

Cloud Computing: Case Studies and Total Costs of Ownership

This paper consists of four major sections: The first section is a literature review of cloud computing and a cost model. The next section focuses on detailed overviews of cloud computing and its levels of services: SaaS, PaaS, and IaaS. Major cloud computing providers are introduced, including Amazon Web Services (AWS), Microsoft Azure, and Google App Engine. Finally, case studies of implementing web applications on IaaS and PaaS using AWS, Linode and Google AppEngine are demonstrated. Justifications of running on an IaaS provider (AWS) and running on a PaaS provider (Google AppEngine) are described. The last section discusses costs and technology analysis comparing cloud computing with local managed storage and servers. The total costs of ownership (TCO) of an AWS small instance are significantly lower, but the TCO of a typical 10TB space in Amazon S3 are significantly higher. Since Amazon offers lower storage pricing for huge amounts of data, the TCO might be lower. Readers should do their own analysis on the TCOs.

A 2009 study from Ithaka suggested that faculty perceive three traditional functions of a library: (1) gateway, the library as

a starting point for locating information for research; (2) buyer, the library as a purchaser of resources; and (3) archive, the library as a repository of resources. The 2009 survey indicates a gradual decline in their perception of the importance of “gateway,” no change in “archive,” growth in “buyer,” and increased importance for two new roles: “teaching support” and “research support.”¹ To meet customers’ needs in these roles, libraries are innovating services, including catalogs and home websites (as “gateway” services), repository and digital library programs (as “archive,” “teaching support,” and “research support” services), and interlibrary loan (as a “buyer” and “research support” services).

These services rely on stable and effective IT infrastructure to operate. In the past, the growing needs of these web applications increased IT expenditures and work complexity. More web applications, more storage, and more IT support staff are weaved into centralized on-site IT infrastructure along with huge investments in physical servers, networks, and buildings. However, decreasing budgets in libraries have had huge impact on all aspects of library operations and staffing. Web applications running on local, managed servers might not be effective in technology nor efficient in cost. Web applications utilizing cloud computing can be much more effective and efficient in some cases.

Literature Review

There are a growing number of articles related to cloud computing in libraries. Chudnov described his personal experience of using cloud services Amazon EC2 and S3 in an informal tone, costing him 50 cents.² Jordan discussed OCLC’s strategies of building its next generation of services in cloud and provided a clear view of OCLC’s future directions for us.³ Mitchell wrote two

articles: one was to make a case for using the cloud;⁴ while the other provided more details of moving a library’s IT infrastructure (ILS, website, and digital library systems) to a cloud along with discussing motivation, results, and evaluation in three areas (quality and stability, impact on library services, and cost).⁵ On the cost discussion, Mitchell mentioned the difficulty of calculating technology Total Cost of Ownership (TCO) and cited two papers suggesting minimal cost savings. Mitchell suggested the same but did not provide detailed cost information. In comparison, this paper has a detailed breakdown cost analysis along with different services, such as web applications and storage. Mirsa and Mondal proposed a suitability index and a Return on Investment (ROI) model by taking into consideration impacts and real value.⁶ Their suitability index and ROI model is well thought but consider using the cloud for every aspect of all IT operations as a whole. As a result, a company using this model will have the final conclusion of a “suitable,” or “may or may not be,” or “not suitable.” However, modular IT operations and services (e.g., e-mail and storage) can be evaluated individually because these services can be easily upgraded or changed with minimal impacts to customers. I/O intensive services and storage intensive services have different resource requirements and thus the same evaluation criteria may not give an accurate picture of costs and benefits. For example, storing digital preservation files for libraries is a one-time data intensive operation. Giving the above different nature of IT operations and services, cloud computing may be suitable for some IT operations but not for others. Healy suggested that many companies did not have a complete financial analysis by missing staff retraining and system management. He listed the following areas for TCO: hardware, software, recurring licensing and maintenance, bandwidth,

Yan Han (hany@u.library.arizona.edu) is Associate Librarian, University of Arizona Libraries, Tucson, Arizona.

staffing allocation, monitoring, backup, failover, security audit and compliance, integration, training, and speed to implementation.⁷

The author published his first paper regarding cloud computing in 2010.⁸ Since then, the author has implemented and has been managing multiple web applications and services using IaaS and PaaS providers. Several web applications of the University of Arizona Libraries (UAL) have been migrated to the cloud. This paper focuses on enterprise-level applications and services, not individual-level cloud applications such as Google Docs. The purposes of this article are to

- define cloud computing and levels of services;
- introduce and compare major cloud computing providers;
- provide case studies of running two web applications (DSpace and a home grown Java application) utilizing cloud computing with justification;
- provide a comparison of TCO of running web applications comparing a cloud computing provider with a local managed server;
- provide a comparison of TCO of 10TB storage space comparing a cloud computing provider with local managed storage; and
- briefly discuss technology advantages of cloud computing.

Definition of Cloud Computing and Levels of Services

Cloud Computing Services and Providers

Cloud computing is becoming popular in the IT industry. Over the past few years, the supply-and-demand of this new area has been seeing a huge increase of investment in infrastructure and has been drawing broader uses in the United States. The author believes that it has a

potential to transform the IT industry and IT services, shifting the way IT infrastructure and hardware are designed, purchased, and managed. Many experts have their own version of cloud computing, which was discussed before.⁹ The National Institute of Standards and Technology defines cloud computing as “a model for enabling convenient, on-demand network access to a shared pool of configuration computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.”¹⁰ NIST also gives its three service models layered based on computing infrastructure:

- *Software as a Service* (SaaS) allows users to use the cloud computing providers’ applications through a thin client interface such as a web browser.¹¹ In the SaaS model, the cloud computing providers manage almost everything in the cloud infrastructure (e.g., physical servers, network, OS, applications). It is directly targeted for general end users. The end users can directly run applications on the clouds and do not need install, upgrade, and backup applications and their work. Typical SaaS products are Google Apps and Salesforce Sales CRM.
- *Platform as a Service* (PaaS) allows users to deploy their own applications on the provider’s cloud infrastructure under the provider’s environment such as programming languages, libraries, and tools.¹² In this model, the cloud computing providers manage everything except the application in the cloud infrastructure. PaaS is directly targeted for general software developers. They can develop, test, and run their codes on a PaaS platform. Typical examples of this model includes Google AppEngine, Windows Azure, and Joyent.
- *Infrastructure as a Service* (IaaS) allows users to manage processing, storage, networks, and other

fundamental computing resources so that they can deploy and run arbitrary software such as operating systems and applications.¹³ In this model, the providers only manage underlying physical cloud infrastructure (e.g. physical servers and network), and provides services via virtualization. The users have maximum control on the infrastructure as if they own underlying physical servers and network. Leading providers of this model includes Amazon, Linode, Rackspace, Joyent, and IBM Blue Cloud.

Major cloud computing providers include Amazon Web Services (AWS), Microsoft Windows Azure, and Google AppEngine. AWS is considered to be an IaaS, PaaS, and SaaS provider, which offers a collection of multiple computing services through the Internet, including a few well-known services such as Amazon Elastic Compute Cloud (EC2),¹⁴ Amazon Simple Storage Service (S3), and Amazon SimpleDB. EC2 started as a public beta in 2006. It allows users to pay for computing resources as they use them. With scalable use of computing resources and attractive pricing models, EC2 is one of the biggest brand names in cloud computing. It offers different OS options, including multiple Linux distributions, OpenSolaris, and Windows Server. EC2 uses Xen virtualization, each virtual machine is called an instance. An instance in EC2 has no persistent storage, and data stored will be lost if the instance is terminated. Therefore it is typical to use EC2 along with Amazon Elastic Block Store (EBS) or S3, which provides persistent storage for EC2 instances. Amazon claims that both EBS and S3 are highly available and reliable. A user can create, start, stop, and terminate server instances through multiple geographical locations for benefits of resource optimization and high availability. For example, a user can start an instance in northern Virginia, a

mirroring instance in Ireland, and another mirroring instance in Asia. Amazon keeps increasing its offering by introducing new PaaS and SaaS services, such as SimpleDB, Simple E-mail Service, and e-commerce.

Google App Engine is a PaaS provider offering a cloud platform for web applications in Google's data centers. It was released as a beta version in 2008 but is currently in a full service mode. AppEngine functions like a middle layer, which frees customers worrying about running OSs, modules, and libraries. It currently supports Python and Java programming languages and related frameworks, and it is expected to support more languages in the future. Google App Engine uses BigTable with its GQL (a SQL-like language). BigTable¹⁵ is Google's proprietary database, used in multiple Google applications such as Google Earth, Google Search, and App Engine. The design of GQL intentionally does not support "join" statement for multiple machine optimization.¹⁶ Unlike AWS, Google AppEngine has a nice feature that allows customers a taste of the platform: it is free of charge up to a certain level of resource use. After that, fees are charged for additional CPU time, bandwidth and storage.

Windows Azure also is a PaaS provider, which runs on Microsoft data centers. It provides a new way to run applications and storing data in Microsoft way. Microsoft customers can install and run applications on Microsoft Cloud. Customers are provided with two different instance types: web role instances and worker role instances. Customers can use a "web role instance" to accept incoming HTTP/HTTPS requests using ASP.NET, Windows Communication Foundation (WCF) or another.NET technology working with IIS. A "worker role instance" is not associated with IIS, but functions as a background job. The two instances can be combined to create desired web services. It is clear that Windows

Table 1. List of Major Cloud Computing providers

| Cloud Computing Provider | Layer |
|--------------------------|---------------------------|
| Akamai | PaaS, SaaS |
| Amazon Web Services | IaaS, PaaS, SaaS |
| EMC | SaaS |
| Eucalyptus | IaaS open source software |
| Google | PaaS(AppEngine), SaaS |
| IBM | PaaS, SaaS |
| Linode | IaaS |
| Microsoft | PaaS (Azure), SaaS |
| Rackspace | IaaS, PaaS, SaaS |
| Salesforce.com | PaaS, SaaS |
| VMware vCloud | PaaS, IaaS |
| Zoho | SaaS |

Azure allows non-Windows applications to run on the platform. For example, Apache web server can be run as a "worker role."¹⁷ There also are a few small-to-medium size providers such as Linode.¹⁸ Table 1 lists major cloud computing providers.

The cloud computing providers operate in two business models: variable (pay-for-your-usage) plans and fixed plans. Variable plans allows customers to pay only for the resources actually consumed (e.g., instance-hours, data transfer). AWS offers a variable plan. Google App Engine works in a similar way. Google App Engine offers two interesting features: daily budgets and free quotas. A daily budget allows customers to control the amount of resources used every day. The free quota is currently set as 6.5 hours of CPU time per day, 1 GB data in and out per day, and 1GB of data storage.¹⁹ By the end of each month, customers receive a bill listing the number of running hours, the amount of storage used, the size of data transfers, and other add-on services. Linode only offers a fixed plan. The charge is based on the amount of RAM, data storage, and data transfer by assuming an instance is always running. For example, the smallest instance has 360MB RAM,

16GB storage, 200GB transfer, and the cost is \$19.95 per month.²⁰ Customers pay up front.

Open-Source Cloud Computing Software and Private Cloud

Cloud computing also goes to open source if any person or organization wants to set up their own clouds. Eucalyptus is an open-source cloud computing system developed by the University of California at Santa Barbara. Some of its eye-catching features include full compatibility with Amazon EC2 public infrastructure and multiple hypervisors, which allows different virtual machines (e.g., Xen, KVM, VSphere) to run on one platform.²¹ Its open-source company, Eucalyptus Systems, provides technical supports to end users. Building a cloud infrastructure on cloud(s) is also possible and might be desirable in certain situations. Current Linux distributions work with Eucalyptus to provide private cloud services such Ubuntu Enterprise Cloud and Red Hat's Deltacloud.

Some organizations have been setting up private clouds to utilize advantages of cloud computing. The

private cloud eases concerns in the public cloud such as security of data, control of data, and legal issues. For example, an institution can build its own cloud infrastructure using Eucalyptus (or Ubuntu Cloud) with its own computing resources or simply using Amazon AWS. The private cloud computing service becomes customizable cloud computing resources which can be configured and reconfigured as needed. Why is this valuable? In traditional computing approaches, servers, storage, and networking equipment are purchased, configured, and then used without significant changes for three to five years until lives end. In this case, some planning must be scheduled ahead of time thinking of computing resource needs in three to five years. It is certain that additional resources (e.g., RAM, hard disks, CPU) will be reserved for future needs and are currently wasted. The private cloud reduces concerns regarding security and data control. However, one must still buy, build, and manage the private cloud, increasing TCO and reducing the cost benefit.

Case Studies: Applications on the Cloud

Case Study 1: DSpace Implementation and Analysis

Many libraries are running their institutional repositories at locally managed servers. UAL has been running its repositories since 2004 as one of the earliest DSpace adapters. One of the DSpace instances was tested on the cloud in January 2010 after comparing costs and supports. Later the author chose to run a production DSpace in AWS starting March 2010. The repository (<http://www.afghan.data.org/>) currently holds 1,800 titles of digitized unique Afghan materials. Since then, several content and system updates have been applied.

Cloud Computing Provider Selection and Implementation

A typical DSpace instance requires Java and related libraries, J2EE environment, and PostgreSQL as database backend. Three cloud computing providers have been evaluated: AWS, Linode, and Google AppEngine. Two instances were successfully installed and configured in AWS and Linode after a few days of testing. Building a DSpace instance on the cloud is the same process as running it on local except that it is much quicker to build, restart, rebuild, and backup. For example, an initial OS installation in a traditional server will take a few hours compared to doing the same task that takes a few minutes using an IaaS provider.

Installation on the AWS EC2 and Linode is almost the same except creating a login and setting up security policies. To log on to AWS, command line tools using an X.509 certificate using Public/Private key are by default. A generated keypair is required to SSH an instance and no password SSH option is provided. In addition, appropriate “security groups” are required to set up to enable network protocols. In this case, protocols such as SSH and HTTP along with typical port number 80 and 8080 must be enabled. Activities such as manage instances, creating images, and setup security policies can be set up through AWS web interface (see figure 1). Steps and commands of running regular operations can be found in the appendix. In Linode, using “root” to log on is allowed. Users do not need to set network and security policies, as protocols and ports are already open. In system administration practice, running applications without enforcing security policies does present security risks to applications and systems. Linode allows users to set up security policies. The author decided not to proceed with installation in Google AppEngine because of its proprietary database GQL. If implemented in Google AppEngine,

the work of modification of SQL-style code would have been significant. The author has a monthly bill of \$40 using an AWS small instance.

Case Study 2: Japanese GIF Holding Library Finder Application

The author helped the North American Coordinating Council on Japanese Library Resources (NCC) to develop and maintain a web service to identify Japanese Global ILL Framework (GIF) libraries to facilitate interlibrary loan (ILL) service. The application was developed in Java using J2EE framework, and run in typical Java servlet container such as Tomcat. The application was initially operated in a small, locally managed server, and was migrated to Linode and Google AppEngine in May 2010.

Cloud Computing Provider Selection and Implementation

Unlike case 1, the author tested and installed the application to AWS, Linode and Google AppEngine. AWS and Linode are IaaS providers which give users greater control over virtual nodes on their cloud infrastructure. Google AppEngine might be a better choice when applications run on normal OS environments, because system administration tasks can be completed by PaaS providers, saving users’ time and resources. As a PaaS provider, Google maintains its infrastructure environment such as OS, programming languages, and tools. Installing the application in Google AppEngine can go through an Eclipse plug-in or through command lines.

In this case, the GIF application is a simple system written in Java without any database transactions. Therefore Google App Engine’s proprietary GQL database is not a barrier. However, users should be aware that Google AppEngine has other unique features. For example,

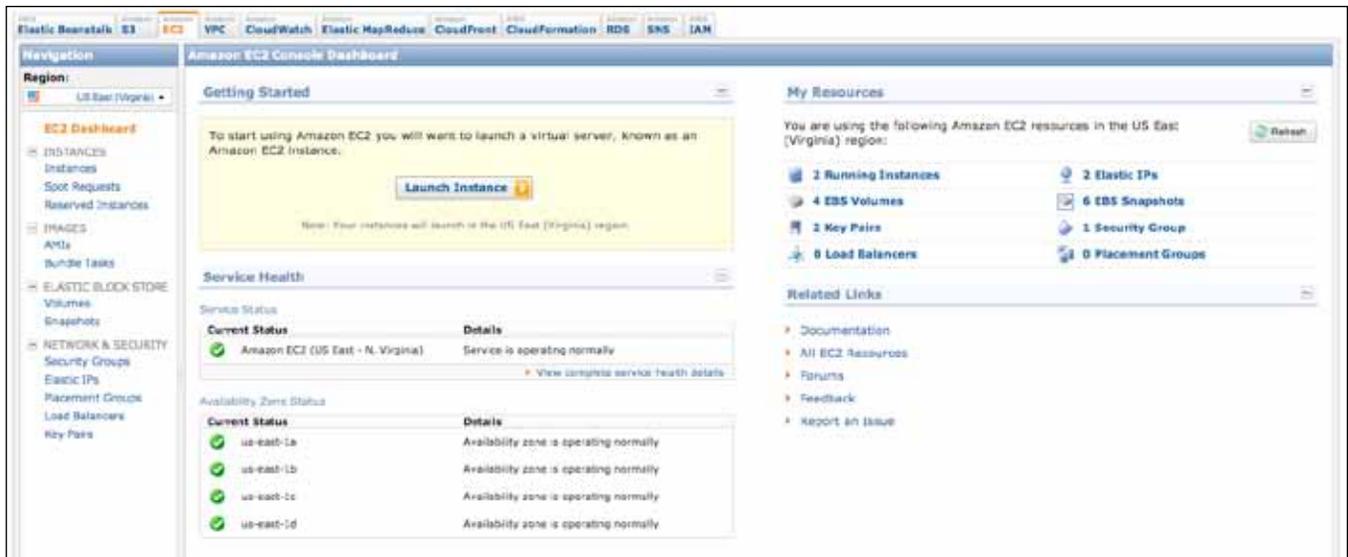


Figure 1. Amazon AWS Management Console

currently Google AppEngine only allows users to have their codes running in Python and Java; it uses its own database query language GQL. This creates an extra step for developers who are willing to migrate existing codes to Google and existing SQL queries have to be rewritten. In addition, other limitations with Google App Engine include allowing only a subset of the JRE standard edition and users are unable to create new threads.²² The cost of running the application on Google App Engine is great, because Google App Engine offers free of charge up to its free quota. Google identified 90 percent of applications were hosted free.²³ This is a great PaaS resource for small web applications.

Applications on the Cloud

Since 2009, the author has been running multiple web applications and services on multiple IaaS and PaaS providers and has been very happy regarding services and overall costs. The running applications and services are listed in table 2.

Analysis and Discussions

Cost Analysis

Running applications on the cloud gives many technical advantages and results in significant cost savings over running them on local managed servers. In this section, the author presents detailed cost comparisons between virtual managed nodes in the cloud computing and local managed storage and servers in the traditional model. Cost saving and low barriers to launch web services using the cloud is significant when considering easy start-up, scalability, and flexibility. One of the biggest advantages of the cloud computing lies in its on-demand, allowing users to start applications with minimal cost. The current cost of starting an instance on AWS is 0.03 per hour if reserved. *Above the Clouds: A Berkeley View of Cloud Computing* cites a comparison: "It costs \$2.56 to rent \$2 worth of CPU" and "costs are \$6.00 when purchasing vs. \$1.20-\$1.50 per month on S3."²⁴ Clearly Healy made

a good case for calculating the TCO.²⁵ In cases below, readers should be aware that there are the following assumptions:

- Software, training, licensing, and maintenance costs are the same by assuming using on the same software environment on the local managed infrastructure and on the cloud.
- Monitoring costs are the same based on the fact that monitoring software has to be hosted somewhere.
- Bandwidth and network costs ignored.
- Security audit and compliance ignored by assuming all data are open.

The author runs an instance of 100GB in AWS and a monthly bill of this node is around \$40. In comparison, if running a local managed server, a physical server would have been purchased. In our case, a comparison of TCO shows that the cloud computing model has a significant 50 percent cost saving, assuming a server life expectancy is five years.

Table 2. Some UAL Web Applications and Cloud Computing Service Providers

| Computing Infrastructure | Functions | Applications | Computing Environment | Instances | Service Providers |
|--------------------------|----------------------------------|---------------------------------|---------------------------------|---|------------------------------------|
| Data Storage | Data storage | N/A | Linux / Windows | Data storage using EBS or S3 | AWS |
| Access | Digital repository | DSpace | J2EE, Java, Tomcat, PostgreSQL, | Afghanistan Digital Collections < http://www.afghandata.org > | AWS Linode |
| | Content Management System | Joomla | Linux, Apache, PHP, MySQL, | Afghanistan Digital Libraries < http://www.afghandigitallibraries.org/ > | AWS Linode |
| | Website | HTML | HTML | Sonoran Desert Knowledge Exchange < http://sdke.org/ > | AWS Linode |
| | Integrated Library System | Koha | Linux, Apache, Perl, MySQL | Afghanistan Higher Education Union Catalog < http://www.afghandigitallibraries:4000/ > | AWS Linode |
| | Web applications | Home-grown J2EE web application | J2EE, Java, Tomcat | Japanese GIF (Global Interlibrary-loan) Holding Finder < http://74.82.5.165:8080/jpn/ > at Linode < http://yhan818.appspot.com/ > at Google App Engine | AWS Linode Google App Engine |
| Computing Services | Monitoring | Nagios | Linux, Perl | Internal application | AWS Linode |
| | Networked Devices Administration | SSH, SFTP | Linux | N/A | AWS Linode |

- The TCO of an AWS small instance for 5 years: \$2,750–\$3,750.
 - Hardware: \$0.
 - Operation expense: \$2,750–\$3,750
 - System administrator cost: \$0–\$1,000?. By eliminating physical infrastructure, there is no need or minimal cost to manage a server.
 - \$2,750 = \$350 (AWS initial subscription fee) + \$40/month x 12 months x 5 years. The instance’s capacity can be found on AWS, and CPU power can be evaluated by using `/proc/cpuinfo`. Amazon indicated that “One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0–1.2 GHz 2007 Opteron or 2007 Xeon processor.”²⁶
- The TCO of a physical server comparable to an AWS small instance for 5 years: \$5,858–\$7,608.
 - An AWS small instance is roughly 50 percent of computing power of a server quoted. (The TCO here is calculated as 50 percent of \$11,715–\$15,215).
 - Hardware: \$4,525.
 - \$4,525 = \$2,658 (server) + \$1,125 (3-year support) + \$1,125 x2 /3 (additional 2-year support). Note: Dell PowerEdge server: Intel Xeon E56302.53Ghz with 5-year support for mission critical 6-hours repair (source: Dell.com quoted on Oct. 20, 2010).
- Operation expense: \$7,190–\$10,690. Ignoring downtime and failure expenses, insurance cost, technology training, and backup process.
 - System administrator cost: \$3,500–\$7,000 = 5 years x 1–2 percent time x (50,000 salary + 50000 x 40 percent benefits). 1–2 percent time is about 5–10 minutes per day assuming this administrator works at 8 hours per day 5 days per week at 100 percent capacity.
 - Space cost: \$1,500.
 - Space cost for a book in UAL is \$2.80 per year. A physical server is estimated to be \$300 dollars per year for space.
 - Electricity cost: \$2,190.

- Electricity cost: $\$2,190 = 5 \text{ years} \times 365 \text{ days/year} \times 24 \text{ hours/day} \times 0.5 \text{ kilowatt/hour} \times \$0.10/\text{kilowatt}$.

Most libraries running digital library programs require big storage for preserving digitization files. The analysis below just illustrates a comparison of the TCO of 10TB space. It shows that the TCO of locally managed storage has lower costs than Amazon S3's storage TCO. Though the cloud computing model still have the advantage of on-demand, avoid big initial investment on equipment, the author believes that locally managed storage may be a better solution if planned well. Since Amazon S6 storage pricing decreases from \$0.14/GB to \$0.095/GB over 500TB, Amazon S3's TCO might be lower if an organization has huge amounts of data. The author suggests readers should do their own analysis.

- The TCO of 10TB in Amazon S3 per year: \$16,800. Note: Amazon S3 replicate data at least 3 times, assuming these preservation files do not need constant changes. Otherwise, data transfer fees could be high.
 - Operation expense: \$16,800 per year.
 - $\$16,800 = \$1,400/\text{month} \times 12 \text{ months}$. (based on Amazon S3 pricing of \$0.14/GB per month)
 - Network cost ignored.
- The TCO of a 10TB physical storage per year: \$11,212–\$12,612.
 - To match reliability of Amazon S3, local managed storage needs three copies of data: two in hard disk and one in tape. Note: Dell AX4–5I SAN storage: quoted on October 26, 2010. Replicate data 3 times, including 2 copies in hard disks, one copy in tape. Ignoring time value of money, 3 percent inflation per year based on CPI statistic data.
 - Hardware: \$4,168 per year.
 - \$20,840 a SAN storage

includes 12TB hard disks (about 10TB usable space after RAID 5 configuration) with 5-year support, assuming 5-year life expectancy.

- Operation expense: \$1,438–\$2,138 per year.
 - System administrator cost: \$700–\$1,200. See above.
 - Space cost: \$300. See above.
 - Electricity costs: \$438 per year. See above.
 - Network cost ignored.

Technology Analysis

There is no need to purchase a server; no need to initial a cloud node; no need to setup security policies; no need to install Tomcat, Java and J2EE environment; and no need to update software. Compared to the traditional approach, PaaS eliminates upfront hardware and software investment, reduces time and work for setting up running environment, and removes hardware and software upgrade and maintenance tasks. IaaS eliminates upfront hardware investment along with other technical advantages discussed below.

The cloud computing model offers much better scalability over the traditional model due to its flexibility and lower cost. In our repository, the initial storage requirement is not significant, but can grow over time if more digital collections are added. In addition, the number of visits is not high, but can increase significantly later. An accurate estimate of both factors can be difficult. In the traditional model, a purchased server has preconfigured hardware with limited storage. Upgrading storage and processing power can be costly and problematic. Downtime will be certain during the upgrade process. In comparison, the cloud computing model provides an easy way to upgrade storage and processing power with no downtime if handling well. Bigger storage and larger instances with high-memory or high-CPU can be added or removed to

meet users' needs at will.

Rebuilding nodes and creating imaging are also easier on the cloud. Server failure resulting from hardware error can result in significant downtime. The UAL has a few server failure in the past few years. Last year a server's RAID hard drives failed. The time spent on ordering new hard disks, waiting for server company technician's arrival, and finally rebuilding software environment (e.g., OS, web servers, application servers, user and group privileges) took six or more hours, not to mention about stress rising among customers due to unavailability of services. Mirroring servers could minimize service downtime, but the cost would be almost doubled. In comparison, in the cloud computing model, the author took a few snapshots using the AWS web management interface. If a node fails, the author can launch an instance using the snapshot within a minute or two.

Factors such as software and hardware failure, natural disasters, network failure, and human errors are the main causes for system downtime. The cloud computing providers generally have multiple data centers in different regions. For instance, Amazon S3 and Google AppEngine are claimed to be highly available and highly reliable. Both AWS and Google App Engine offer automatic scaling and load balancing. The cloud computing providers have huge advantages in offering high availability to minimize hardware failure, natural disasters, network failure, and human errors, while the locally managed server and storage approach has to be invested a lot to reduce these risks. In 2009 and 2010 the University of Arizona has experienced at least two network and server outages each lasting a few hours; one failure was because of human error and the other was because of a power failure from Tucson Electric Power. When a power line was cut by accident, what can you do? In comparison, over the past two years minimal downtime from

the cloud computing providers was reported.

There are some issues when implementing cloud computing. *Above the Clouds: A Berkeley View of cloud computing* discusses ten obstacles and related opportunities for cloud computing.²⁷ All of these obstacles and opportunities are technical. The author's first paper on this topic also discusses legal jurisdiction issues when considering cloud computing.²⁸ Users should be aware of these potential issues when making a decision of adopting the cloud.

Summary

This paper starts with literature review of articles in cloud computing, some of them describing how libraries are incorporating and evaluating the cloud. The author introduces cloud computing definition, identifies three-level of services (SaaS, PaaS, and IaaS), and provides an overview of major players such as Amazon, Microsoft, and Google. Open source cloud software and how private cloud helps are discussed. Then he presents case studies using different cloud computing providers: case 1 of using an IaaS provider Amazon and case 2 of using a PaaS provider Google. In case 1, the author justifies the implementation of DSpace on AWS. In case 2, the author discusses advantages and pitfalls of PaaS and demonstrates a small web application hosted in Google AppEngine. Detailed analysis of the TCOs comparing AWS with local managed storage and servers are presented. The analysis shows that the cloud computing has technical advantages and offers significant cost savings when serving web applications. Shifting web applications to the cloud provides several technical advantages over locally managed servers. High availability, flexibility, and cost-effectiveness are some of the most important benefits. However, the locally managed storage is still an attractive solution in a typical

case of 10TB storage. Since Amazon offers lower storage pricing for huge amounts of data, readers are recommended to do their own analysis on the TCOs.

References

1. Roger C. Schonfeld and Ross Housewright, *Faculty Survey 2009: Key Strategic Insights for Libraries, Publishers, and Societies*, 2010, <http://www.ithaka.org/ithaka-s-r/research/faculty-surveys-2000-2009/faculty-survey-2009> (accessed Apr. 20, 2010).
2. Daniel Chudnov, "A View From the Clouds," *Computers in Libraries* 30, no. 3 (2010): 33–35.
3. Jay Jordan, "Climbing Out of the Box and Into the Cloud: Building Web-Scale for Libraries," *Journal of Library Administration* 51, no. 1 (2011): 3–17.
4. Erik Mitchell, "Cloud Computing and Your Library," *Journal of Web Librarianship* 4, no. 1 (2010): 83–86.
5. Erik Mitchell, "Using Cloud Services For Library IT Infrastructure," *Code4Lib Journal* 9 (2010), <http://journal.code4lib.org/articles/2510> (accessed Feb 10, 2011).
6. Subhas C. Misra and Arka Mondal, "Identification of a Company's Suitability for the Adoption of Cloud Computing and Modelling its Corresponding Return on Investment," *Mathematical & Computer Modelling* 53 (2011): 504–21, doi: 10.1016/j.mcm.2010.03.037.
7. Michael Healy, "Beyond CYA as a service," *Information Week* 1288 (2011): 24–26.
8. Yan Han, "On the Clouds: A New Way of Computing," *Information Technology & Libraries* 29, no. 2 (2010): 88–93.
9. Ibid.
10. Peter Mell and Tim Grance, *The NIST Definition of Cloud Computing*, NIST, <http://csrc.nist.gov/groups/SNS/cloud-computing/> (accessed Oct. 21, 2010).
11. Ibid.
12. Ibid.
13. Ibid.
14. Amazon, *Amazon Elastic Compute Cloud (Amazon EC2)*, 2010, <http://aws.amazon.com/ec2/> (accessed Oct. 21, 2010).
15. Fay Chang et al., "Bigtable: A Distributed Storage System for Structure Data," in *7th Symposium on Operating Systems Design and Implementation (OSDI*

'06), Nov. 6–8, 2006, Seattle, Wash., https://www.usenix.org/events/osdi06/tech/chang/chang_html/ (accessed Apr. 21, 2010).

16. Google, "GQL Reference, 2010, <http://code.google.com/appengine/docs/python/datastore/gqlreference.html> (accessed Apr. 21, 2010); Google Developers, "Campfire One: Introducing Google App Engine (pt. 3)," 2010, <http://www.youtube.com/watch?v=oG6Ac7d-Nx8> (accessed Apr. 21, 2010).
17. David Chappell, "Introducing Windows Azure," 2009, http://download.microsoft.com/download/e/4/3/e43bb484-3b52-4fa8-a9f9-ec60a32954bc/Azure_Services_Platform.pdf (accessed Apr. 2, 2010).
18. Linode, "Linode—Xen VPS Hosting," 2010, <http://www.linode.com/> (accessed Apr. 7, 2010).
19. Google, "Quotas—Google App Engine," 2010, <http://code.google.com/appengine/docs/quotas.html> (accessed Oct. 21, 2010).
20. Jay Jordan, "Climbing Out of the Box and Into the Cloud: Building Web-Scale for Libraries," *Journal of Library Administration* 51, no. 1 (2011): 3–17.
21. Nurmi Daniel et al., "The Eucalyptus Open-Source Cloud-Computing System," in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2009, doi: 10.1109/CCGRID.2009.93.
22. Google, "The JRE White List—Google App Engine—Google Code," 2010, <http://code.google.com/appengine/docs/java/jrewhitelist.html> (accessed Apr. 9, 2010); Google, "The Java Servlet Environment," 2010, <http://code.google.com/appengine/docs/java/runtime.html> (accessed Apr. 9, 2010).
23. Google, "Changing Quotas To Keep Most Apps Serving Free," 2009, <http://googleappengine.blogspot.com/2009/06/changing-quotas-to-keep-most-apps.html> (access Oct. 21, 2010).
24. Michael Armbrust et al., *Above the Clouds: A Berkeley View of Cloud Computing* (EECS Department, University of California, Berkeley: Reliable Adaptive Distributed Systems Laboratory, 2009), <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html> (accessed July 1, 2009).
25. Amazon, "Amazon EC2 Pricing," 2010, <http://aws.amazon.com/ec2/pricing/> (accessed Feb. 20, 2010).
26. Michael Healy, "Beyond CYA as a service," *Information Week* 1288 (2011): 24–26.

27. Erik Mitchell, "Cloud Computing and Your Library," *Journal of Web Librarianship* 4, no. 1 (2010): 83–86.

28. Michael Armbust et al., *Above the Clouds: A Berkeley View of Cloud Computing*,

(EECS Department, University of California, Berkeley: Reliable Adaptive Distributed Systems Laboratory, 2009), <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>

(accessed July 1, 2009).

29. Yan Han, "On the Clouds: A New Way of Computing," *Information Technology & Libraries* 29, no. 2 (2010): 88–93.

Appendix. Running Instances on Amazon EC2

Task 1: Building a New DSpace instance

- Build a clean OS: select an Amazon Machine image (AMI) such as Ubuntu 9.2 to get up and running in a minute or two.
- Install required modules and packages: install Java, Tomcat, PostgreSQL, and mail servers.
- Configure security and network access on the node.
- Install and configure DSpace: install system and configure configuration files.

Task 2: Reloading a New DSpace instance

- Create a snapshot of current node with the EBS if desired: use AWS's management tools to create a snapshot.
- Register the snapshot using AWS's management tools and write down the snapshot id, specify the kernel and ramdisk.

command: `ec2-register`: registers the AMI specified in the manifest file and generate a new AMI ID (see Amazon EC2 Documentation)

(example: `ec2-register -s snap-12345 -a i386 -d "Description of AMI" -n "name-of-image" --kernel aki-12345 --ramdisk ari-12345`)

- In the future, a new instance can be started from this snapshot image in less than a minute.

command: `ec2-run-instances`: launches one or more instances of the specified AMI (see Amazon EC2 Documentation)

(example: `ec2-run-instance ami-a553bfcc -k keypair2 -b /dev/sda1=snap-c3fcd5aa: 100:false`)

Task 3: Increasing Storage Size of Current Instance

- To create an instance with desired persistent storage (e.g., 100 GB)

command: `ec2-run-instances`: launches one or more instances of the specified AMI (see Amazon EC2 Documentation)

(example: `ec2-run-instances ami-54321 -k ec2-key1 -b /dev/sda1=snap-12345:100:false`)

- If you boot up an instance based on one of these AMIs with the default volume size, once it's started up you can do an online resize of the file system:

Command: `resize2fs`: ext2 file system resizer

(example: `resize2fs /dev/sda1`)

Task 4: Backup

- Go to AWS web interface and navigate to the "Instances" panel.
- Select our instance and then choose "Create Image (EBS AMI)."
- This newly created AMI will be a snapshot of our system in its current state.