An Optimized Model for Files Storage Service in Cloud Environments

Haythem Yahyaoui and Samir Moalla
Department of Computer science, University of Tunis el Manar, Tunis, Tunisia

Keywords: Cloud Computing, Data Centers, File Storage, File Redundancy, Virtual Data Center.

Abstract: Cloud computing represents nowadays a revolution in the distributed systems domain because of its several services. One of the most used services in Cloud environments is file storage, which consist on uploading files to the Cloud's data center and using them at anytime from anywhere. Due to the highest number of Cloud customers we risk to have a bad management of cloud's data center such as losing space by files redundancy which can be provided by uploading one file for several times by only one customer or the same file by many customers. Such a problem is very frequent when we have a big number of customers. Many studies have been done in this order, researchers propose many solutions and each one has its advantages and disadvantages. In order to save space and minimize costs we propose an optimized model which consists on deleting file redundancy before the duplication step in the data centers. Experimentally, during the evaluation phase, our model will be compared with the some existing methods.

1 INTRODUCTION

File storage is one of the most used services in the network, the appearing of some technologies such as public and private Clouds, and Big data, increases the growth of the stored data around the world (Xu and Shi, 2013).

Storing customer's file with the lowest cost is one of the biggest challenges between providers that offer such a service.

Cloud providers are thus studying new solutions to minimize the costs of their services (Khanafer and Puttaswamy, 2013). Indeed, there are many ways to achieve that goal such as green data centers, which consist on optimizing the energy consumption, consequently minimizing the service's cost (Mansouri and Buyya, 2013).

However, such a way is based on the algorithm optimization techniques and green power consumption, but not on the storage process optimization, and consequently, this method will be unable to prevent a file redundancy and save space in the data center.

Furthermore, we have to understand that there is a strong relation between optimizing the data center's space and minimizing costs, a great optimization gives us more free space, consequently more customers.

In this paper, we focus on finding a solution for the files redundancy in data centers, we propose an automated model based on the generation of a fingerprint for every file before the duplication phase, and using it as a file identifier. Before the duplication, the customer's file will be passed through a Redundancy Checker (RC), if the file exists in the data center, the customer will be notified that his file was stored succefully, if not, the cutomer's file will be stored and duplicated around the different data centers. This solution allows Cloud providers to save a free space around the data centers by preventing the files redundancy. Our model will be implemented in a virtual environment, and its evaluation will be based on the needed storage space results from data centers.

This paper is organized as follows, section 2 presents previous studies which focused on optimizing cost and speed of the storage process, we introduce our model in section 3, detailing each of its steps, in section 4 we present an evaluation of our model before we conclude the paper and discuss future works in section 5.

2 RELATED WORKS

The problem of minimizing data storage cost in cloud environments has been studied by many researchers, considering different types of criteria

Yahyaoui, H. and Moalla, S.
An Optimized Model for Files Storage Service in Cloud Environments.
In *Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016) - Volume 1*, pages 155-159
ISBN: 978-989-758-182-3

such as files security, availability of the storage service (Bo and Hong, 2015), etc. In the following sub part, we present some studies about customer files management in clouds.

2.1 Current Techniques

The availability of the customer files is the constraint that should be respected, to consider such a constraint, Google File System (GFS) duplicate customer's data around multiple data centers in order to make them available from any place (Unuvar and Tantawi, 2015), besides if one of the data centers breaks down, the others will ensure the performance, including scalability, reliability, and the availability of the stored files.

Currently, Google is using multiple GFS clusters, the largest ones have about 1000 storage nodes, and about 300TB of storage space (Feng and Meng, 2012).

In the case of Hadoop Distributed File System (HDFS), the customer's file is broken up into 64 MB chunks, every chunk is replicated for three times around the data centers (Kulkarni and Tracey, 2004).

Amazon Simple Storage Service (S3) makes some snapshots to every data center in order to ensure availability and security of customer's file (Ramgovind and Smith, 2010).

2.2 Limitations of Current Techniques

We studied above the most used techniques in the term of file storage systems, such as GFS, HDFS and S3, the providers of these techniques are the dominants of the field of Cloud environments and they have the biggest number of customers.

What we can distinguish from these methods is that they have some common problems. In fact, if there is a files redundancy in a data center used by one of these providers, all of these methods will be unable to prevent it and to save the storage space, consequently, these methods have a limitation in term of files management around the data centers.

3 PROPOSED MODEL

3.1 Problem Statement

Actually, the problem of files redundancy before the duplication phase in Cloud environments is one of the critical problems for providers, because it represents a loss of storage space, this problem is due to the increase of the number of customers around the world, and the popularity of the storage service.

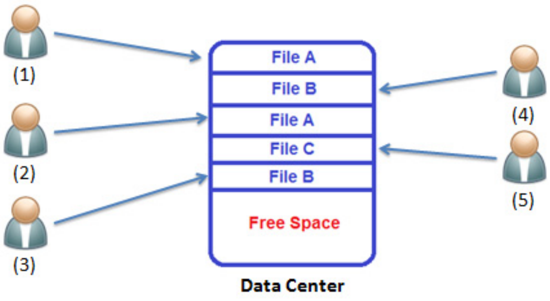

Figure 1: File Redundancy.

Figure 1 represents the redundant files in a data center caused by different customers before the duplication, we can distinguish that many users can upload the same file to the Data Center (DC), and consequently this scenario represents loss that can be accumulated after the duplication.

Precisely, the duplication of a redundant file represents a loss of storage space that can be calculated by the following formula:

$$L = S \text{ x } N \text{ x } M \qquad (1)$$

The equation (1) represents the loss of the storage space in the case of duplicating a redundant file:

- *S*: the size of the redundant file.
- *N*: number of the redundancy.
- *M*: number of data centers used for the duplication.

The optimal case here is to delete all the redundant files before the duplication phase:

$$OC = \ M \qquad (2)$$

The equation (2) represents the Optimal Case (OC), which is the duplication of only one file from all the redundant files, and delete all the others.

In fact, the duplication phase insures the performance, scalability, reliability, and the availability of the stored files. Indeed, the availability and security of the customer's file are constraints that should be taken into consideration by Cloud providers, consequently, all the solution in our context should respect these constraints.

3.2 Proposed Solution

Firstly, we have to mention that our solution will take the place just before the duplication process, because if it takes place after, it can ignore the constraints of the security and the availability.

To obtain the optimal storage space in data centers, we have to reject the file to store if it is available in the data centers, in this order we have to add a Redundancy Checker (RC) module.

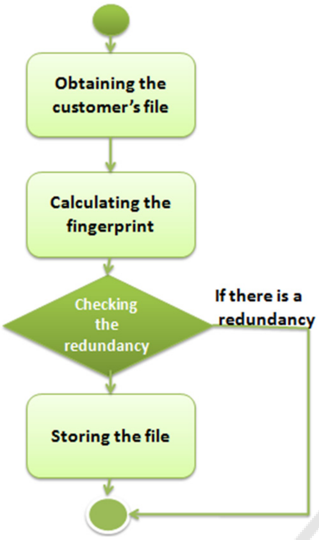

Figure 2: File Storage Steps.

Figure 2 represents the addition of an RC, which consists on comparing the customer's file with the existing stored files, the comparison is based on the File's Fingerprint (FFP), if there is a redundancy, the RC module will react by deleting the file in question.

To make our solution more efficient, we will calculate the FFP twice with different hash functions. We have to mention that the FFP comparison step will be executed only if the files to compare have the same size.

Here is the algorithm of the redundancy checker:

Algorithm 1: RC Algorithm

```
input: fileA, fileB
output: redundancy= true OR redundancy= false

sizeA ← size(fileA);
sizeB ← size(fileB);
redundancy ← false;

if sizeA==sizeB
  Then
    SHA1_A ← SHA1(fileA);
    SHA1_B ← SHA1(fileB);
    If SHA1_A== SHA1_B
      Then
        MD5_A ← MD5(fileA);
        MD5_B ← MD5(fileB);
        If MD5_A== MD5_B
          Then
            Redundancy ← true;
          Else
            Redundancy ← false;
        End
    End
End
```

Algorithm 1 describes all the steps realized by our RC, the FFP is calculated twice using SHA1 and MD5 hash functions.

In the case when a same file is stored twice but with different names, our model will leave just one file and link it with the two names using a Virtual Data Center (VDC), a such action will also be executed in the case of n same files with n different names.

Cloud environments are also characterized by the transparency, every customer should feel that he is alone in the platform, for that order we have to add a VDC module.

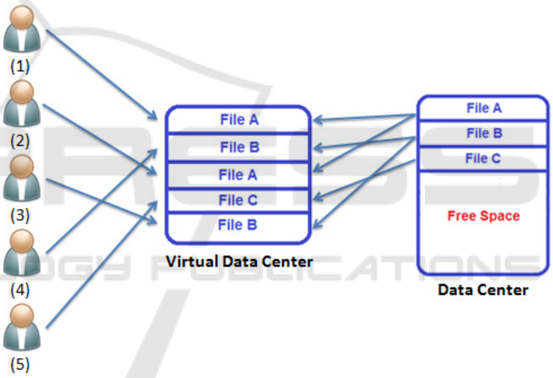

Figure 3: Virtual Data Center.

Figure 3 represents the VDC and also the data center after the redundancy checker step.

To summarize and clarify our solution we have to present it by figure 4:

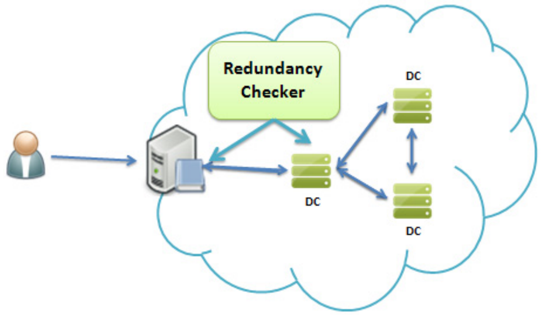

Figure 4: Proposed Solution.

Figure 4 represents the scenario when our customer sends his file to the cloud platform, in this case, the RC will intervene and make the decision to store or to delete the file in question.

4 EVALUATION

During this phase, we will use a set of files in our experimental tests. The evaluation will be based on the performance of our model compared to the GFS model.

The comparison will be based on the calculation of the loss space obtained from each model, also we have to mention that we will vary the values of S, N and M in the order to obtain different results.

Starting by the calculation of the storage space needed using GFS method and our proposed method (MHY):

Table 1: Obtained results by the variation of N.

Tests	N	S	M	GFS	MHY
1	2	1GB	4	8	4
2	3	1GB	4	12	4
3	4	1GB	4	16	4

Table 1 presents the obtained results from GFS and MHY by the variation of the number of redundancy, to facilitate this phase, we fixed the size of files to 1GB.

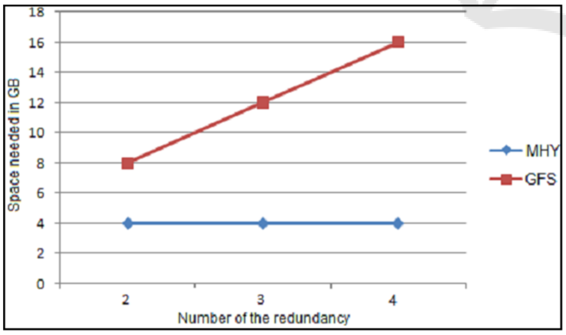

Figure 5: Results by the variation of N.

Figure 5 represents the obtained results from GFS and MHY by the variation of the number of redundancy, what we can conclude from this figure is that the space needed is proportional to N when we use the GFS method, but in the case of MHY it remains constant.

Table 2: Obtained results by the variation of M.

Tests	N	S	M	GFS	MHY
1	2	1GB	4	8	4
2	2	1GB	5	10	5
3	2	1GB	6	12	6

Table 2 presents the obtained results from GFS and MHY by the variation of the number of data centers used for the duplication.

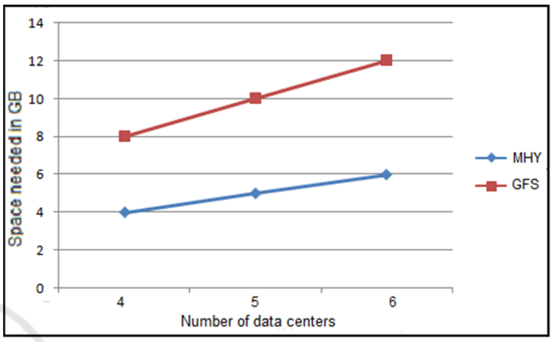

Figure 6: Results by the variation of M.

Figure 6 represents the obtained results from GFS and MHY by the variation of the number of the data centers used for duplication, what we can conclude from this figure is that increasing the number of data centers M, improves the availability of the files but in the case of GFS it also increases the loss space.

5 CONCLUSIONS

In this paper, we presented an optimized model for file storage service in Cloud environments in order to save storage space and minimize the storage's cost. Our model is based on the addition of an RC, which consist on checking if there is a redundant file in the data center just before the duplication. We evaluated our model by comparing the obtained results of the space needed to save a file to that obtained by using the Google File System (GFS) method and we found that our method is more efficient in term of saving storage space in the data centers.

We proved that a great management of the stored files in the data center can minimize the service's cost and also make the difference between providers.

In term of future work, we will make some modifications for our model in order to make it in term of execution time, because if we test it with a large set of files it is likely to generate some problems.

REFERENCES

Xu, G. L. Z. G. and Shi, K. (2013). Expander code: A scalable erasure-resilient code to keep up with data growth in distributed storage. IEEE 32nd International performance computing and communications conference.

Khanafer, A. M. and Puttaswamy, K.P.N. (2013). The constrained Ski-Rental problem and its application to online cloud cost optimization. IEEE INFOCOM.

Mansouri, Y. A.N. and Buyya, R. (2013). Brokering Algorithms for Optimizing the Availability and Cost of Cloud Storage Services. IEEE 5th International Conference on Cloud Computing Technology and Science.

Bo, M. S. and Hong, J. (2015). Improving Storage Availability in Cloud-of-Clouds with Hybrid Redundant Data Distribution. IEEE International Parallel and Distributed Processing Symposium.

Unuvar, M. S. Y.N. M.G. and Tantawi, A.N. (2015). Selecting Optimum Cloud Availability Zones by Learning User Satisfaction Levels. IEEE Transactions on Services Computing.

Feng, Q. H. G. and Meng, D. (2012). Magicube: High Reliability and Low Redundancy Storage Architecture for Cloud Computing. IEEE 7th International Conference on Networking, Architecture and Storage.

Kulkarni, P. F. J. and Tracey, J. M. (2004). Redundancy Elimination Within Large Collections of Files. Annual Technical Conference.

Ramgovind, S. M.M. and Smith, E. (2010). The management of security in Cloud computing. Information Security for South Africa.