Comparison of Performance, Availability and Scalability of a Private Cloud Storage Infrastructure Service for Document Preservation

Antonio M. A. Ferreira
Computer Science Department,
University of Brasília (UnB), Brazil
amarcos.aferreira@gmail.com

Aleteia P. F. Araújo
Computer Science Department,
University of Brasília (UnB), Brazil
aleteia.araujo@gmail.com

André C. Drummond
Computer Science Department,
University of Brasília (UnB), Brazil
andred@cic.unb.br

Abstract— Cloud computing platforms provide easy and transparent access to a company's storage infrastructure through services. it allows users to store their data at remote disks and access them anytime from any place. On the other hand, with the deployment of Electronic Document Management (EDM) to manage the insertion, sharing, and retrieval of information assigned to a digital document it is necessary that the processes of searching, indexing and storage of digital documents are carried out efficiently. Besides the use of computational tools focused on archiving digital documents, the storage solution must provide reliability and authenticity on the availability of digital documents. With object storage systems being increasingly accepted as replacements for traditional file or block systems and recognized as a preferred way to expose one's storage infrastructure, it is important to effectively measure the performance of these services. This paper addresses the deployment project of a content-addressable storage infrastructure in a large company and conducts performance, availability and scalability evaluation to measure the gains in r/w access of the proposed solution in different scenarios proposed.

Index Terms— cloud computing; cloud storage; content addressable storage; deduplication; electronic document management

I. INTRODUCTION

The technological advancements in the last decade in different computational areas such as the Internet, programming languages, and on-demand services, have generated a need for storage systems for large amounts of digital data that support high performance, redundancy and scalability [1].

Currently it is possible to have access to information and digital documents regardless of their geographical location and they can be preserved for longer periods of time when compared to standard media. According to [2], "the problem of digital preservation presents itself as a real problem to be solved by the institutions, especially those that have a legal obligation to maintain long-term document".

The Directed Attached Storage (DAS) solution, widely used worldwide, has physical limitations due to a low capacity to support the expansion of storage drives, and presents fundamental flaws in terms of access security. Such technology should therefore not be used in production environments, especially those institutions that deal with classified documents.

The digital preservation must ensure that the content in digital format remain accessible over time and should provide reliability (reliability of a document as proof of what it is) and authenticity (trustworthiness over time) [3], factors that are related to digital document storage.

Cloud Computing portends a major change in how to store information and run applications. Instead of running programs and storing data on an individual desktop computer, everything is hosted in the cloud. Cloud computing lets you access all your applications and documents from anywhere in the world, freeing them from the edge of a physical computer a making it easier for group members in different locations to collaborate [4].

Cloud computing is a resource delivery and usage model to get resources (hardware, software, applications) through on-demand and at scale network as services in a shared, multi- tenant and elastic environment [5] and [6]. Computational capabilities in clouds rely in the concept of virtualization, which is based on several virtual machines providing services to consumers. Cloud computing aims to offer virtually unrestricted pay-per-use computational resources without the need to manage the underlying infrastructure.

This paper proposes the implementation of a system of content addressable storage (CAS) distributed with file-level deduplication, providing reliability and authenticity of stored digital documents, operating as a service in a private cloud, which is a cloud computing platform implemented within the corporate firewall, under the control of the organization.

The proposed solution is evaluated in a real-world scenario, the Court of Auditors of the State of Tocantins, Brazil (TCE-TO), and presents the gains related to the overall performance of the infrastructure, the expansion of the availability of the data and the scalability due to replication of objects to multiple CAS servers. The results obtained at the end of this study demonstrate that the processes of searching, indexing and storing digital documents are realized with more effectiveness. For a benchmark, the results obtained are also compared with traditional block storage infrastructures.

II. CLOUD STORAGE

Cloud computing is typically defined as a specialized form of distributed computing that introduces utilization models for provisioning scalable and measured resources [7]. One of the primary uses of cloud computing is data storage. With cloud storage, data is stored on multiple third-party servers, rather than on the dedicated servers used in traditional networked data storage. STaaS (Storage as a Service) is a term used to describe a storage model where a business or organization (the client) rents or leases storage space from providers.

The deployment model refers to the location and management of the cloud infrastructure, which provides different levels of access and services. The four implementation models are categorized into public, private, hybrid, and community cloud. The private cloud model describes that a private cloud infrastructure is for the exclusive use of a company or person [8].

Storage as a Service has become popular because there is usually no start-up costs (e.g., servers, hard disks, IT staff and so on) involved. Businesses pay for the service based only on the amount of storage space used.

When storing data, the user sees a virtual server – that is, it appears as if the data is stored in a particular place with a specific name. But those place doesn't exist in reality. It's just a pseudonym used to reference virtual space carved out of the cloud. In reality, the user's data could be stored on any one or more of the computers used to create the cloud. Some key benefits of using cloud storage are [4]:

- Ease of management: the maintenance of the software, hardware and general infrastructure to support storage is drastically simplified by an application in the cloud.
- Cost effectiveness: for total cost of ownership, cloud storage is a clear winner. Elimination of the costly systems and the people required to maintain them typically provides organizations with significant cost saving that more than offset the fees for cloud storage
- Lower impact outages and upgrades: typically cloud computing provides cost effective redundancies in storage hardware. This translates into uninterrupted service during a planned or unplanned outage. This is also true for hardware upgrades which for the end user will no longer be visible.
- Disaster preparedness: off site storage isn't new. Cloud storage services not only keep your data off premise, but they also make their living at ensuring that they have redundancy and systems in place for disaster recovery.
- Simplified planning: cloud-based solutions are flexible and provide storage as needed. This eliminates the need to over provision for storage that may be needed to meet.

So, in general, cloud storage has both financial and security-associated advantages. Financially, virtual resources in the cloud are typically cheaper than dedicated physical resources connected to a personal computer or network. As for security, data stored in the cloud is secure from accidental erasure or hardware crashes, because it is duplicated across multiple physical machines; since multiple copies of the data are kept continually, the cloud continues to function as normal even if one or more machines go offline. If one machine crashes, the data is duplicated on other machines in the cloud.

A. Content Addressable Storage

In [9], CAS is defined as a storage device based in object, which has a unique address known as content address (CA - Content Address). Unlike location-based addresses, the addresses of content, once calculated, are not changed and always refer to the same content. In the event of a change in the contents of an object, a new CA is calculated and assigned to the new content. With this technology, it is straightforward to check if the data returned for a given document is exactly the same as the data that has been stored (authenticity) using the name or UUID (Universally Unique Identifier). The need for applications to know and manage the physical location of the information in the storage device is eliminated.

The integrity of a message (reliability) is guaranteed through an end-to-end communication process based on the content- MD5. The integrity of a message (reliability) is carried out through from a communication end-to-end that uses the MD5 to compute the content at 128-bit and the base64 to encode the data to transfer networked.

Even if an MD5 hash attack occurs and compromises the integrity of the content, there is also the procedure of content authenticity. For each reading procedure performed, the CAS device uses an hash algorithm to recalculate the address of the content of the object and compare the result with the address of the original content, that is, it has an implicit process of validation of the stored content.

The CAS technology is proposed as the storage infrastructure due to the fact that it allows direct access to the stored information and employs file-level deduplication, preserving the documents in an immutable way with reliability and authenticity.

III. RELATED WORK

The content-addressable storage (CAS) solution has been the target of a body of work that studies the problem of improving the performance of data traffic and preservation of digital data in storage devices. The paper [10] introduces a framework directed at storing objects to keep the digital data intact for long periods of time, based on the use of a content addressable storage system that is scalable and efficient. The article also discusses the preservation of data on storage devices by checking data integrity during archiving, location and recovery.

The use of cloud storage architecture for document archiving is recommended in [11], as well as the use of document replication mechanisms among CAS servers (to minimize service unavailability and the need for daily backups, and to improve scalability in a distributed storage environment). However, it becomes necessary to assess whether the structure of CAS provides real performance gains when compared to the direct-attached storage architecture.

As well as the studies presented by [10] and [12], aproach of this dissertation is directed to the digital preservation facing its management on the principles of authenticity and integrity in a storage structure of addressable content in which it will be applied a function hashing using an MD5 algorithm to identify similarity and eliminate redundancy when storing objects.

The authors [12] show that hashing functions, such as MD5 and SHA-1, are recommended for use in systems that need to uniquely identify a file, block, or byte and thus avoid storing identical data. A comparative analysis is also performed between these two hash functions, the results of which demonstrate that MD5 provides a better calculation speed to generate the content address of each data block input and thus detect possible duplications in the system compared to SHA-1. Based on this context, the authors present a distributed deduplication storage system designed to provide gains with flow and scalability.

In [13], the authors develop a tool to assess the performance of services directed at object storage devices. It was stated that the following metrics should be considered in order to properly evaluate a solution: mean response time (elapsed time between the beginning and the completion of an operation), throughput (total number of transactions per second) and bandwidth (total amount of data transferred per second). The experimental setup consisted of 23 read transactions of 64 KB objects in 128 different containers (storage "directories") in a total time of 300 seconds in CAS servers. However, according to the authors, this tool is still in the development phase and is not available for analysis.

In [14], it is stated that the three major metrics that should be taken into account by organizations wanting to take advantage of cloud storage are: performance, availability, and scalability. It argues that an architecture aimed at cloud storage must offer “unlimited” storage capacity, as well as lay the foundation for the complete elimination of the need for daily backups through cloud data replication mechanisms

Both [13] and [14] point out that to analyze a storage structure in a private cloud, one should measure the response time, throughput and bandwidth using the following variables: concurrency (multiple concurrent connections), file size (from very small to very large) and the type of workload (reading, writing and mixed).

IV. EXPERIMENTAL ENVIRONMENT

In order to propose a storage solution that could be employed in TCE-TO, a test scenario with the same logical structure and considering the same type of devices was setup. This experimental environment consists of three data storage structures. In this environment there is one file server with block-level deduplication (block size of 4 KB), in which the “Opendedup” [15], software is used to provide a storage volume with inline deduplication (synchronous), which is responsible for the interaction between the application and the operating system through a one gigabit Ethernet interface.

As illustrated in Fig. 1, there will also be 1 file server per block with an operating system that has 1 Gigabit Ethernet interface, and 3 (IP: 192.1680.89; IP: 192.168.0.90 and IP: 192.168.0.91) CAS servers with 4 Gigabit Ethernet interfaces belonging to the server, which will be grouped for load balancing between network interfaces and fault tolerance.

All CAS servers will be associated in a cluster as “Tdinfocascluster” with the local replication procedure being handled by the network. The LAN interconnecting the web server and the benchmark computer is the same LAN that connects all the user desktops on TCE-TO, thus providing a realistic network traffic load scenario. All communication via the web is through a communication channel that uses a cryptographic function to protect the transferred data and metadata. The data storage servers are connected using the same subnet IP addresses and have direct communication with the web server, which is then connected to a computer with the benchmark software responsible for carrying out the evaluation process.

The goal of the experiment is to measure the performance, availability and scalability of the different storage structures. Table I presents the computational resources used in the testing environment.

Finally, in the actual configuration, the storage service is provided by a private cloud as a web service. Moreover, each CAS server is allocated in separate buildings (within a radius of 500 meters). Moreover, the global distribution is present in the solution by configuration, thus, the administrator determines which servers are active and able to receive the files, and these servers can be in any network, local or global. Regarding allocation on demand, it is important to note that the servers can be used by other public service agencies when not in service to the TCE-TO. When storage space is required by enterprise (i.e., demand for the storage service increases), the servers will prioritize meeting the internal demand. Therefore, in the cloud, storage space is allocated and released on-demand.

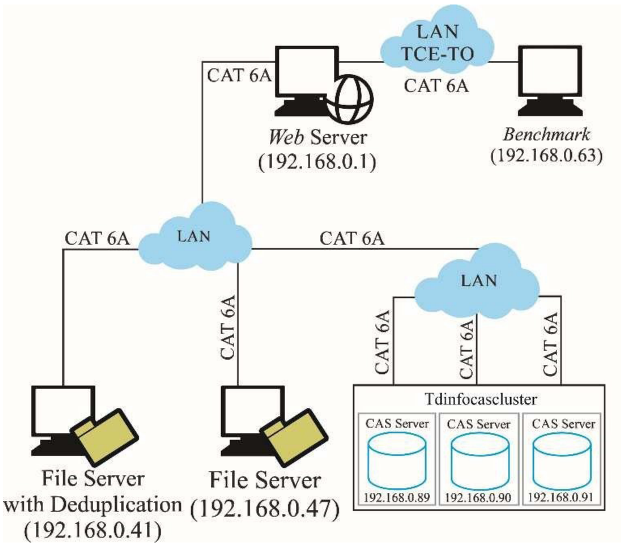

Fig. 1. Evaluation Environment.

Furthermore, there is no need to implement any pricing model because it is a private cloud belonging to the TCE-TO.

TABLE I. EQUIPMENT USED IN TESTS.

Hardware	Software / Tool
Server Dell PowerEdge R710 - Xeon(R) E5506 2.13GHz, RAM 24 GB	Apache 2.2.22 Debian 7.4
Server Dell PowerEdge R710 - Xeon(R) E5506 2.13GHz, RAM 24 GB	Opendedup 2.9.2

A. Methodology

Simulations are run in order to measure the performance, availability and scalability of all storage structures, through the software “Siege”. This software is a benchmarking tool for carrying out HTTP load tests to determine the maximum traffic load that can be served by the storage system. The command line options used with Siege were [16]:

- -d: set the delay between user requests
- -c: define the number of simultaneous users (concurrent connections)
- -t: set the total simulation time

In order to verify this proposed data storage solution, study tree different test scenarios were tested:

- Scenario 1: Evaluation of the Read and Write Procedure in a 282 KB Document.
- Scenario 2: Evaluation of the Read and Write Procedure on a 10 MB Document.
- Scenario 3: Evaluation of the Procedure of Reading and Writing in Multiple Documents.

As described in Section 3, the metrics to be measured are the response time and bandwidth while varying the concurrency (multiple simultaneous connections), file size (varying from very small to very large files) and the type of workload (reads, writes, mixed) of the storage structures.

V. EVALUATION

In this section the performance, availability, and scalability of the proposed solution are assessed. The metrics used for the evaluation of these parameters are the throughput and response time. The tests conducted covered the reading and writing of multiple documents.

Throughout this section, the term “transaction” refers to the number of operations (read/write) concluded successfully between the benchmark computer and the data storage structure. A "fault transaction" is defined as an operation that could not be completed between the benchmark computer and one of the storage servers.

The number of connections established at a certain point in time is defined as the number of users connected between the benchmark and the respective storage structure, with respect to the beginning of a test cycle with Siege. Moreover, all the concurrent connections are initialized at the same time. Finally, the data traffic limit is set to one Gbps, which is the bandwidth limit of the deployed network (gigabit Ethernet).

A. Evaluation Environment

The performance test considers reading and writing over multiple concurrent connections, accessing multiple documents with varying sizes. In order to accurately simulate the real scenario, a document set was generated based on the real environment. The document format used was the PDF. This is the adopted standard by TCE-TO. Fig. 2 presents the histogram of frequency distribution of the size of the documents whose tests were performed.

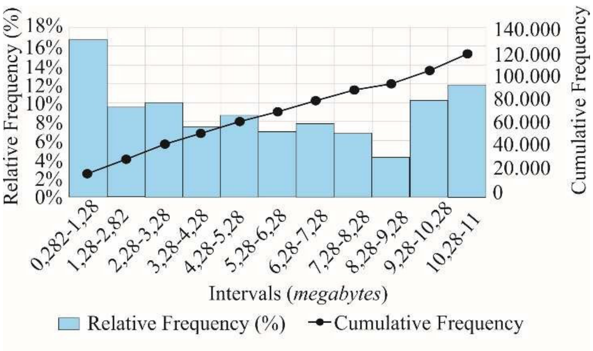

Fig. 2. Histogram of Frequency Distribution of the Size of the Documents.

In total, 10 cycles of executions were carried out. The first cycle used a single connection, doubling with each subsequent cycle until 512 parallel connections were achieved. The independent replication method was used for each reading and writing test in order to obtain confidence intervals with 95% confidence level (reliability), in a total of 5 independent data collections.

The results presented were obtained in the environment discussed in Section 4. In order to ensure an evaluation with realistic background traffic, the measurements were made during work hours between 08:00 and 18:00, over a period of 60 days, with every measurement experiment for each of the different operation modes lasting 60 seconds.

One of the main advantages of cloud storage is the possibility of increasing its service capacity on demand. To test the scalability, an analysis was performed to determine how many objects could be added to the cloud (server cluster for object storage) and whether the performance remained constant as the number of CAS servers was increased. The availability of the cloud at different workloads was also assessed.

For the evaluation of the object storage solution (CAS), a single container (storage "directory") was created to receive all the saved documents. In addition, a comparison was made between block and object storage. Lastly, block-level deduplication with 4 KB block size was tested. The focus of this comparison was to obtain the number of read/write transactions per second supported by the storage solutions and their service availability.

As discussed in [17], the goal of this evaluation was to demonstrate through the results that the performance of object storage is comparable and in some cases better than block storage systems. All parameters used in the simulations are based on the literature [13] and [14].

B. Evaluation of the Read and Write Procedure in a 282 KB Document

Initially, tests were performed to evaluate the performance, availability and scalability with cyclic reading on a single document with a size of 282 KB, as shown in Fig. 3. In this section the graphs present the average transactions in the read/write process of such a document, the columns displayed on the x-axis illustrate the number of concurrent connections, and on the y-axis the total number of successful transactions (left side) and percentage of service availability (right side).

For the graphs that show the average flow and response time for read/write of a document with a size of 282 KB, the columns displayed on the x-axis illustrate the number of concurrent connections and on the y-axis there flow rate (left side) and response time (right side).

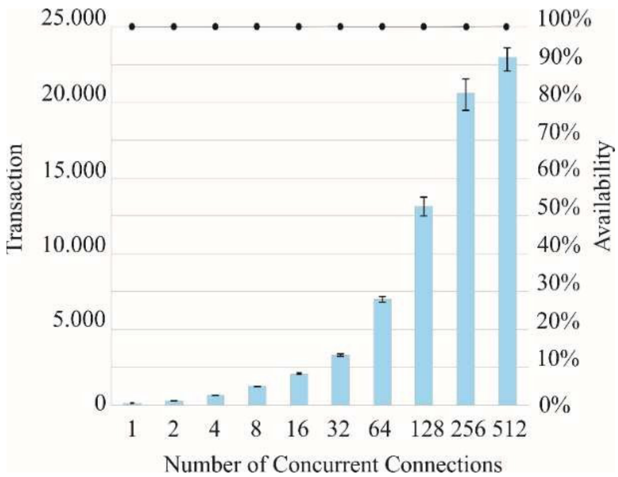

Fig. 3. Average Transactions in the Read Process of a Document with Size of 282 KB.

The results show that by increasing the number of concurrent connections, the average number of transactions (represented by a slash) performed increased linearly and the average availability (represented by a line) of the service was maintained at 100% of the read transactions. This increase was possible since the number of transactions did not exceed the limit of network data traffic (282 KB) considered small in this scenario.

Based on the presented results it can be concluded that the structure for object storage to repetitive requests on a single document provides for good scalability for reading transactions. According to the tests performed, running 512 concurrent connections resulted in an average reading transaction of 22,882.80.

The graph presented in Fig. 4 complements the graph shown in Fig. 3, since it shows that there was an increase in the average read-out of all 10 connection cycles evaluated in this work and a mean of almost constant response time up to 64 connections, between 64 To 256 concurrent connections the average response time tends to double, and with 512 concurrent connections the response time increased in a quadratic manner.

Therefore, it is recommended for this structure to limit the 55.58% concurrent connections to 256 for the processing of requests with documents with a size of 282 KB, which allows a low response time for transactions of less than 256 concurrent connections. More connections in this scenario reduce the number of successful transactions and increase response time due to the high traffic of competing connections.

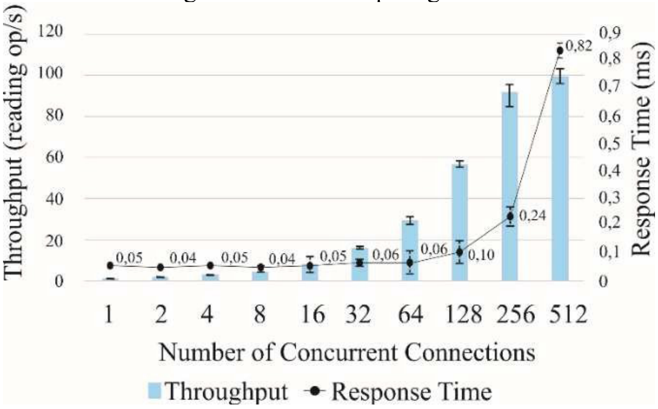

Fig. 4. Average Flow Rate and Response Time for Reading a Document Size 282 KB.

The results presented in Fig. 5 show that the structure of storage by objects (Average Transactions CAS) of the other analyzed structures is the one that operation increased in the number of transactions with each new cycle and at the same time maintained the average of its availability in 100% in all scenarios. It was only possible to obtain this total numbers of transactions because this object storage structure offers scalability, that is, a horizontal growth relative to the number of concurrent connections.

The block storage (Average Transactions per block) shown in Fig. 5, presented a growth according to the number of transactions executed, whose results prove to be the most feasible for this environment since with 512 concurrent connections there was a reduction only 95.42% in the availability of the service. The reduction occurs because of the limit of data traffic reached and processing by the file server. However, this solution provides less scalability than an object structure as well as reduced reliability as presented in a subsection for evaluation of the reading and writing procedure in multiple documents.

Deduplication blocking (Medium Deduplication Transactions) has shown that up to 64 concurrent connections are feasible, after which the availability of the service decreases as the number of concurrent connections increases. According to the tests performed, with 512 concurrent connections, the average service availability decreased to 55.88%, with this having a transaction error rate greater than the successfully completed transactions, as explained by the interval The "Average Transactions Deduplication" bar.

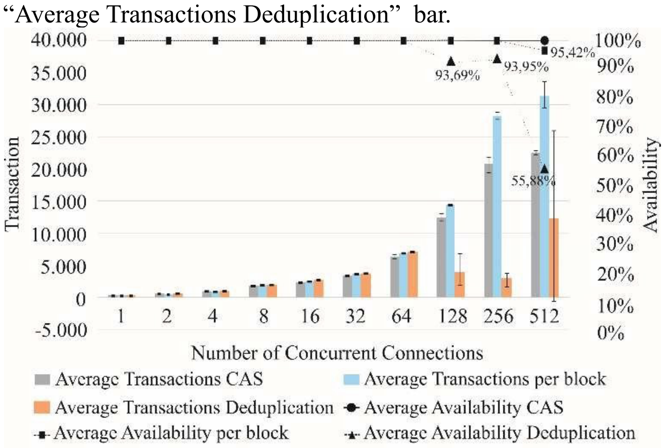

Fig. 5. Average Transactions and Availability to Read a Document Size of 282 KB.

It can be seen in Fig. 6 that the availability (represented by a line) is kept at 100% for writing files with up to 256 concurrent connections. The bar represents the average of transactions performed, so that with 512 concurrent connections it was possible to save an average of 4,666.80 documents in the storage device per object and obtain a service availability of 97.29%.

Both the number of transactions and the availability decreases upon reaching the acceptable limit of supported data traffic capacity as shown in Fig. 7 for a total of 512 concurrent connections.

When analyzing Fig. 7, it is seen that the increase in flow reaches its peak with 256 concurrent connections by performing 4,734 transactions (Fig. 6) and a linear growth up to 64 connections, with a mean response time of 2.85 ms (milliseconds). From the point of view of the traffic capacity of this environment, more concurrent connections beyond this point only increase the response time, however they do not increase the flow, since the limit regarding the transfer of data has already been reached.

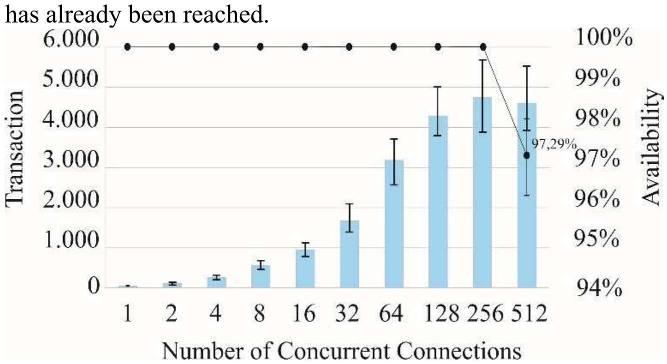

Fig. 6. Average Transactions in the Write Process of a Document with a Size of 282 KB.

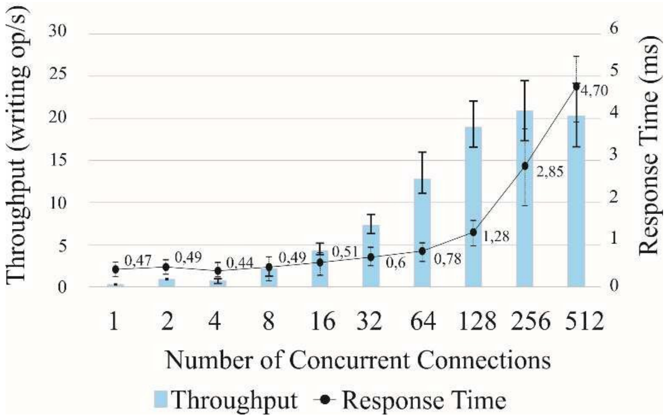

Fig. 7. Average Flow and Response Time for Writing a Document with a Size of 282 KB.

It is possible to verify in Fig. 8 the number of transactions and the percentage obtained for each of the 3 data storage environments evaluated in this work whose results show that the data storage by block with block-level deduplication for the writing procedure of one single document of 282 KB is the most feasible since the process of comparing hashes to a document of this size has low response time.

Still on Fig. 8, the apex for the average transactions performed with the deduplicated block data storage structure represented by the "Average Deduplication Transactions" was 8,331.8 for 128 concurrent connections, and with an average service availability of 100%. In this scenario, approximately

70 data blocks were generated when using block level deduplication with a fixed length of 4 KB. In this approach, the blocks were compared in the course of their insertion with the

hash blocks of the first document already stored.

Since in this context it is the document already stored in the first operation performed in one of the test cycles, there is no need to save the same document on this server. For this reason, the number of transactions that were performed is much higher than the other storage structures evaluated, since there is no need for new insertions, only the memory addresses of the positioning of each block are added.

When analyzing the storage per object represented in the graph by the "Average CAS Transactions" of Fig. 8, with the data storage per block represented in the chart by "Average Transactions per block", it is observed that the number of transactions performed tends to increase as the number of connections grows. As be seen, the block structure reaches its threshold with 64 concurrent connections and with 256 concurrent connections there is a 35% difference of reduced transactions compared to the Average CAS Transactions.

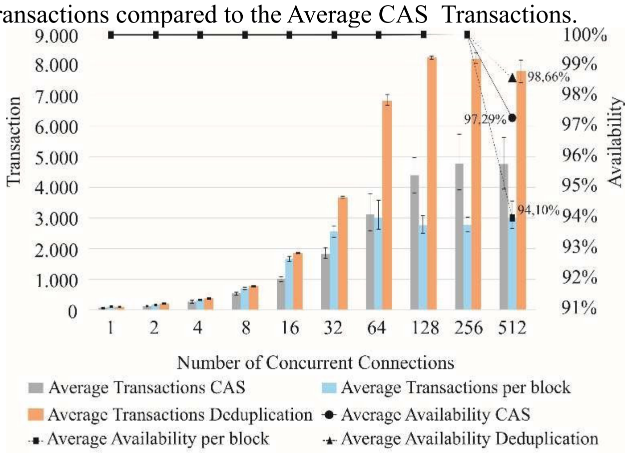

Fig. 8. Average Transactions and Availability to Write a Document with a Size of 282 KB.

Taking this into account, it can be observed that the average number of transactions obtained for 256 concurrent connections was 4,734, and the average service availability was 100%. Equivalent to that shown in Fig. 6, more connections beyond this point did not lead to an increase in the flow of traffic data.

C. Evaluation of the Read and Write Procedure on a 10 MB Document

In this section, the graphs that present the average transactions in the read/write process of a document with a size of 10 MB, the columns presented on the x-axis illustrate the number of concurrent connections and on the y-axis the total number of successful transactions (left side) and the percentage of service availability (right side).

For the graphs that show the average flow and response time for read / write of a document with a size of 10 MB, the columns displayed on the x-axis illustrate the number of concurrent connections and on the y-axis the flow rate (left side) and response time (right side).

For measurement in the cyclic reading of a single document with a size of 10 MB, the object storage structure (Fig. 9) resulted in a growth in the number of transactions executed up to the total of 128 concurrent connections, as well an average of 100% service availability (represented by a line) for this number of connections.

The number of transactions was sustained after 128 concurrent connections based on the available data traffic flow and only when a threshold of 256 concurrent connections reaches the threshold does a decrease occur. This reduction occurs because object-stored servers are unable to service all 512 concurrent connections to the cyclic read procedure in a single 10 MB document. In this way, both the number of transactions and the availability of the service are negatively affected.

Fig. 10 shows that the response time remained below 1.20 ms with up to 128 concurrent connections. This result confirms the values presented in Fig. 9, since it demonstrates that the data traffic limit was reached. As the number of connections grows, the response time practically doubles, and the number of operations per second (op/s) for reading a 10 MB document is reduced.

Fig. 11 illustrates that for the storage by object (represented by "Average CAS Transactions"), the number of transactions achieved was higher than the other analyzed structures, as well as the average service availability. It is feasible to say that the percentage suggestive of the service availability with 512 concurrent connections is equivalent to that obtained in the block structure (represented by "Average Transactions per block") since the number of transactions carried out with the same number of connections and execution time has been higher.

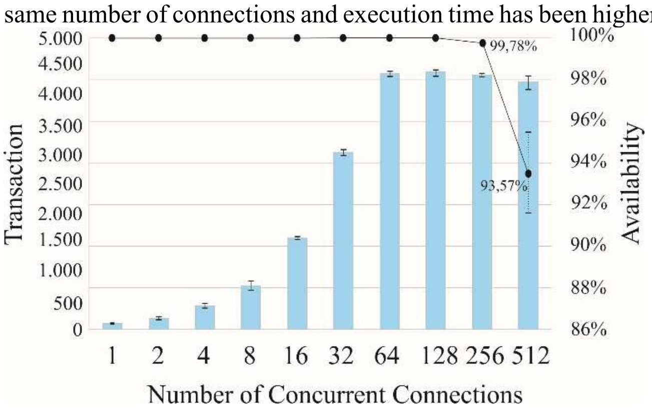

Fig. 9. Average Transactions in the Process of Reading a Document with a 10 MB Size.

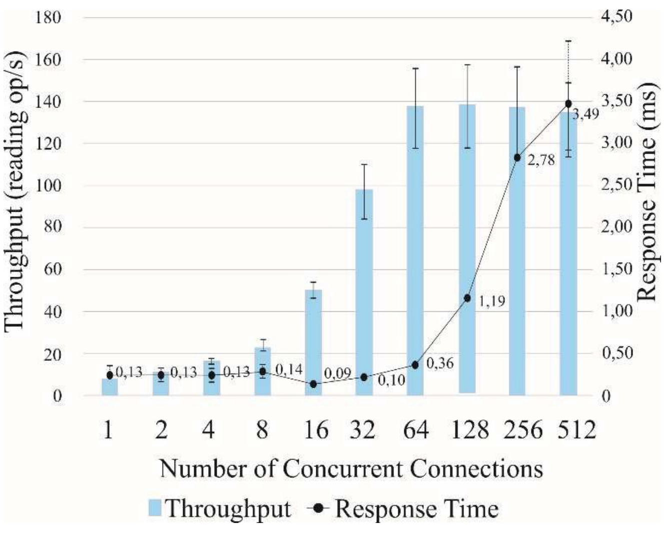

Fig. 10. Average Flow and Response Time for Reading a Document with a 10 MB Size.

The storage structure per block (represented by "Average Transactions per block") has also been shown to be equivalent to other structures with up to 8 concurrent connections. After this number of connections, in almost all cases it was superior to the structure with deduplication with the exception of the operating cycle with 64 concurrent connections among which the results are approximate. The average availability of the service remained at 100% up to 128 concurrent connections. For the other cases there was a reduction in the number of transactions supported. It is concluded that this architecture is equivalent to an object storage structure for the cyclic reading procedure with a single document with size of 10MB.

According to the tests performed and illustrated in Fig. 11, the storage structure per block with deduplication (represented by "Average Transactions Deduplication") was shown to be proportional to the other structures with up to 8 concurrent connections. By increasing the number of connections, in none of the other cases were results higher than the number of transactions performed, and their mean availability was reduced after 128 concurrent connections.

It is possible to observe that through the tests performed with deduplication, in the worst case, that is, with 512 concurrent connections their average availability fell to 84.60%. Such a reduction occurs due to the time required for hashing each of the 10 MB documents to which the read procedure is performed, so as the number of competing connections grows, it negatively impacts the number of transactions that can be made.

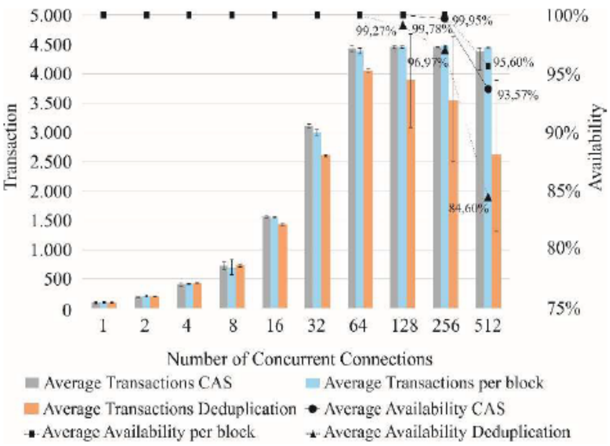

Fig. 11. Average Transactions and Availability to Read a Document with a 10 MB Size.

In relation to the writing of a document with a size of 10 MB, the results obtained slow the inverse of that reached in subsection B. As shown in Fig. 12, the average number of transactions carried out was 4.471,40 transactions for a total of 128 concurrent connections. The line represents the average service availability obtained, which was 100% for the same number of competing connections.

Service availability declines to 97.57% as the number of concurrent connections increases to 512, meaning this infrastructure supports up to 256 concurrent connections with 100% availability, attempts at transactions with more connections are not completed. It is concluded that, as the recommended threshold, there are a total of 128 concurrent connections for data transfer with gain in performance since, when increasing the number of connections, the tendency is to maintain the flow for these values analyzed and with constant rate without greater gains as can be seen in Fig. 13.

According to the results shown in Fig. 13, the write flow rate increase occurs up to 128 concurrent connections per client and reaches its limit by performing 4.471,40 write transactions, with a mean response time of 1.18 ms. Based on these results, it is concluded that more connections beyond this point only directly impact the response time yet the data traffic capacity is maintained.

In Fig. 14 it can be seen that the number of transactions represented by the "CAS Transactions Average" performed successfully is higher than the other storage structures analyzed, as well as the service availability of the number of requests made. In this graph the average number of transactions with the CAS was 4,410.80 for a total of 512 concurrent connections, and that in this cycle of operations the service availability decreased to 97.57%, ie its transaction threshold was reached.

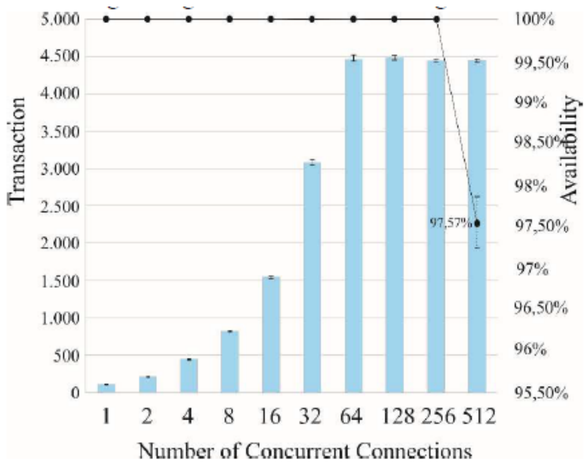

Fig. 12. Average Transactions in the Writing Process of a 10 MB Document Size.

In Fig. 14 it can be seen that the number of transactions represented by the "CAS Transactions Average" performed successfully is higher than the other storage structures analyzed, as well as the service availability of the number of requests made. In this graph the average number of transactions with the CAS was 4,410.80 for a total of 512 concurrent connections, and that in this cycle of operations the service availability decreased to 97.57%, ie its transaction threshold was reached.

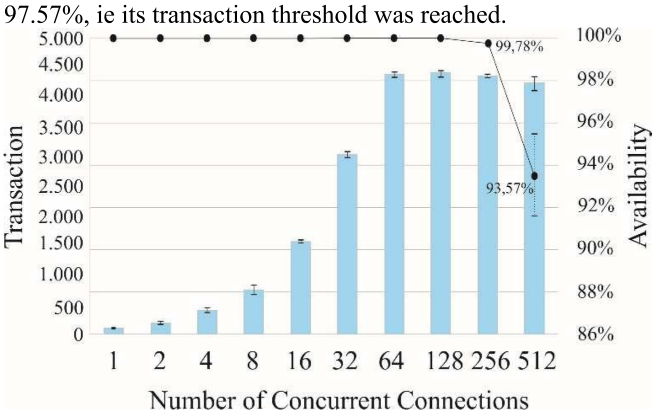

Fig. 13. Average Flow and Response Time for Writing a 10 MB Document.

None of the results obtained for data storage per block (Fig.

14) represented "Average Transactions per block" and availability were higher than the other structures evaluated. However, by increasing the number of connections, the average obtained for transactions performed declined, and the percentage relevant to service availability after 128 concurrent connections was reduced by up to 20.61% with 512 concurrent connections. It is noticed that for the writing procedure repeated times based on a single document with size of 10 MB, the structure of storage by blocks presents vertical growth and reliability inferior to the structure by object.

Also, based on this graph (Fig. 14), the data storage per block with deduplication represented by the "Average Transactions Deduplication", obtained a number of transactions successfully completed up to a total of 128 concurrent connections, after this value the service availability decays to 71.01% with 256 concurrent connections and 39.39% with 512 concurrent connections. Finally, in comparison to the storage of data by objects, both the number of transactions and the availability of the service decline by more than 75%. Therefore, based on these values, it can be seen that this deduplication structure is not suitable for large files (10 MB), since the time required to process blocks with a size of 4 KB is high.

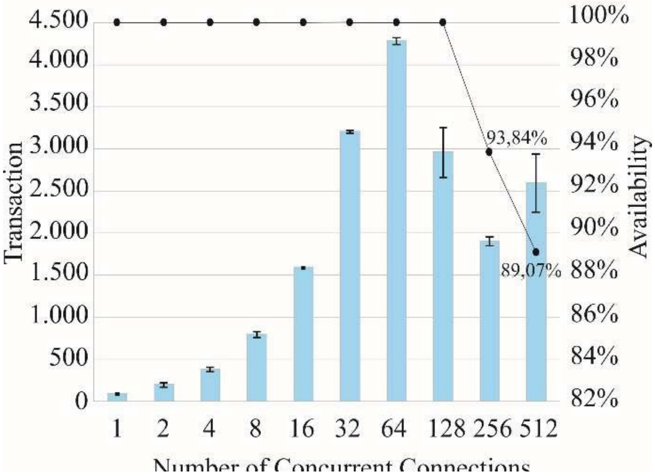

Fig. 14. Average Transactions and Availability for Writing a 10 MB Document.

D. Evaluation of the Procedure of Reading and Writing in Multiple Documents.

In the results in this section that show the average number of transactions in the process of reading/writing multiple documents, the horizontal axis is the number of concurrent connections and the vertical axis indicates the number of successful transactions (on the left) and the percentage of service availability (on the right).

In the results showing the average throughput and response time for reading/writing multiple documents, the horizontal axis is the number of concurrent connections and the vertical axis shows the average throughput (left) and response time (right).

As seen in Fig. 15, the results obtained indicate that for transaction volume and service availability while reading documents of varying sizes (kilobytes and megabytes) in a single container, storage per objects is recommended for up to 64 concurrent connections with 100% availability.

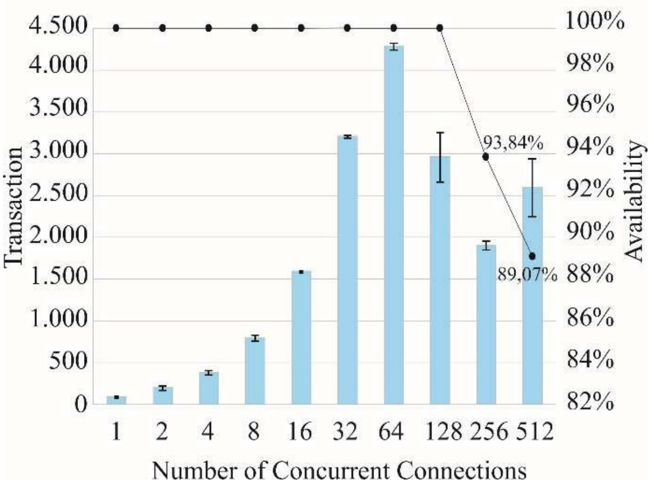

Fig. 15. Average of Transactions in the Process of Reading Multiple Documents with Varying Sizes.

By analyzing Fig. 15, one can see that the average number of transactions was 4,278.40 for 64 concurrent connections and that the service availability and number of transactions decrease as the number of concurrent connections increases. This situation occurs because the number of transactions that can be performed on this storage structure has reached its data traffic limit.

Fig. 16 shows the average throughput and response time for reading multiple documents. In this environment the average response time is maintained below 6ms up to 128 concurrent connections, between 128 and 256 this time doubles. Therefore, it is observed that the throughput grows up to a total of 128 concurrent connections and from this point on a sharp decrease in the number of transactions that can be performed takes place.

After this point, the response time doubles for 256 concurrent connections and with 512 concurrent connections the number of fault transactions grows. This occurs because there is a large amount of network traffic due to the number of read operations with documents of varying sizes, the limit supported between the web server and CAS server having already been reached which directly impacts the number of transactions that can be performed.

Fig. 17 illustrates that the storage per objects represented by "Average CAS Transactions" is similar to the other structures up to 16 concurrent connections. After this, the number of transactions with this structure increases up to 64 concurrent connections with an average of 100% service availability. This growth in the number of transactions occurs due to the distributed storage among the 3 CAS servers, namely scalability.

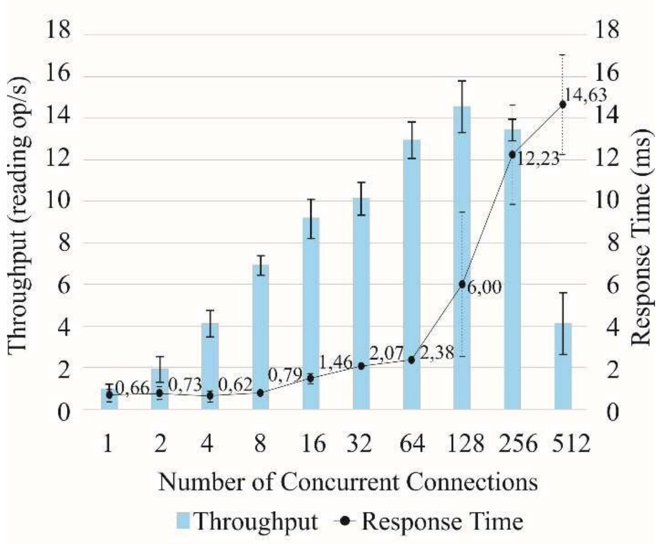

Fig. 16. Throughput Average and Response Time for Reading Multiple Documents with Varying Sizes.

From Fig. 17 one can also conclude that both the structure per block represented by the "Average transactions per block" and the structure with deduplication represented by the "Average transactions deduplication" suffer from a reduction in the number of transactions when reading multiple documents with over 64 concurrent connections. This occurs because the response time for these servers when reading from multiple documents with varying sizes is higher than CAS server.

For the file server with deduplication, as the number of concurrent connections increases it takes more time to process all of the transactions which negatively impacts the number of supported transactions and service availability. The average service availability is maintained at 100% up to 128 connections. By increasing the number of concurrent connections, the service availability reduces and, in the worst case, decreases 29.13%.

Tests conducted for writing documents with distinct sizes through multiple concurrent connections also showed that storage per object is the most recommendable solution. Fig. 18 illustrates that as the number of concurrent connections increases, the number of transactions grows up to a threshold of 128 concurrent connections. At that level one sees an average of 5,205.40 transactions and service availability of 100%. Above 128 concurrent connections, service availability decreases to 99.72% with 256 concurrent connections and 97.54% with 512 concurrent connections. Another negative point is that there was a considerable reduction in the number of transactions, yet the throughput was maintained as can be seen in Fig. 19. The reduction in the number of transactions occurs because the data traffic threshold has been reached.

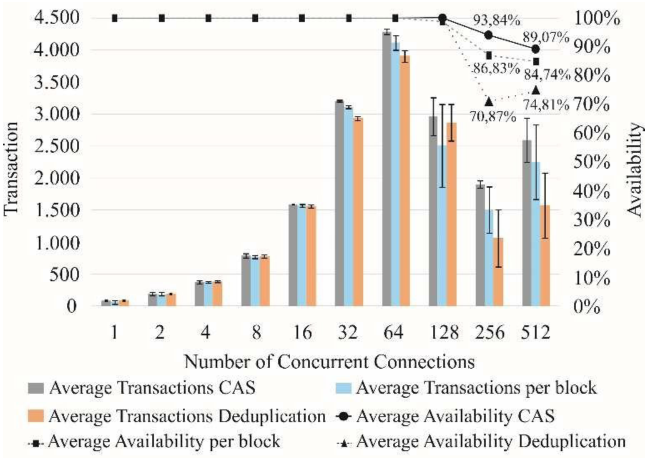

Fig. 17. Average Transactions and Availability for Reading Multiple Documents with Varying Sizes in Distinct Scenarios.

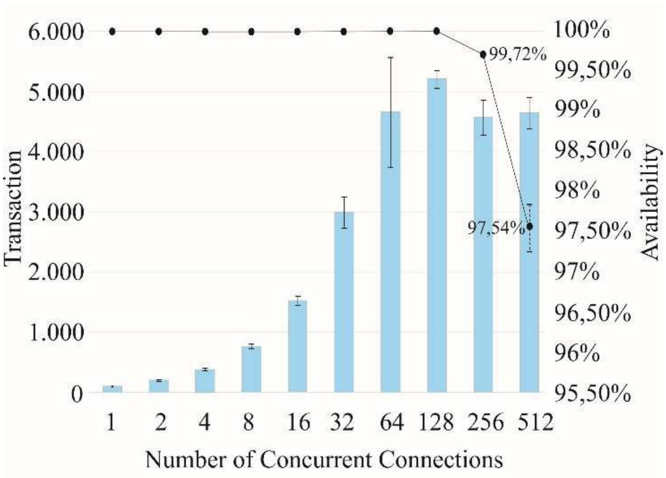

Fig. 18. Average of Transactions in the Writing Process of Multiple Documents with Varying Sizes.

Fig. 18 shows that throughput increases up to 128 concurrent connections and the response time remains below 0.94 ms. Above this number of concurrent connections, the average throughput decays but remains near the limit and response time grows linearly. In this way it is possible to conclude that there will be no more performance gain after 128 concurrent connections.

Fig. 20 shows the average number of transactions and availability of services across 3 structures, in which the storage by objects has maintained its average availability in all cases up to 128 concurrent connections. In this scenario the CAS server presented a very relevant feature compared to the other structures in that it kept the number of transactions performed close to its limit, or in other words, its scalability when running multiple connections with multiple documents.

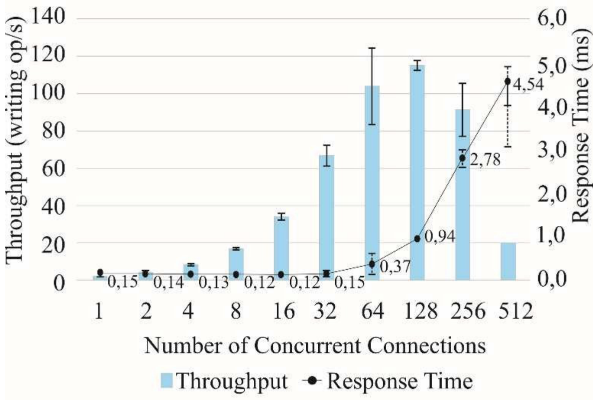

Fig. 19. Average Throughput and Response Time for Writing Multiple Documents with Varying Sizes.

Fig. 20 illustrates that the results obtained from the data storage with deduplication represented by the "Average Transactions Deduplication" compared to the other structures analyzed has a high confidence interval after 64 concurrent connections and low service availability as the number of concurrent connections exceeded the acceptable limit of transactions, which impacted the results obtained for each test cycle.

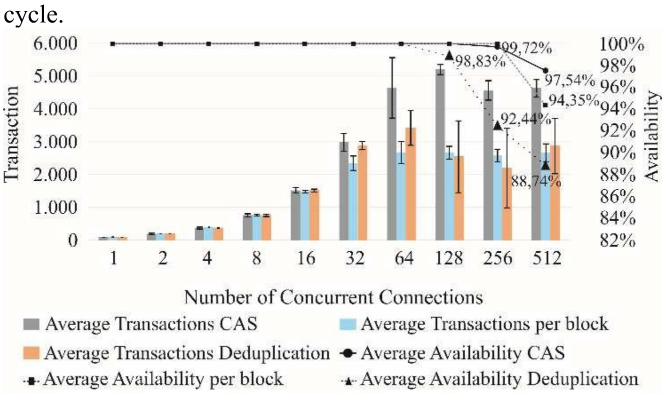

Fig. 20. Average Transactions and Availability for Writing of Multiple Documents with Varying Sizes in Distinct Scenarios.

Finally, it is worth mentioning that in this environment, for each cycle after 64 concurrent connections, the web server locked up both with the block structure and the block structure with deduplication, the only solution being to reboot the server. No studies were performed to identify the reason for the web server freeze.

An important lesson is that the content-addressable storage architecture proved to be the most viable for read transactions with 64 concurrent connections and writing with 128 concurrent connections, with 100% service availability for each of the cases analyzed.

E. Scalability and Replication

Scalability is obtained by performing the reading and insertion of digital documents over an object structure with horizontally distributed container(s). In this way, search procedures, indexing and storage based on the ID of a digital document result in less network traffic than with hierarchical file systems.

A comparison was created to show off the capacity supported by each of the object storage servers when reading multiple documents. Based on the example in Fig. 21, it can be observed that by adding 2 more servers, the throughput grows until it reaches 128 concurrent connections, the limit reached in this test environment. Above this value one can conclude that as more connections are made, the number of fault transactions increases once the throughput has reached its limit.

In the event of unavailability of a storage unit, the others devices belonging to the group continue to operate without impacting service availability.

If a total failure occurs and a CAS server requires replacement, a process of recovery begins which recovers the data from the other 2 online servers.

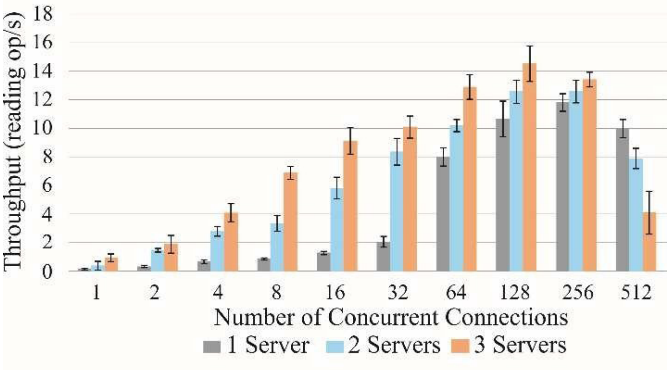

Fig. 21. Average Throughput in a Distributed System.

VI. CONCLUSION

With the object storage service becoming increasingly accepted as a substitute for traditional block storage systems, it is important to measure the performance of these services effectively. This paper proposes the deployment of content addressable storage (CAS) architecture with file-level deduplication for digital preservation in a private cloud platform and evaluates the resulting system.

In the first scenario, as presented in section V, subsection B, according to the reading and writing tests performed in a document with size of 282 KB, the structure by objects presented gains for all cycles, both in the number of transactions performed whose variation for each cycle was maintained linearly in the percentage related to service availability, a situation that did not occur with the other storage structures. However, the structure of data storage by block presented gains of 38.19% in the percentage of transactions in the reading procedure. The structure of data storage by blocks with deduplication presented gains of 75.16% in the percentage of transactions in the writing procedure.

For the scenario presented in section V, subsection C (documents with a size of 10 MB), the structure has been shown to be equivalent to a block structure with gains of 0.57% in the percentage of transactions in the procedure of reading a single document in this dissertation as great. However, for the writing procedure, gains with results 93.8% higher than the other structures for the number of transactions performed, and 100% service availability.

The third scenario, shown in section V, subsection D (various size documents), shows an increase of 15.36% in the percentage of transactions in the reading procedure and 48.67% in the percentage of transactions in the writing procedure in relation to the others. In addition, the results obtained with both the flow rate and the response time for the CAS show the acceptable threshold for the proposed structure in each environment.

We conclude that the object-based storage architecture enables real gains in the number of transactions carried out with marked improvement in both reading and writing until the flow rate threshold is reached. On the other hand, it was observed that when the number of connections is increased beyond the limit obtained for each test environment, the response time is prolonged and the data transfer rate is kept.

Similarly, the test for scalability also demonstrates that there is a growth in the number of transactions to the extent that more CAS servers must be added to reach the apex of data traffic supported by this infrastructure.

REFERENCES

[1] L. C. Miller, “Object storage for dummies - NetApp special edition”. Wiley, 2013, p. 14.
[2] M. B. Almeida, B. V. Cendón and R. R. Souza, “Methodology for digital document preservation program implementation long term”. Meetings Bibli: electronic journal of library and information science. 17(34):103- 130, 2012, p. 103.
[3] L. Duranti, “Reliability and authenticity: the concepts and their implications”. Retrieved from: http://journals.sfu.ca/archivar/index.php/archivaria/article/view/12063/1 3035, february 2015.
[4] J. Wu, L. Ping, X. Ge, Y. Wang and J. Fu, “Cloud storage as the infrastructure of cloud computing”. Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on, 2010, pp. 380-383.
[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility”. Future Generation Computer Systems, vol. 25, pp. 599-616, June 2009.
[6] Q. Li and Y. Guo, “Optimization of resource scheduling in cloud computing”. Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2010 12th International Symposium on, 2010.
[7] T. Erl, Z. Mahmood and R. Puttini, “Cloud computing: concepts, technology & architecture”. Prentice Hall, 2013, p. 528.
[8] P. Mell and T. Grance, “The NIST definition of cloud computing (draft)”. The National Institute of Standards and Technology, 2011, pp. 7.
[9] G. Somasundaram and A. Shrivastava, “Information storage and management: storing, managing, and protecting digital information in classic, virtualized, and cloud environments, 2nd edition”. Wiley, 2012, p.187.
[10] L. L. You, K. T. Pollack, D. D. E. Long and K. Gopinath, “Presidio: a framework for efficient archival data storage”. ACM Transactions on Storage, vol. 7, No. 2, 2011, pp. 1, 4.
[11] K. Jindarak and P. Uthayopas, “Enhancing cloud object storage performance using dynamic replication approach”. Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on, 2012, p. 800.
[12] Y. Zhang, Y. Wu and G. Yang, “Droplet: A distributed solution of data deduplication”. Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on, 2012, p. 114.
[13] Q. Zheng, H. Chen, Y. Wang, J. Duan and Z. Huang, “COSBench: A benchmark tool for cloud object storage services”. Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, 2012, pp. 998- 999.
[14] Nasuni, “State of cloud storage providers industry benchmark report: a comparison of performance, stability and scalability”. Technical report, Nasuni, 2011.
[15] Opendedup, “Opendedup - opensource dedupe to cloud and local storage”. Retrieved from: http://opendedup.org/, apr of 2014.
[16] J. Fulmer, “Joe dog software - Proudly serving the Internets since 1999”. Retrieved from: http://www.joedog.org/siege-home/, may of 2014.
[17] Y. Jinsong, W. Chunlu and L. Chuanyi, “Performance comparisons of a content-addressable storage network system and other typical ip-san based storage systems”. Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on, 2011, p. 1142.