A File Storage Service on a Cloud Computing Environment for Digital Libraries

Victor Jesús Sosa-Sosa and Emigdio M. Hernandez-Ramirez

ABSTRACT

The growing need for digital libraries to manage large amounts of data requires storage infrastructure that libraries can deploy quickly and economically. Cloud computing is a new model that allows the provision of information technology (IT) resources on demand, lowering management complexity. This paper introduces a file-storage service that is implemented on a private/hybrid cloud-computing environment and is based on open-source software. The authors evaluated performance and resource consumption using several levels of data availability and fault tolerance. This service can be taken as a reference guide for IT staff wanting to build a modest cloud storage infrastructure.

INTRODUCTION

The information technology (IT) revolution has led to the digitization of every kind of information.[1] Digital libraries are appearing as one more step toward easy access to information spread throughout a variety of media. The digital storage of data facilitates information retrieval, allowing a new wave of services and web applications that take advantage of the huge amount of data available.[2] The challenges of preserving and sharing data stored on digital media are significant compared to the print world, in which data "stored" on paper can still be read centuries or millennia later. In contrast, only ten years ago, floppy disks were a major storage medium for digital data, but now the vast majority of computers no longer support this type of device. In today's environment, selecting a good data repository is important to ensure that data are preserved and accessible. Likewise, defining the storage requirements for digital libraries has become a big challenge. In this context, IT staff—those responsible for predicting what storage resources will be needed in the medium term—often face the following scenarios:

- Prediction of storage requirements turn out to be below real needs, resulting in resource deficits.
- Prediction of storage requirements turn out to be above real needs, resulting in expenditure and administration overhead for resources that end up not being used.

In these situations, considering only an efficient strategy to store documents is not enough.[3] The acquisition of storage services that implement an *elastic* concept (i.e., storage capacity that can be

Victor Jesús Sosa-Sosa (vjsosa@tamps.cinvestav.mx) is Professor and Researcher at the Information Technology Laboratory at CINVESTAV, Campus Tamaulipas, Mexico. **Emigdio M. Hernandez-Ramirez** (emhr1983@gmail.com) is Software Developer, SVAM International, Ciudad Victoria, Mexico.

increased or reduced on demand, with a cost of acquisition and management relatively low) becomes attractive. Cloud computing is a current trend that considers the Internet as a platform providing on-demand computing and software as a service to anyone, anywhere, and at any time. Digital libraries naturally should be connected to cloud computing to obtain mutual benefits and enhance both perspectives.[4] In this model, storage resources are provisioned on demand and are paid according to consumption. Services deployment in a cloud-computing environment can be implemented three ways: private, public, or hybrid. In the private option, infrastructure is operated solely for a single organization; most of the time, it requires an initial strong investment because the organization must purchase a large amount of storage resources and pay for the administration costs. The public cloud is the most traditional version of cloud computing. In this model, infrastructure belongs to an external organization where costs are a function of the resources used. These costs include administration. Finally, the hybrid model contains a mixture of private and public. A cloud-computing environment is mainly supported by technologies such as virtualization and service-oriented architectures.

A cloud environment provides omnipresence and facilitates deployment of file-storage services. It means that users can access their files via the Internet from anywhere and without requiring the installation of a special application. The user only needs a web browser.

Data availability, scalability, elastic service, and pay-per-use are attractive characteristics found in the cloud service model. Virtualization plays an important role in cloud computing. With this technology, it is possible to have facilities such as multiple execution environments, sandboxing, server consolidation, use of multiple operating systems, and software migration, among others. Besides virtualization technologies, emerging tools that allow the creation of cloud-computing environments also support this type of computing model, providing dynamic instantiation and release of virtual machines and software migration.

Currently, it is possible to find several examples of public cloud storage, such as Amazon S3 (http://aws.amazon.com/en/s3), RackSpace (http://www.rackspace.com/cloud/public/files), and Google Storage (https://developers.google.com/storage), each of which provide high availability, fault tolerance, and services and administration at low cost. For organizations that do not want to use a third-party environment to store their data, private cloud services may offer a better option, although the cost is higher. In this case, a hybrid cloud model could be an affordable solution. Organizations or individual users, can store sensitive or frequently used information in the private infrastructure and less sensitive data in the public cloud.

The development of a prototype of a file-storage service implemented on a private and hybrid cloud environment using mainly free and open-source software (FOSS) helped us to analyze the behavior of different replication techniques. We paid special attention to the cost of the system implementation, system efficiency, resource consumption, and different levels of data privacy and availability that can be achieved by each type of system.

INFRASTRUCTURE DESCRIPTION

The aim of this prototyping project was to design and implement scalable and elastic distributed storage architecture in a cloud-computing environment using free, well-known, open-source tools. This architecture represents a feasible option that digital libraries can adopt to solve financial and technical challenges when building a cloud-computing environment.

The architecture combines private and public clouds by creating a hybrid cloud environment. For this purpose, we evaluated tools such as KVM and XEN, which are useful for creating virtual machines (VM).[5] Open Nebula (http://opennebula.org), Eucalyptus (http://www.eucalyptus.com), and OpenStack (http://www.openstack.org) are good, free options for managing a cloud environment. We selected Open Nebula for this prototype.

Commodity hard drives have a relatively high failure rate, hence our main motivation to evaluate different replication mechanisms, providing several levels of data availability and fault tolerance. Figure 1(a) shows the core components of our storage architecture (the private cloud), and figure 1(b) shows a distributed storage web application named Distributed Storage On the Cloud (DISOC), used as a proof of concept. The private cloud also has an interface to access a public cloud, thus creating a hybrid environment.

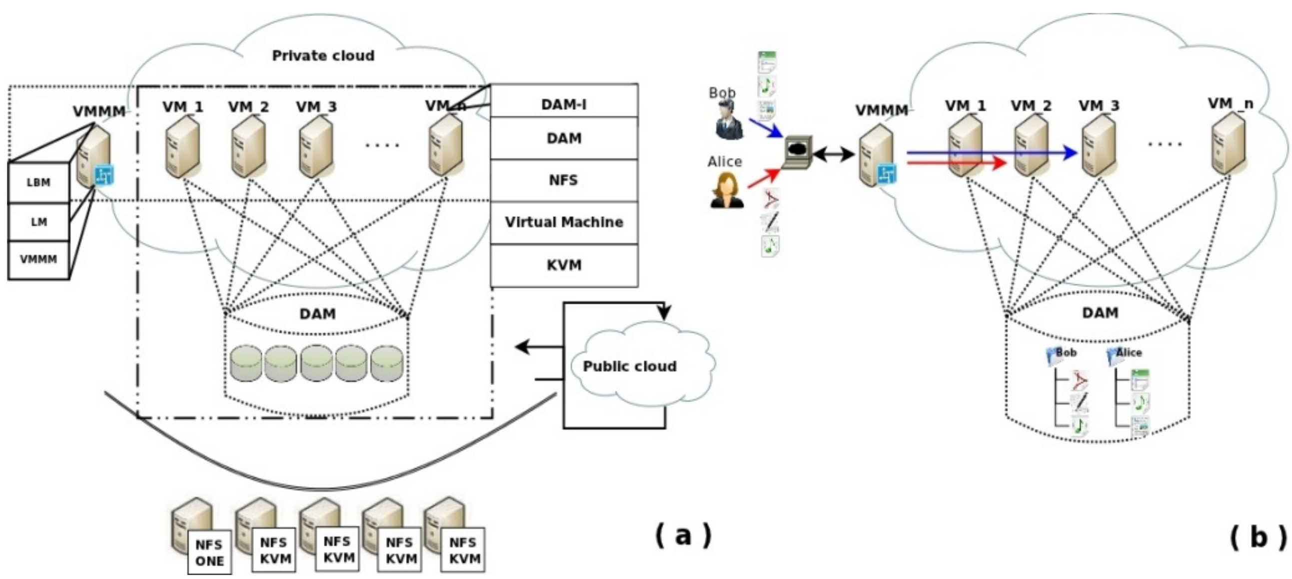

Figure 1. Main Components of the Cloud Storage Architecture

The core components and modules of the architecture are the following:

- Virtual Machine (VM). We evaluated different open-source were evaluated, such as KVM and XEN, for the creation of virtual machines.[6] Some performance tests were done, and KVM showed a slightly higher performance than XEN. We selected KVM as the main Virtual Machine Manager (VMM) for the proposed architecture. VMMs also are called

Hypervisors. Each VM has a Linux operating system that is optimized to work in virtual environments and requires a minimum consumption of disk space. The VM also includes an Apache web server, a PHP module, and some basic tools that were used to build the DISOC web application. Every VM is able to transparently access a pool of disks through a special data access module, which we called DAM. More details about DAM follow.

- Virtual Machine Manager Module (VMMM). This has the function of dynamic instantiation and de-instantiation of virtual machines depending on the current load on the infrastructure.
- Data Access Module (DAM). All of the virtual disk space required by every VM was obtained through the Data Access Module Interface (DAM-I). DAM-I allows VMs to access disk space by calling DAM, which provides transparent access to the different disks that are part of the storage infrastructure. DAM allocates and retrieves files stored throughout multiple file servers.
- Load Balancer Module (LBM). This distributes the load among different VMs instantiated on the physical servers that make up the private cloud.
- Load Manager (LM). This monitors the load that can occur in the private cloud.
- Distributed Storage on the Cloud (DISOC). This is a web-based file-storage system that is used as a proof of concept and was implemented based on the proposed architecture.

REPLICATION TECHNIQUES

High availability is one of the important features offered in a storage service deployed in the cloud. The use of replication techniques has been the most useful proposal to achieve this feature. DAM is the component that provides different levels of data availability. It currently includes the following replication policies: no-replication, total-replication, mirroring, and IDA-based replication.

- No-Replication. This replication policy represents the data availability method with the lowest level of fault tolerance. In this method, only the original version of a file is stored in the disk pool. It follows a round-robin allocation policy whereby load assignation is made based on a circularly linked list, taking into account disk availability. This policy prevents all files from being allocated to the same server, providing a minimal fault tolerance in case a server failure.
- Mirroring. This replication technique is a simple way to ensure higher availability without high resource consumption. In this replication, every time a file is stored in a disk, the DAM creates a copy and places it on a different disk.
- Total-replication. This represents the highest data availability approach. In this technique, a copy of the file is stored on all of the file servers available. Total-replication also requires the highest consumption of resources.
- IDA-based replication. To provide higher data availability with less impact on the consumption of resources, an alternative approach based on information-dispersal techniques can be used. The Information Dispersal Algorithm (IDA) is an example of this

strategy.[7] When a file (of size |F|) is required to be stored using the IDA, the file is partitioned into *n* fragments of size |F|/*m*, where *m*<*n*. These fragments are distributed in *n* different disks. The IDA only needs to obtain *m* fragments to reconstruct the original file. In this context, even if *n-m* disks failed, the file would still be recovered. It is desirable that no more than *n-m* file servers fail. The IDA provides better fault tolerance than mirroring without the need to completely replicate the original file. In this prototype, the IDA was evaluated with *n*=5 and *m*=3 (which means only 60 percent of the original file was replicated). The IDA is attractive for a hybrid cloud environment because it is not necessary to save the entire file on a single file server (disk). In this way, it could be possible to send *k* fragments of the file (where *k*<*m*) to a public cloud storage without revealing the complete content of the original file. This strategy works similar to Redundant Array of Independent Disks (RAID) 5 in that it is a type of block-level striping storage technology, which distributes parity or data about the data along with the data itself.[8] However, the IDA uses a different strategy for data reconstruction and could be implemented in a distributed environment using from 2 to *n* storage servers.

PERFORMANCE EVALUATION

We implemented and used a prototype of this architecture as the evaluation scenario. It includes five commercial PCs (commodity), whose characteristics are shown in the first section of table 1. The features of the VMs that were instantiated on these PCs are shown in the second section of table 1.

Physical Machines				
PCs	Cores	Memory	Hard disk	Network
1 pc	4	4 Gb	640 Gb	Ethernet 10/100
4 pc	2	2 Gb	250 Gb	Ethernet 10/100
Virtual Machines				
8 vm	1	1 Gb	1 Gb	Virtual Ethernet
1 vm	1	128 Mb	1 Gb	Virtual Ethernet

Table 1. Characteristics of the Physical PCs and VMs Used in the Private Cloud

We created nine VMs in a private cloud environment for this evaluation. To build and test a hybrid cloud environment, it was necessary to access a public-storage cloud (third-party infrastructure). We used two public storage providers in this experiment: Dropbox and Phoenix (also known as TreeStore).[9] DAM also was responsible for offering transparent access to the external storage infrastructure. Both providers required valid user credentials on a per-request basis. The Dropbox

API additionally requires a developer key. It is relevant to note that Dropbox also is able to keep files in the Amazon S3 storage infrastructure.

We emulated different workloads, running concurrent client applications that sent many parallel file upload and download requests to our cloud-storage prototype. The private-storage cloud configuration was first tested by receiving 50, 100, and 150 parallel requests. It is worth mentioning that when testing the hybrid cloud configuration, it was not possible to send the same number of parallel requests used in the private configuration. It was necessary to decrease this number because the public cloud-storage providers (Dropbox and TreeStore) could view parallel requests of the same number as an attack against their servers and, consequently, block the service. Since we were more interested in having a hybrid cloud scenario than in comparing public vs. private cloud, we decided to send only 10, 20, and 30 parallel requests to the public storage in the hybrid cloud configuration. The private and hybrid (private + public) cloud-storage scenario was designed to evaluate the impact of having an elastic service and the behavior of the cloud-storage infrastructure when applying different replication techniques to offer several levels of data availability.

The Impact of Having an Elastic Service

This section presents a comparison between an elastic storage service (using virtual machines) and a fixed storage service (using only a physical machine). In the elastic service, a new VM is instantiated when a workload exceeds a defined upper threshold. In the opposite situation (when the response time is shorter than a lower threshold), a VM is released. Several response-time restrictions were configured during this evaluation to determine the best time to instantiate or release a VM. The evaluation used different workloads generated by Autobench.[12] A physical machine with a single hard disk receiving an increasing workload was compared with the same physical machine that incrementally instantiated a set of VMs running the same workload. For this test, the workload consisted of a set of requests of a dynamically generated PHP webpage. This webpage emulated the processing time on a server by running a sorting algorithm (bubble type). Different levels of load were emulated by sending variable-size lists of elements that had to be sorted by the server process. The results are shown in figure 2. The vertical axis represents the average response time (in seconds) after sending different requests to the storage service. The horizontal axis represents the evaluation time (500 seconds), which was the period of time different clients were sending requests to the storage service. The Fixed line indicates the performance when the service load balancer accessed one physical machine (i.e., the fixed storage service). The Elastic line represents the performance when the service load balancer accessed one to three VM instantiations in the same physical machine (i.e., the elastic service). The instantiation and activation time (called the Deployment time) of a new VM was between sixty and ninety seconds. At the beginning of the test, when the workload is low, the response time offered by the fixed service (running only on one physical machine) is better—in some cases up to four or five orders of magnitude—than that obtained in the execution of the service accessing only one virtual machine. However, the elastic service improved the response time by including more VMs in the same physical machine.

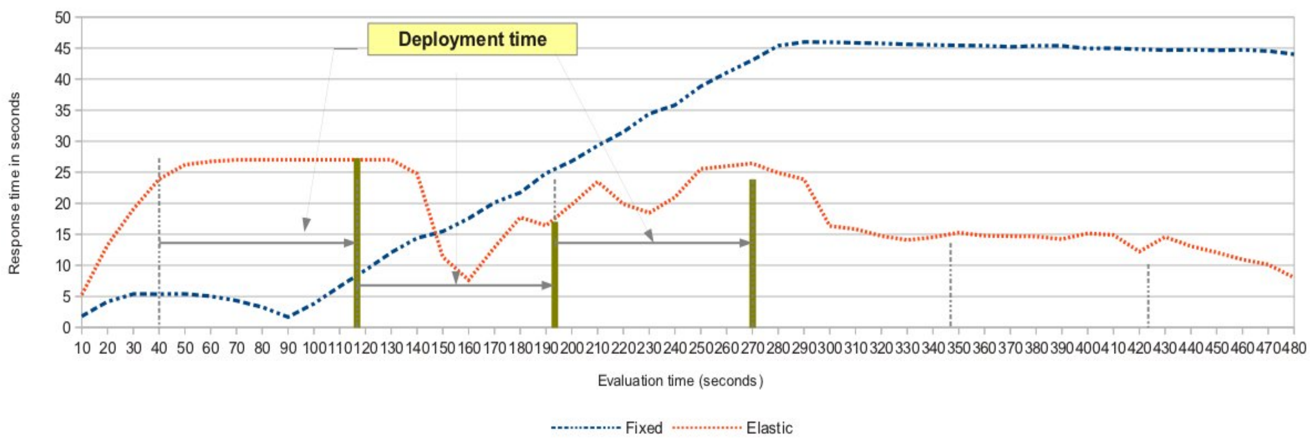

Figure 2. Performance Comparison between a Fixed and Elastic Storage Service

Response times of fifteen seconds and five seconds were defined as the upper and lower threshold, respectively. Figure 2 shows how the response time in the elastic service had a considerable reduction over the course of the test. This behavior was caused by the new VMs that were instantiated in the storage service and taken into account by the load balancer. At the end of this evaluation, the elastic service was able to finish the workload offering good response times, while the fixed service collapsed and could not finish all of the requests sent by the client.

Evaluation of Different Replication Techniques

With DAM, it is possible to define the level of data availability in the cloud-storage prototype by applying different replication techniques. We defined a benchmark to evaluate the benefits when using a distributed storage system compared with a centralized version. In the centralized version, DAM had access to a single disk using only one VM (emulating a centralized process) with a single file server (emulating centralized storage).

The distributed version considered the use of distributed processes (eight VMs) accessing a distributed storage system (five disks distributed on different storage servers encapsulated by DAM). Since the replication with the IDA technique is attractive for a hybrid cloud service, its behavior was compared in both cases: when it is only accessing a private storage cloud and when it is accessing both a private and a public storage cloud (hybrid model).

Two main metrics were taken into account for these experiments. The first was response time, the time from when the user clicks on the button to upload or download a file to when the file loading or downloading has finished—in this test until the TCP connection is closed. The second was service time, the time needed by DAM to locate a file (or part of it) and prepare the file to be read by the system component that is requesting it.

Replication Techniques in a Private Cloud

This test evaluated the storage service response and service time using different levels of data availability and fault tolerance on a private cloud environment. We implemented different replication techniques in DAM to carry out this task.

The left side of the figure 3 shows the response and service time produced by different replication techniques during the file uploading process. In this case, even though the total replication technique had a poor service time, the storage service showed its worst performance when implementing a centralized architecture. It is interesting to see how the no-replication technique is showing the best performance during the uploading process. This behavior may exist because this technique does not require any additional work for replicating a file and does not have to send any additional data through the network.

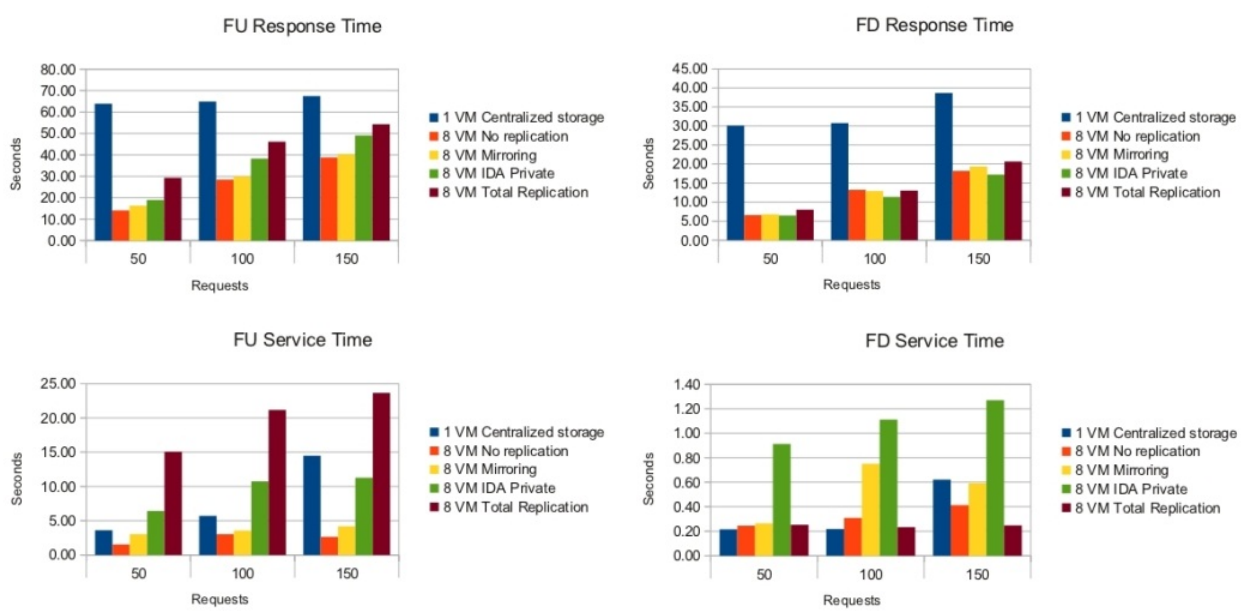

Figure 3. Average Response and Service Time for File Uploading (FU) and Downloading (FD) using Different Replication Techniques in a Private Cloud Environment

The right side of figure 3 shows the response and service time perceived during the file downloading process using the private cloud. In this case, even though the IDA technique is producing the worst service time, the response time showed by the different replication techniques was similar. The IDA shows a competitive response time and offers an acceptable level of fault tolerance. The total replication technique offers high data availability and fault tolerance, but it is not producing the best response time. This slowness could be because the way DAM is managing the distributed disk pool. It is important to note that this replication technique produces the highest storage consumption.

Replication Techniques in a Hybrid Cloud

The aim of this test was to evaluate the behavior of the IDA replication technique implemented in a hybrid cloud (accessing both private and public cloud infrastructures). In this context, we generated fewer requests than we originally planned because of restrictions made by the public storage providers. It is important to note that the IDA technique could be attractive in hybrid cloud storage. The IDA offers data availability, fault tolerance, and a certain level of privacy because it does not require a copy of a complete file to be sent to the public cloud storage. In this context, we compared the response and service times during the file uploading and downloading processes. The performance of the version of the IDA implemented in the private cloud is taken as a reference point. We compared the private version two IDA versions that access each public cloud storage provider, Dropbox and Phoenix (TreeStore). The left side of figure 4 shows the response and service time during the uploading process. The figure shows that the IDA suffers a high penalty when accessing external storage (up to ten orders of magnitude). Even when the downloading process (right side of figure 4) showed better performance, the response time of the IDA is still penalized when accessing external storage in range of six or seven orders of magnitude. This penalty on the IDA version in a hybrid environment mainly is caused, we think, by a poor Internet connection (it is not a dedicated link) used to send and receive file fragments from the external infrastructure (storage providers). It is worth keeping in mind that one of the benefits of storing some file fragments in the external infrastructure is having more storage space available in the private cloud. It also is important to remember that, for security reasons, the number of fragments sent to the public infrastructure will never be greater than or equal to *m*, where *m* is the number of pieces required to build the original file.

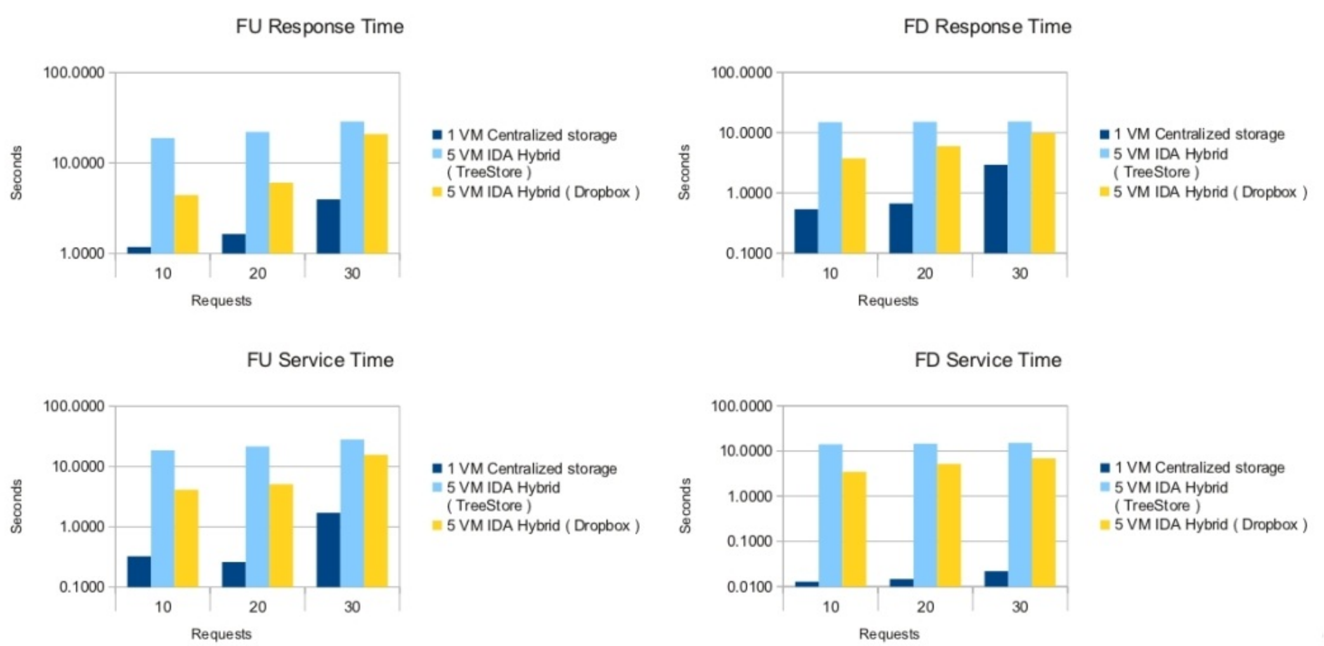

Figure 4. Average Response and Service Time for File Uploading (FU) and Downloading (FD) in the Evaluation of the Hybrid Cloud

For testing the behavior of this version of the IDA, DAM always had to obtain a fragment of a file from the public cloud (external providers). It should be noted that this is not the typical case; in a real scenario, the hybrid version of the IDA only would obtain a fragment of a file from the public cloud when the DAM couldn't obtain the *m* needed fragments from the private cloud, which means that more than *n*-*m* disks had failed (worst case). The two public storage providers showed similar performance. However, the behavior of Dropbox was slightly better than TreeStore. This could be because of the maturity of the Dropbox API or because of a better network connection to the Dropbox sites.

RELATED WORK

Amazon S3 is considered a pioneer of cloud storage solutions. Data storage rates vary according to the amount of data stored and on the availability required by users. Data availability relates to the replication technique the Amazon infrastructure uses.[10]

There also are solutions that take advantage of public cloud storage using replication techniques based on RAID, for example Redundant Array of Cloud Storage (RACS), a proxy located between multiple cloud storage providers and customers.[11] RACS distributes data in a way that provides an opportunity for clients to tolerate interruptions in a public cloud storage service or when the price for using such services gets too high. It uses replication to support such sit uations. RACS offers to its users an interface similar to Amazon S3, allowing operations such as Put, Get, Delete, and List. Another such service is High-Availability and Integrity Layer (HAIL), a cryptographic distributed system that allows file servers to provide a secure storage environment.[12] HAIL supports the failure of any of the servers that make up the system, adding a degree of security to stored data using an approach based on the Reed Solomon (RS) error correction codes.[13] The RS codes describe a systematic way of building codes that could detect and correct multiple random symbol errors by adding additional check symbols to the data.

Public cloud-storage infrastructures such as Amazon 3, Rackspace, and Google Storage are being used by distributed file systems such as Dropbox (http://www.dropbox.com), Wuala (http://www.wuala.com), and ADrive (http://www.adrive.com), which allow users to store and share files through web applications.[14] A commonality between these infrastructures and applications is their use of public clouds. These services are helpful for users wanting to have unlimited storage space with which to back up their data. However, the use of these types of solutions can be challenging in a business environment. The fears that some organizations have about storing sensitive data in a public infrastructure or about future data availability are issues that discourage the use of such third-party infrastructure.

Our approach suggests a viable option is creating a hybrid cloud storage environment (private + public) based on low cost infrastructure in which only part of the stored data are in the public environment, thereby minimizing the likelihood of unauthorized access.

CONCLUSIONS

Digital-data preservation represents a threat to digital libraries. Data are an essential part of a library, and its storage is of the utmost importance. Digital-data storage requires extreme durability and scalability. However, component failures, obsolescence, human-operation errors, natural disasters, attacks, or management errors are some common difficulties that must be carefully studied when implementing a digital library. These threats may be minimized using a distributed data-storage approach. In this area, cloud computing may help, as both the storage and the services are completely distributed. This paper presented a comparison of different replication techniques implemented in a private and hybrid cloud-storage infrastructure. We described the components of this infrastructure and demonstrated that it is possible to improve the time of system deployment and performance when elastic services (virtualized) are implemented on physical machines. We illustrated how to optimize the use of physical machine resources, especially when running systems (like the storage service) with an unpredictable workload. The replication techniques evaluated in this paper were implemented in a data access module called DAM. DAM is a simple mechanism for storage consolidation on a private and hybrid cloud environment, and it is able to offer different levels of data availability based on user requirements. It uses a lightweight algorithm for file allocation, reducing the amount of metadata needed with low resources consumption. We showed how a hybrid cloud environment, implemented with freely available software tools, can be a good solution for those institutions not confident storing sensitive data in public storage clouds and those institutions having economic and technical limitations for building their own private cloud. The prototype described in this paper showed how feasible it is to build a modest private cloud and combine it with a consolidated public cloud. In this context, this paper showed how the use of a replication technique based on an IDA has the benefits of the public cloud storage without exposing the complete content of files via a third-party infrastructure.

REFERENCES

1. John F. Gantz et al., "The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010," An IDC White Paper—sponsored by EMC, March 2007, http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf (accessed September 12, 2012).

2. 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, doi: 10.1080/01930826.2011.531637.

3. Przemysław Skibiński and Jakub Swacha, "The Efficient Storage of Text Documents in Digital Libraries," *Information Technology & Libraries* 28, no. 3 (2009): 143–53.

4. Yan Han, "On the Clouds: A New Way of Computing," *Information Technology & Libraries* 29, no. 2 (2010): 87–92; Tom Ipri, "Where the Cloud Meets the Commons," *Journal of Web Librarianship* 5, no. 2 (2011): 132–41, doi: 10.1080/19322909.2011.573295.

5. Kernel Based Virtual Machine homepage, http://www.linux-kvm.org (accessed September 12, 2012); Paul Barham et al., "Xen and the Art of Virtualization," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles*(New York: ACM, 2003), doi: 10.1145/1165389.945462.

6. "XenVsKVM," Linux Virtualization Wiki, last updated August 16, 2008, http://virt.kernelnewbies.org/XenVsKVM (accessed September 12, 2012).

7. Michael O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," *Journal of the ACM* 36, no. 2 (April 1989): 335–48, doi: 10.1145/62044.62050.

8. David A. Patterson, Garth Gibson, and Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data* (New York: ACM, 2008), 109–16, doi: 10.1145/50202.50214.

9. "Features—Simplify Your Life," Dropbox, http://www.dropbox.com/features (accessed September 12, 2012).

10. Jose L. Gonzalez and Ricardo Marcelin-Jimenez, "Phoenix: A Fault-Tolerant Distributed Web Storage Based on URLs," in *Proceedings of the IEEE 9th International Symposium on Parallel and Distributed Processing with Applications* (ISPA, 2011): 282–87, doi: 10.1109/ISPA.2011.33.

11. "Autobench," Xenoclast, last updated March 31, 2012, http://www.xenoclast.org/autobench (accessed September 12, 2012).

12. Hussam Abu-Libdeh, Lonnie Princehouse, and Hakim Weatherspoon, "RACS: A Case for Cloud Storage Diversity," in *Proceedings of the 1st ACM Symposium on Cloud Computing* (New York: ACM, 2010), 229–40, doi: 10.1145/1807128.1807165.

13. Kevin D. Bowers, Ari Juels, and Alina Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," in *Proceedings of the 16th ACM Conference on Computer and Communications Security* (New York: ACM, 2009), 187–98, doi: 10.1145/1653662.1653686.

14. I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields*," Journal of the Society for Industrial & Applied Mathematics* 8, no. 2 (1960): 300–304.