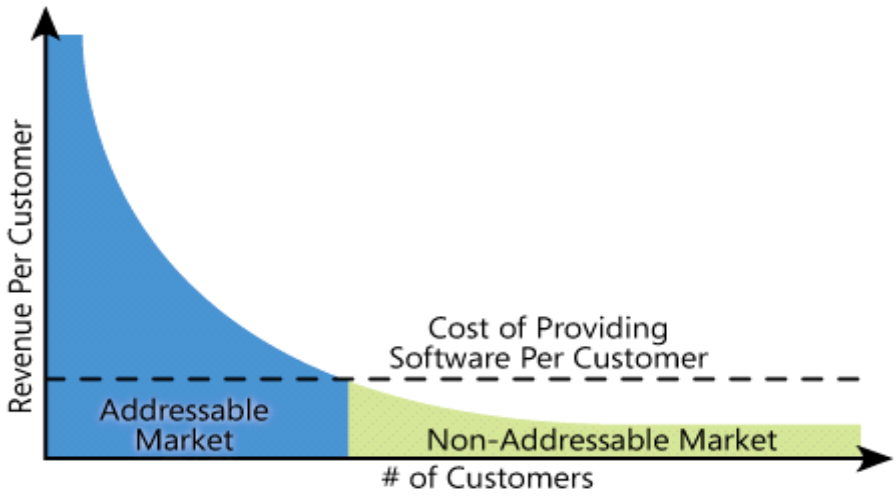

Figure 2. The long-tail market concept for traditional ISVs. (Chong and Carraro [8], p. 29)

By lowering the cost of service provisioning by utilizing multi-tenancy technology and taking advantage of economy of scale achieved through multiple clients from the cloud, business software services are available to the long-tail market, as suggested in Figure 3.

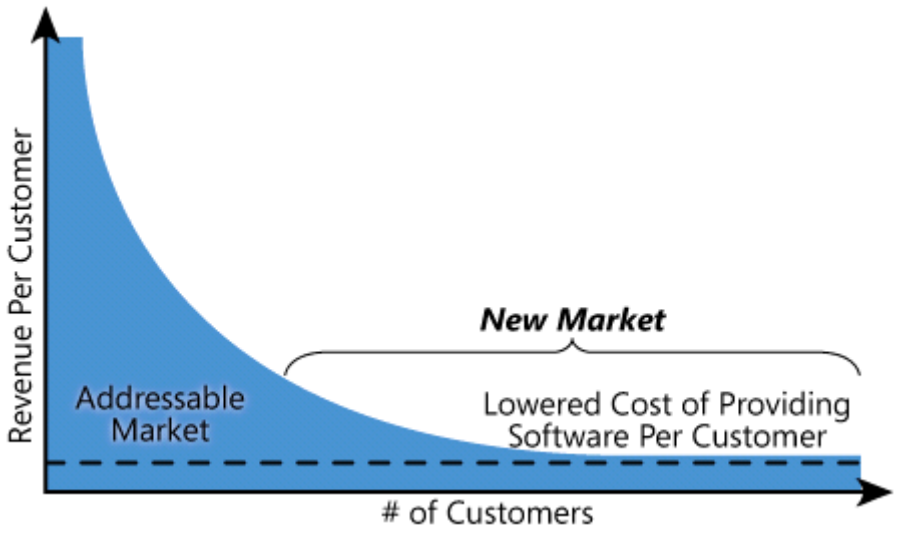

Figure 3. Long-tail market opportunities through the cloud and software services. (Chong and Carraro [8], p. 30)

Sharing Model

The *sharing model* refers to the fact that three major classes of technical resources are available on a shared basis through the availability of cloud platforms: data and information, infrastructure services, and data storage facilities. In the case of *data and information*, content can be shared between users from the same client, between clients, and between platforms from the same user. Effectively, more information is available to more users.

Infrastructure sharing is a major category and a major cost to an IT shop. It includes software, hardware, and security services. With software, comprehensive facilities are available at a lower price, because the cost is shared among thousands of users. With hardware, the end user need not have to plan for peak periods and growth, since elasticity is designed into the architecture of cloud platforms. This is the "scalability" characteristic of utility services. With security, federated security systems are shared among users and clients enabling mobility between diverse computing platforms.

Data storage sharing refers to the common habitation of data on a cloud platform by several clients. Storage multiplicity, commonly available on cloud platforms, reduces organizational concerns in the general area of disaster planning.

Voting Model

The value of certain services, such as web auctions (e.g., e-Bay), user-supported encyclopedias (e.g., Wiki), and modern search engines (e.g., Google), is derived from the fact that many people use them. The power of such facilities lies in the fact that each individual user votes by choosing to use the respective service. With a web auction, it is the interchange between users that gives the facility its democratic power. With an updatable online encyclopedia, an individual end user has the power of updating an entry. This option allows the informational resource to evolve as more people use it. With a search engine, such as Google, it is the method of page ranking,

wherein the number of page references to an object page gives its score, and allows a universe of users to democratize a page.

Collectively, cloud service democratization essentially enables the delivery of computing service to more clients at a reduced cost. This topic is introduced in the next section.

MONETIZATION OF CLOUD SERVICES

The basis for the monetization of cloud computing is software-as-a-service (SaaS), commonly regarded as software hosted service from a cloud platform. The varieties of software-as-a-service are non-configurable, single-tenant, and multi-tenant. With *non-configurable SaaS*, the service provider delivers a unique set of features that are hosted in the cloud through a cloud platform and Internet accessibility to the client. With *single-tenant SaaS*, the client has isolated access to a common set of features, perhaps configured in a distinctive way. With *multi-tenant SaaS*, the provider hosts common program logic and unique data elements for multiple clients on scalable infrastructure resources supported via a cloud platform.

Application Domain

The application domain of cloud computing, introduced earlier under "Cloud Service Characteristics," is usually divided into two diverse groups: business services and consumer services. The emphasis with service monetization is on business services, since the client and the independent software vendor (ISV) would have some control over the business model, whereas with consumer services, the freemium model monopolizes the service landscape. In either case, the major considerations involved with the development of cloud applications are multi-customization, extensibility, isolation, and cloud storage, known as data scaling.

Business Model

The business model for cloud computing reflects how service providers can increase revenue and how clients can reduce operational costs of services over on-premises facilities. There are two areas that can be addressed: the application architecture and the operational structure. From a monetization viewpoint, there are two options for application architecture: common features and unique features. For operational structure, the options are single-tenant and multi-tenant. The various options are regarded as service drivers and are summarized in Figure 4.

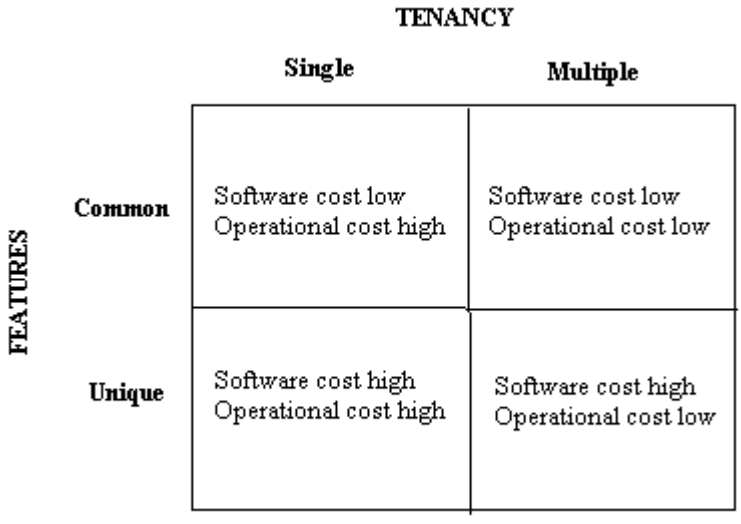

Figure 4. Service Drivers for Cloud Service Monetization.

The salient features of the cloud computing business model are summarized as follows:

- The ownership of the software is transferred to the cloud service provider.
- The responsibility for hardware, application software, storage facilities, and professional services resides with the provider.
- Systems software is available from a trusted vendor for supporting cloud services.
- Data centers are available for sustaining the operational structure and supporting the requisite fabric needed to utilize service farms.

Accordingly, the business model provides the economy of scale needed to target the long tail by providers and reducing up-front and operational costs for the client.

The SaaS provider with cloud computing will characteristically experience high up-front costs for infrastructure and software development. This is covered later. The SaaS client will have to give up a certain level of control to benefit from the economy-of-scale supplied by the provider. There are lingering questions over "who owns the software," "who owns the data," and information security.

Client Perspective

The client perspective with SaaS, and essentially all of cloud computing, involves features provided by application and systems software, service-level agreements, and price. How the software is designed and deployed in the cloud is a provider consideration. Any technical decisions made by the provider as to adopting non-configurable, single-tenant, or multi-tenant services are reflected in the user's price and service-level agreement.

It is now possible to be specific about service monetization for software utilization, and the various options are reflected in four categories: perpetual license, subscription, transaction based, and ad funded [5].

Perpetual license refers to an "up front" payment for service, and unlimited access for an unlimited time. The is an attractive form for seller and buyer, since customer churn is reduced and utilization need not be a monetization concern. Effectively, the financial considerations and technical issues are essentially separated. When considering cloud versus on-premises hosting, the cards are definitely in favor of the client, even though the provider can always attach monetization schemes such as maintenance fees under the guise of software upgrades.

Subscription, as a form of cloud service monetization, can be conceptualized as a time-based perpetual license, often applied to multiple users. In the modern view of software acquisition, an individual buyer purchases only the right to use the software. When applied to organizations, the subscription method provides a reduced rate to multiple users along with a time-dependent renewal fee.

Transaction based monetization requires a close association between hosting software and financial billing. This form of pricing allows the provider with a means of recouping its up-front infrastructure costs, while permitting the client to benefit from economy-of-scale. This form of monetization requires a level of trust between client and provider and experiences certain "no repudiation" issues. As with telephone service, monthly plans are often preferable to pay-as-you-go plans.

Ad funded monetization schemes appear to be most popular with consumer services as described with the freemium mode offered previously. The software service is provided for free and a sponsor pays the cost in return for the consumer's attention. A related form is *Software plus Service* (S+S), wherein the provider supplies access to enhanced features at a fee to support the basic consumer service.

Business Perspective

An element of software is essentially a collection of features – functions that perform a computational task. The set of features in a product or service support the selling aspects of that item. The cost of providing software

services in a cloud environment are substantial and involve design, implementation, testing, marketing, and support activities. In short, complex software has a higher cost over simple software. When cloud software is architected for multi-tenancy to achieve sharing and economy-of-scale for the client, the costs are dramatically higher. Cloud software provisioning is a trade-off between features and tenancy. Each of the items, i.e., features and tenancy, is actually a continuum, even though it is customarily treated as "lower" and "higher" to simplify the analysis and understanding. The scenario is inherent in the graphs in Figure 4.

Cost per tenant is the cost of delivering an element of software to one client and covers any requisite expenses needed for the associated service. Multi-tenant architecture maximizes sharing between users and increases the total revenue and economy of scale for the client; however, it increases development costs.

Cost per feature is the cost of implementing a specific functionality. Simpler features cost less to develop and maintain. Adding multi-tenancy increases the cost of a feature. The tenancy/feature conundrum is summarized as follows: "… multi-tenancy incurs a higher cost per feature, but lower cost per tenant while isolation has lower cost per feature but higher cost per tenant."[5] Figure 5 summarizes in graphical form the advantages of each approach over time. As the tenant base grows (t3 in the diagram), multi-tenancy has good monetization for the provider.

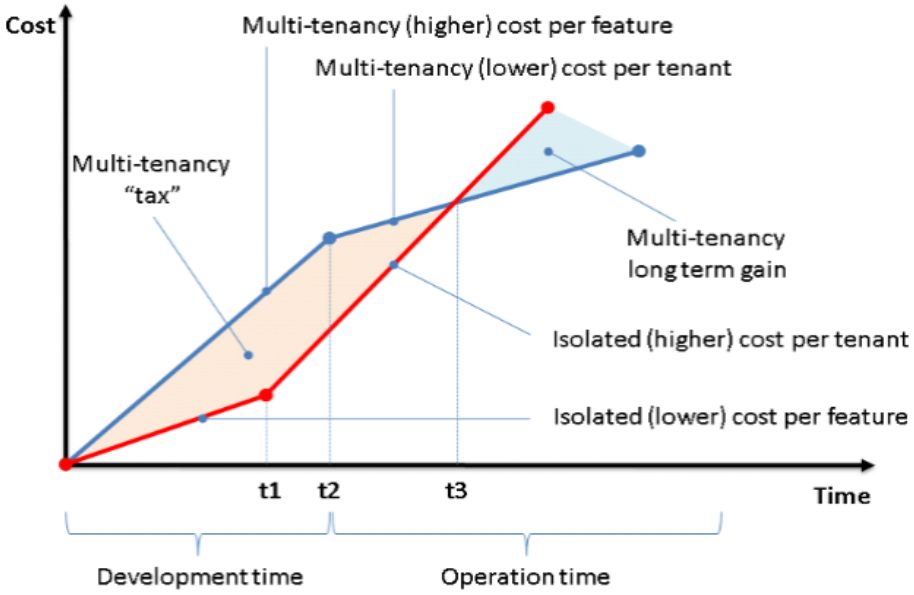

Figure 5. Cost per Feature vs. Cost per Tenant. (Carraro [5], p. 2)

A promising approach to achieving economy-of-scale is to use virtualization at the cloud platform operating system level instead of designing an application for multi-tenancy.

QUICK SUMMARY

1. The disciplines of information systems and cloud computing are based on service science.
2. The principles of service science can be derived from the essential work of Adam Smith in the areas of "value in use" and "value in exchange."
3. Cloud computing is a means of accessing computer facilities via the Internet. (The *cloud* is a metaphor for the Internet.)
4. Cloud service utilities are characterized by four key factors: necessity, reliability, usability, and scalability.

5. The long tail is a conceptualization of business opportunities available through Internet access.
6. Software as a service (SaaS) is software deployed as a hosted service and accessed over the Internet.
7. For the cloud client, business service is a balance between control and economy of scale.
8. A cloud platform is based on an operating system that runs in the cloud and provides an infrastructure for software development and deployment.
9. Cloud service *democratization* refers to ubiquitous information and computing availability, information sharing, and the exercise of user preference in supplying information service.
10. Cloud service *monetization* refers to gaining financial benefit through cloud access for both provider and client.

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 through the cloud.

ACKLNOWLEDGEMENT

Thanks to William Dowling and Margaret Katzan for reading this paper.

REFERENCES

1. Anderson, C., "The Long Tail," *Wired Blog Network*, (2004).
2. "Cloud Computing: The Evolution of Software-as-a-Science," Arizona State University W.P. Carey School of Business, June 4, 2008, knowledge.wpcarey.asu.edu, (2008).
3. Carraro, G., "Cost per feature vs. cost per tenant," Microsoft Corporation, blogs.msdn.com/gianpaolo/archive, (2007).
4. Carraro, G., "I don't believe we are still talking about whether SaaS = multi-tenancy … ," Microsoft Corporation, blogs.msdn.com/gianpaolo/archive, (2008).
5. Carraro, G., "Monetization: the next frontier of SaaS/S+S architecture," Microsoft Corporation, blogs.msdn.com/gianpaolo/archive, (2008).
6. Chappell, D., "A Short Introduction to Cloud Platforms," Microsoft Corporation, (August 2008).
7. Chappell, D., "Introducing the Azure Services Platform," Microsoft Corporation, (October 2008).
8. Chong, F. and G. Carraro, "Architecture Strategies for Catching the Long Tail," Microsoft Corporation, (April 2006).
9. Chong, F., "Application Marketplaces and the Money Trail," Microsoft Corporation, (March 2008).
10. "Utility Based Cloud Power," *Computers Journal*, February 12, 2009, www.dirjournal.com, (2009).
11. Foley, M., *Microsoft 2.0*, Indianapolis: Wiley Publishing, Inc., (2008).
12. Katzan, H., *Operating Systems: A Pragmatic Approach (2e)*. Hoboken, NJ: John Wiley and Sons, (1986).
13. Katzan, H., "Cloud Computing, I-Service, and IT Service Provisioning," *Journal of Service Science*, Volume 1, Number 2, (2008).
14. Katzan, H., *Service Science: Concepts, Technology, Management*, New York: iUniverse, Inc., (2008).
15. Knorr, E and G. Gruman, "What cloud computing really means," *InfoWorld*, April 07, 2008, www.infoworld.com, (2008).
16. Martin, R and J. Hoover , "Guide to Cloud Computing," *Information Week*, June 21, 2008, www.informationweek.com, (2008).
17. Miller, M., *Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online*, Indianapolis: Que Publishing, (2008).
18. Rappa, M., "The utility business model and the future of computing services," *IBM Systems Journal*, Volume 43, Number 1, (2004).

19. Perry, G., "How Cloud & Utility Computing are Different," GigaSpace Technologies, February 28, 2008, www.gigacom.com, (2008).
20. Smith, Adam, *The Wealth of Nations*, published as "An Inquiry Into the Nature and Causes of the Wealth of Nations" in London, England, (1776).
21. Wikipedia, "Software as a Service," www.wikipedia.com, (2008).

NOTES