A Comparison of Redundancy Techniques for Private and Hybrid Cloud Storage

E.M. Hernandez-Ramirez*[1], V.J. Sosa-Sosa[2], I. Lopez-Arevalo[3]

[1,2,3] Laboratorio de Tecnologías de Información (Campus Tamaulipas)
del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV)
Cd. Victoria, Tamaulipas, México
*emhr1983@gmail.com

ABSTRACT
File redundancy techniques have been very useful mechanisms for offering fault tolerance and data availability in any kind of storage. Cloud storage is not the exception. This paper presents an evaluation of classical file redundancy techniques implemented in two cloud-storage deployment models, private and hybrid. A small prototype of a private and hybrid cloud storage was implemented for this evaluation. The performance impact when file redundancy is only applied in a private cloud versus when redundancy is also distributed in a public cloud (the hybrid model) is analyzed. Additional to classical file redundancy techniques, an innovative method was evaluated for file redundancy based on an information dispersal algorithm (IDA). The usage of IDA represents a good option for managing sensitive data in hybrid cloud storage. In this technique, only parts of a file need to be sent to the public cloud, avoiding the complete file to be read from outside of the private zone. In this context, there is a trade-off between performance (for reconstructing the original file, it is first necessary to obtain all of its fragments) and the security level that could determine the viability of using IDA.

Keywords: cloud computing, data management, virtualization.

RESUMEN
Las técnicas de redundancia de archivos han sido mecanismos muy útiles para ofrecer tolerancia a fallos y disponibilidad de datos en cualquier tipo de almacenamiento. El almacenamiento en la nube no es la excepción. Este trabajo presenta una evaluación del comportamiento de técnicas clásicas de redundancia de archivos, implementadas en una nube de almacenamiento utilizando dos modelos de despliegue, privado e híbrido. Para esta evaluación se desarrolló un prototipo de un sistema de almacenamiento en la nube que sigue los modelos de despliegue antes mencionados. Se compara el impacto en el rendimiento cuando se aplica redundancia sólo en una nube privada versus cuando la redundancia también se distribuye en una nube pública (modelo híbrido). Adicionalmente, se evaluó un método innovador para la redundancia en archivos basado en el Algoritmo de Dispersión de Información (IDA). El uso del IDA surge como una buena opción para la administración de datos sensibles en una nube de almacenamiento híbrida. Con esta técnica de replicación, sólo fragmentos de un archivo serán enviados a la nube pública (infraestructura de terceros), evitando que el archivo completo pueda ser leído desde afuera de la zona privada. En este contexto, existe un compromiso entre el desempeño (para reconstruir el archivo original es necesario obtener primero sus fragmentos) y el nivel de seguridad que determinará la viabilidad de usar el IDA.

1. Introduction

To define the storage requirements for institutions or companies of any size has become a problem with no trivial solutions. It is mainly due to the very fast generation of digital information whose behavior is very dynamic [1].

In this context, it is common for managers of storage resources, with the responsibility to make predictions about the resources that will be needed in the medium term, to often face the following scenarios:

a) Predictions are below real needs. In this case, there will be a problem of resource deficit.

b) Excessive expenditure on the purchase of storage resources, which can produce a complex

administration, probably with resources that will not be used in the medium term.

In this situation, the acquisition of storage services that implement an *elastic* concept becomes attractive, i.e., storage capacity that can be increased or reduced on demand, with a cost of acquisition and management relatively low.
Nowadays, this service model is called cloud computing. 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 in basically three ways: private, public or hybrid. In the private option, the infrastructure is operated solely for a single organization; most of the time, it implies an initial strong investment because it is necessary for the organization to purchase a big amount of storage resources and pay for the administration costs. The public cloud is the most traditional version of cloud computing. In this model, the 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 both. A cloud computing environment is mainly supported by different technologies such as virtualization and service-oriented architectures.

A cloud environment provides omnipresence and facilitates deployment to file storage services. It means that users can access their files from anywhere, while there exists an Internet connection and without requiring the installation of a special application (only a web browser is needed).

Data availability, scalability, elastic service and pay only for consumption are very attractive characteristics found in the cloud service model. Virtualization plays a very 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, software migration, among others. Besides virtualization technologies, emerging tools that allow the creation of cloud computing environments, providing dynamic instantiation and release of virtual machines and software migration are also supporting the elastic service offered in this kind of computing model.

Currently, it is possible to find several proposals for public cloud storage such as Amazon S3 [2], RackSpace [3], or Google Storage [4], which provide high availability, fault tolerance and services and administration at low cost. However, there still exist companies that do not feel confident about storing their information in a third-party-owned environment. In these cases, such companies wanting to take advantages of the cloud computing facilities would require to implement a private cloud solution. Unfortunately, this option is often beyond their budgets. In this case, a hybrid cloud model could be an affordable solution. Companies or users in general can store sensitive or most 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) helps us to analyze the behavior of different redundancy techniques, paying special attention to the low cost of the system implementation, the system efficiency, resource consumption and the different levels of data privacy and availability that can be reached by a system like this.

This paper is organized as follows: Section 2 introduces a proposal for a cloud computing infrastructure based on free open source software (FOSS). It also describes the redundancy techniques that were implemented for this comparison. Section 3 presents the evaluation scenario and a performance analysis considering several aspects such as: the impact of having an elastic storage service, the implementation of different redundancy techniques in both private and hybrid cloud computing environments. Section 4 includes the related work, and finally Section 5 offers some important remarks and conclusions.

2. Infrastructure description

Nowadays, small and medium businesses (SMB) are facing economical and technical challenges that arise when trying to obtain the benefits of having their own cloud computing environment

(private). The aim of this proposal is to help with those challenges by designing and implementing a scalable and elastic distributed storage architecture based on free and well-known open source tools. This architecture combines private and public clouds by creating a hybrid cloud environment. For this purpose, tools such as KVM [5] and XEN [6] were evaluated, which are useful for creating virtual machines (VM). OpenNebula [7], Eucalytus [8] and OpenStack [9] are good free options for managing a cloud environment. OpenNebula was the selection for this prototype because there is enough available information online that does not require a strong technical background.

The hard disks (HDs) integrated into the storage infrastructure are found in commercial computers (commodities). The use of this type of HDs makes this architecture failure-prone. This situation was the main motivation to evaluate different redundancy 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 (b) a distributed storage web application (DISOC) that is used as a proof of concept. It can also be observed that the private cloud has an interface to access a public cloud creating a hybrid environment.

The core components of the architecture are the following:

• Virtual Machine (VM). Different open source tools were evaluated, such as KVM [10] and XEN [6], for the creation of virtual machines. Some performance tests were done, it was found that KVM showed a slightly higher performance than XEN (similar results can be found at [10]).

• Virtual Machine Manager Module (VMMM). It has the function of dynamic instantiation and de-instantiation of virtual machine 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 use of the Data Access Module Interface (DAM-I). DAM-I allows VMs to get access to disk space by calling the Data Access Module (DAM), which provides transparent access to the different disks that are part of the storage infrastructure. It allocates and retrieves individual files stored on different file servers.

• Load Balancer Module (LBM). It is designed to distribute the load among different VMs instantiated on the physical servers that make up the private cloud.

• Load Manager (LM). It is responsible for monitoring the load that can occur in the private cloud.

• Distributed Storage on the Cloud (DISOC). It is a web-based file storage system that is used as a proof of concept and was implemented based on the proposed architecture.

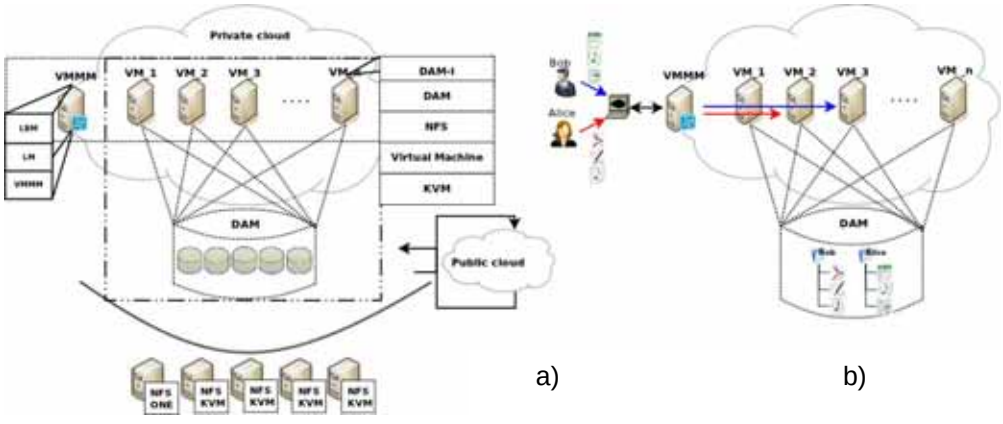

Figure 1. Main components of the cloud storage architecture.

2.1 Redundancy techniques

High availability is one of the important features offered in a storage service deployed in the cloud. To accomplish it, the use of redundancy techniques has been the most useful proposal [19][20]. DAM is the component that is configured to provide different levels of data availability. It currently includes the following redundancy policies: no-replication, total-replication, mirroring and IDA-based redundancy.

• No-Replication. This redundancy 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, which depends on disk availability. This policy prevents all files to be allocated in a single server, providing a minimal fault tolerance.

• Mirroring. This redundancy technique is a simple way to ensure higher availability, without high resource consumption. In this redundancy, every time a file is stored in a disk, DAM creates a copy and places it on a different disk following a round robin policy.

• Total-Replication. It represents the highest data availability approach. In this technique, a copy of the file is stored in all of the file servers available. It is also the strategy that requires the highest consumption of resources.

• IDA-based redundancy. In order to provide better 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) [11] is an example of this strategy. When a file (of size |F|) is required to be stored using IDA, the file is broken into n pieces of size |F|/m, where m<n. These pieces are distributed in n different disks. IDA only needs to obtain m pieces 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 server fail. IDA provides better fault tolerance than mirroring without needing to totally replicate the original file. In this prototype, IDA was evaluated with n = 5 and m = 3 (it means only a 60% of the original file will be replicated). IDA is very attractive for being used in a hybrid cloud environment, since 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.

3 Perfomance evaluation

A prototype of this architecture was implemented and used as the evaluation scenario. It includes 5 commercial PCs (commodities) whose characteristics are shown in the first section of Table 1. The features of the VMs that were instantiated on the mentioned 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.

A total of 9 VMs were created in a private cloud environment for this evaluation. In order to build and test a hybrid cloud environment, it was necessary to access a public storage cloud (third-party infrastructure). Two different public storage providers were used in this experiment, Dropbox [12] and Phoenix (also known as TreeStore) [13]. These 2 storage services were chosen because they have a free service version and provide a simple application interfaces (API) for third-party developers. Our Data Access Module (DAM) was also responsible for offering a transparent access to the external storage infrastructure. It was required to send a valid user and password in both providers. Additionally, for accessing Dropbox, it is also necessary to obtain a key for developers. This key is required by the Dropbox API. It is important to say that Dropbox is also able to keep files in the Amazon S3 storage infrastructure [2].

Different workloads were emulated, 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 take it as an attack against their servers and, as a consequence, to block the service. In the hybrid configuration test, the numbers of parallel requests sent to the public storage were 10, 20 and 30. This private and hybrid (private + public) cloud storage scenario was designed to evaluate the following: a) the impact of having an elastic service and, b) the behavior of the cloud storage infrastructure when applying different redundancy techniques in order to offer several levels of data availability.

3.1The impact of having an elastic service

As a first step, the impact of having elasticity in the storage service was evaluated compared to a static service (without elasticity). In the elastic service, a new virtual machine is instantiated when a workload exceeds a defined threshold. The evaluation uses different workloads generated by Autobench [14]. A physical machine with a single hard disk receiving an increasing workload was compared by applying the same workload on a set of virtual machines that are incrementally instantiated in the same physical machine. For this test, the workload basically consisted of a set of requests of a dynamically generated PHP web page. This web page emulates the processing time on a server by means of running a sorting algorithm (bubble type). Trying to emulate different levels of load on the server, it was defined a list containing different quantities of elements that had to be sorted. The results shown in Figure 2 represent the average response time a customer received when the load balancer only accessed to one physical machine (fixed line), and when the balancer accessed the same physical machine using from 1 to 3 VM instantiations (elastic line). It can be seen, when the workload is low, at the beginning of the test, how the response time offered by the static service (running only on one physical machine) is better, in some cases up to 4 or 5 orders of magnitude, compared to that obtained in the execution of the service accessing to one virtual machine.

In this test, a maximum response time of 25 s was defined as the upper threshold for a new VM instantiation. It means that when the global system response time reaches 25 seconds, a new virtual machine will be instantiated and integrated into the storage service. It also can be seen that the response time in the elastic service has some considerable falls during the test. This behavior is not occurring at the time of a new VM is instantiated, but at the time when the VM is included in the service by the load balancer. The instantiation and activation time of the new VM was between 60 and 90 seconds. When the workload increases, it is necessary to instantiate another VM. For this test, the elastic service was able to finish the workload offering an acceptable response time, while the static service collapsed and could not finish all of the requests sent by the client. Likewise, when the response time goes below the threshold and keeps for a while, a VM is released. This descending activity is monitored until a single VM is running on the entire infrastructure.

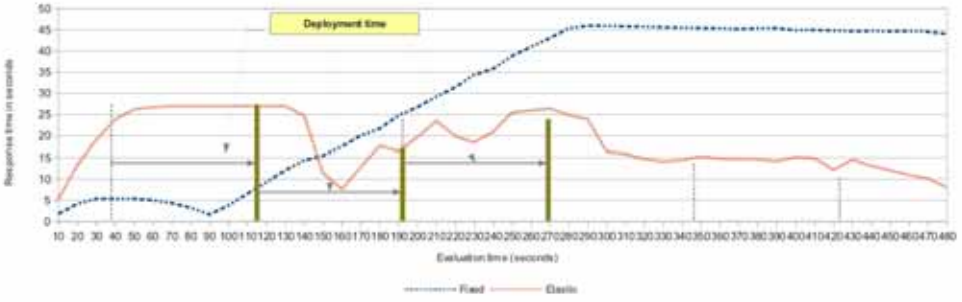

Figure 2. Performance comparison between a fixed and elastic storage service.

3.2 Evaluation of different redundancy techniques

With the DAM component is possible to define the level of data availability required in the cloud storage prototype. It can be done by applying different redundancy techniques. In this test, it was defined a benchmark that allows seeing the benefits obtained of using a distributed storage system compared with a centralized version. In the first test, DAM was configured for having access to a single disk using only one VM (emulating a centralized process) with a single file server (emulating centralized storage).

In the rest of the tests, it was always considered a distributed process (8 VMs) using a distributed storage system (5 disks that were distributed on different storage servers encapsulated by DAM). Since the redundancy 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 also accessing a public storage cloud (hybrid model). Two main metrics were taken into account for these experiments: 1) Response time: it considers the time from when the user clicks on the button to upload or download a file until the point when the file loading or downloading has finished, in this test until the TCP connection is closed down. 2) Service time: the time needed by DAM for locating a file (or part of it) and getting the file ready to be read by the system component that is requesting it.

3.2.1 Redundancy techniques in a private cloud

This test evaluated the response and service time perceived by users that requested different levels of data availability and fault tolerance to the storage prototype, when running on a private cloud environment. Different redundancy techniques (see Section 2.1) were implemented in MAD to carry out this aim.

The left side of Figure 3 shows the response and service time produced by different redundancy techniques during the file uploading process. In this case, even though the worst service time was produced by the total redundancy technique, it can be seen that users perceived the worst performance when they were accessing a centralized storage service. It is interesting to see how the no-replication technique is showing the best performance during the uploading process.

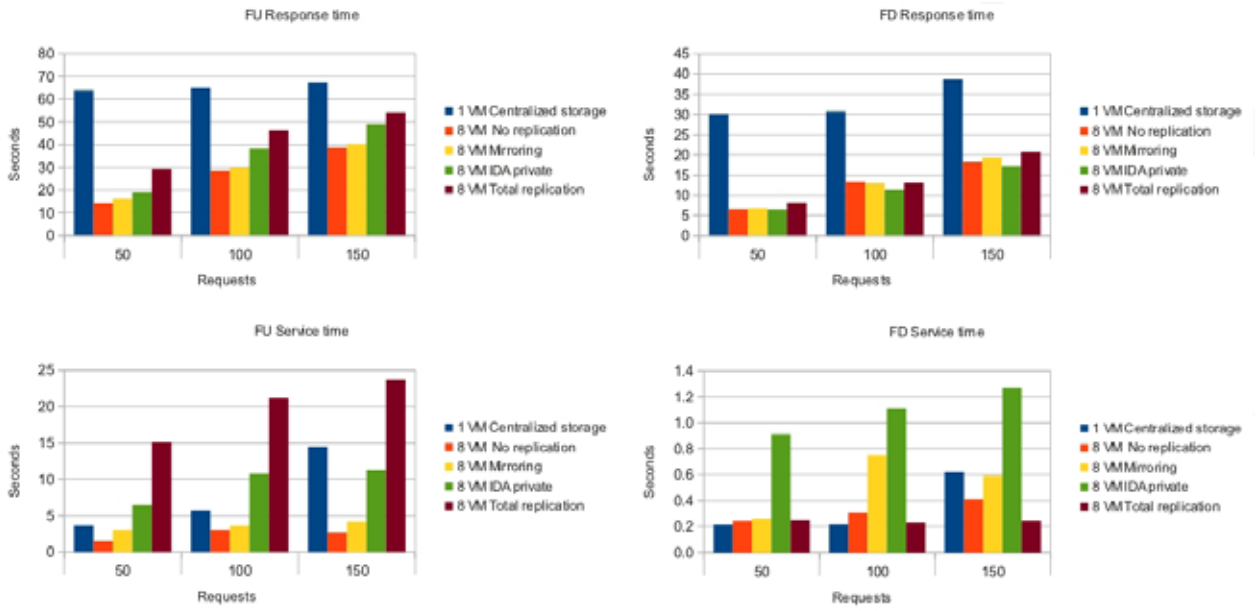

Figure 3. Average response and service time for file uploading (FU) and downloading (FD) using different redundancy techniques in a private cloud environment.

This behavior could be caused 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. 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 redundancy techniques was very similar.

IDA shows a very competitive response time and offers an acceptable level of fault tolerance. The total redundancy technique offers high data availability and fault tolerance, but it is not producing the best response time. It could be caused by the way DAM is managing the distributed disk pool. It is important to note that this redundancy technique produces the highest storage consumption.

3.2.2 Redundancy techniques in a hybrid cloud

The aim of this test was to evaluate the behavior of the IDA redundancy technique implemented in a hybrid cloud (accessing the private and public cloud infrastructures). In this context, a reduced number of requests was generated because of restrictions given by the public storage providers. It is important to note that the IDA technique could be attractive in hybrid cloud storage. IDA offers data availability, fault tolerance and a certain level of privacy, since it does not require a copy of a complete file to be sent to the public cloud storage. In this context, the response and service time perceived by users during the file uploading and downloading processes were compared. The performance of the version of IDA implemented in the private cloud is taken as a reference point. The private version is compared to 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. It can be seen how IDA suffers a high penalty when accessing the external storage (until 10 orders or magnitude). Even when the downloading process (right side of Figure 4) showed a better performance, the response time of IDA is still penalized when accessing the external storage in 6 or 7 orders of magnitude. This penalty on the IDA version on a hybrid environment is given mainly by the poor internet connection (it is not a dedicated link) used to send/receive file fragments from the external infrastructure (storage providers).

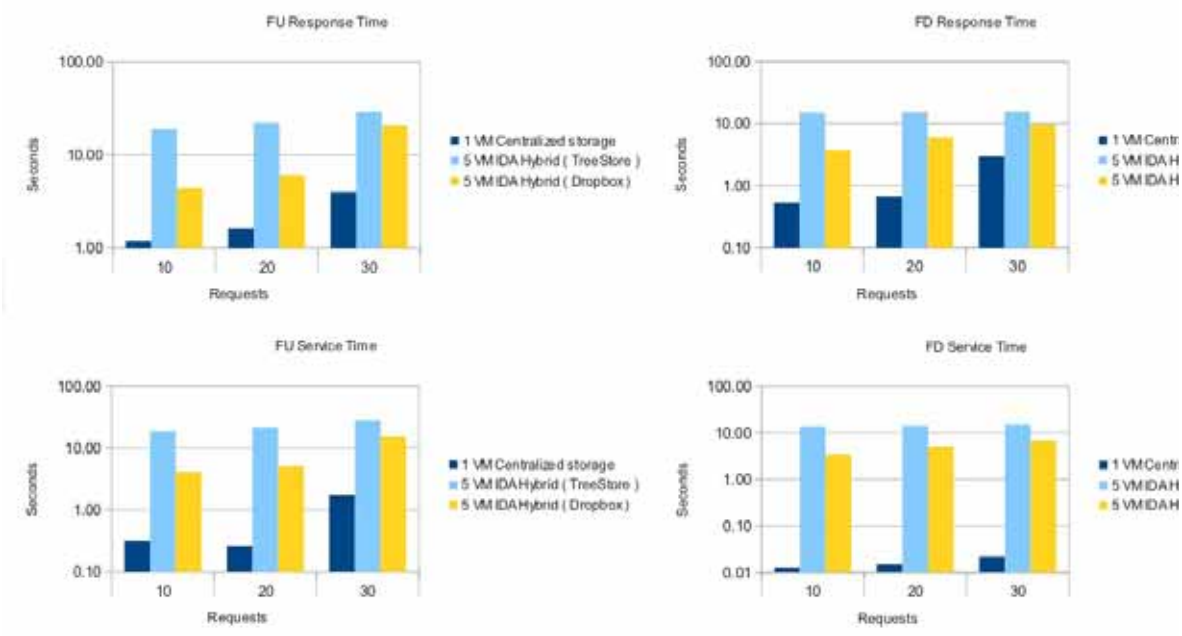

Figure 4. Average response and service time for file uploading (FU) and downloading (FD) in the evaluation of the hybrid cloud.

It is worth keeping in mind that one of the main benefits of storing some file fragments in the external infrastructure is the fact of having more storage space available in the private cloud. It is also important to remember that, for security reasons, the number of fragments that are 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. For testing the behavior of this version of IDA, DAM was forced to always obtain a fragment of a file from the public cloud (external providers). It should be noted that this is not the typical case, because in a real scenario, the hybrid version of IDA only would obtain a fragment of a file from the public cloud in the cases when it was not able for DAM to 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 a similar performance. However, the behavior of DropBox was slightly better than Treestore. It could be due to the maturity of the Dropbox API or a better network connection to the Dropbox sites.

4. Related work

Nowadays, Amazon S3 is considered a pioneer of cloud storage solutions. It offers to its users different storage rates, according to the amount of stored data. These rates vary depending on the data availability required by users. Data availability is related to the redundancy technique that will be used in the Amazon infrastructure [2].
There exist also solutions that take advantage of public cloud storage using redundancy techniques that were originated in RAID, for example RACS [15], which is a proxy that is located between multiple cloud storage providers and customers. It is responsible for distributing data in a way that it provides an opportunity for clients to tolerate interruptions in a public cloud storage service or when the price for using the services is getting high. It uses redundancy in order to support those possible situations. RACS offers to its users an interface similar to Amazon S3, allowing operations such as PUT, GET, DELETE and LIST. Another proposal is HAIL [16], a cryptographic distributed system that allows file servers to provide a secure storage environment. 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 error correction codes.

Similarly, public cloud storage infrastructures such as Amazon S3 [2], Rackspace [3], Google Storage [4] are being used by distributed file systems such as Dropbox [12], Wala [17], and ADrive [18] that allow users to store and share files through web applications.

A common point in these infrastructures and applications is the use of public clouds. These services are being very useful for users wanting to have an unlimited storage space or to backup their data. However, the use of this type of solutions can be a challenging decision for a business environment. The fears that some organizations have about storing sensitive data in a public infrastructure or that the data could not be available at the time they are required are issues that discourage the use of third-party infrastructure.

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

5. Conclusions

This paper presented a comparison of different redundancy techniques that were implemented in a private and hybrid cloud storage infrastructure. A description of the components of this infrastructure was made. It was demonstrated that it is possible to improve the time of system deployment and performance when an elastic services (virtualized) is implemented on physical machines. The use of the physical machines resources will be optimized, especially when they are running systems (like the storage service) with an unpredictable workload. The redundancy techniques evaluated in this paper were implemented in a data access module named (DAM). It is a simple mechanism for storage consolidation on a private and hybrid cloud environment, which is able to offer different levels of data availability based on user requirements. DAM uses a lightweight algorithm for file allocation, reducing the amount of metadata needed with low resources consumption. It is shown how hybrid cloud environment, implemented with free

available software tools, can be a good solution for those institutions that are not confident of storing sensible data in public storage clouds, and have economic and technical limitations for building their own big private cloud. The prototype described in this paper show how feasible 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 redundancy technique based on an information dispersal algorithm (IDA) allows obtaining the benefits of the public cloud storage without exposing the complete content of their files in 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 (online). Available from: http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf

[2] M. R. Palankar et al, "Amazon S3 for science grids: a viable solution?" DADC08, Proceedings of the 2008 international workshop on Data-aware distributed computing, pp. 55-64. June 2008.

[3] Rackspace Cloud Files (online). Available from: http://www.rackspace.com/cloud/cloudhostingproducts/files

[4] Jeffrey Dean, "Evolution and future directions of large-scale storage and computation systems at Google", SoCC10, Proceedings of the 1st ACM symposium on Cloud computing, pp. 1-1 2010.

[5] Irfan Habib, "Virtualization with KVM", Linux Journal. Volume 2008 Issue 166, Article No. 8. Feb. 2008.

[6] P. Barham et al, "Xen and the art of virtualization", SOSP 2003 Proceedings of the 19th ACM symposium on Operating systems principles, pp. 164-177. 2003.

[7] B. Sotomayor et al, "Virtual Infrastructure Management in Private and Hybrid Clouds", IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sep./Oct. 2009.

[8] Bill Childers, "Build your own cloud with Eucalyptus", Linux Journal, volume 2010 issue 195, Article No. 1. July 2010.

[9] Openstack homepage (online). Available from: http://www.openstack.org/

[10] Comparative of XEN and KVM (online). Available from: http://virt.kernelnewbies.org/XenVsKVM

[11] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance", J. ACM 36, 2 (April 1989), pp 335-348.

[12] Dropbox (online). Available from: http://www.dropbox.com/features

[13] J. L. González and R. Marcelín Jiménez. "Phoenix: Fault Tolerant Distributed Web Storage". FTRA, Journal of Convergence of the Future Technology Research Association, vol 2, no.1. 2011 pp. 79-86.

[14] Autobench (online). Available from: http://www.xenoclast.org/autobench

[15] Abu-Libdeh, H et al., RACS: a case for cloud storage diversity, Proceedings of the 1st ACM Symposium on Cloud Computing. Pp. 229-240. June 2010.

[16] Bowers K. D. et al., "HAIL: a high-availability and integrity layer for cloud storage", Proceedings of the 16th ACM Conference on Computer and Communications Security. pp. 187-198. November 2009.

[17] Wala, Secure online storage (online). Available from: http://www.wuala.com

[18] ADrive, Web storage (online). Available from: http://www.adrive.com

[19] M. Quezada Naquid et al, "Fault Tolerance and Load Balance Tradeoff in a Distributed Storage System", Computación y Sistemas, vol. 14, no. 2, October-December 2010.

[20] S. Siva Sathya et al, "Replication Strategies for Data Grids", ADCOM 2006, Proceedings on Advanced Computing and Communications, pp. 123-128. Dec. 2006.