MSSF: A step towards user-friendly multi-cloud data dispersal[1]

Rafael M. de O. Libardi[2], Marcos V.N. Bedo[2],
Stephan Reiff-Marganiec[3] and Julio C. Estrella[2]
[2]University of São Paulo, São Carlos - SP - Brazil
[3]University of Leicester, Leicester - UK
[2]{mira, bedo, jcezar}@icmc.usp.br
[3]srm13@le.ac.uk

***Abstract*—With an increasing number of companies and individuals adopting cloud computing for their data needs. Naturally, there is a shift in financial and operational costs and and provided services should meet users' performance and cost expectations. We focus on storage and propose MSSF, a Multi-cloud Storage Selection Framework. MSSF contains a basic set of algorithms, a set of security rules and a formal definition of user profiles allowing to fit cloud storage services to user needs. Preliminary experiments investigate the cost differences between two baseline algorithms and user profile models. Considering the promising initial results we provide several observations about the MSSF decision making process that will help with future improvements.**

[1] ***Keywords*—Cost model, Dispersion algorithms, Cloud services**

I. Introduction

New opportunities offered by Cloud Computing bring new challenges. With many cloud services offering comparable functionality, Multi-cloud, that is the ability to combine several cloud offerings, is a new research field. The Multi-cloud paradigm aims to orchestrate and combine different service providers to deliver a better service. For example a user might have several cloud storage accounts that offer different capacity, security constraints, access limitations etc. The user on the other hand requires certain security levels for some files, has different access needs and wants to optimize the cost spend on achieving this. Systems implementing the concept of Multi-cloud currently merely act as delivery platforms storing files as per the users setting of parameters rather than providing intelligent, user friendly solutions for automating the matter.

Techniques capable to disperse and retrieve information with guaranteed security and availability levels exist in the literature [1], [2]. Two promising baselines algorithms are the Rabin Information Dispersal Algorithm [3] and the Reed-Solomon Codes [4]. They are based on erasure codes [3], allowing for files to be split into segments of which some, but not all, will be needed to restore the original file (thus providing replication and security in one approach). These algorithms have been studied from a complexity and information security point of view, but not how they will perform in real cloud environments. Considering real runtime behaviour in cloud systems through an empirical performance evaluation to understand processing time and storage needs will bring useful insight to referral strategies for automatic dispersion systems.

In this paper, we present initial design and evaluation results for a novel framework which implements an internal referral strategy, taking into account algorithm's cost and user needs. We refer to the framework as Multi-cloud Storage Selection Framework (MSSF). MSSF is composed of a set of baseline algorithms, a set of security rules, and a formal definition of user profiles. Relating those three elements, MSSF select the best Multi-cloud redundancy storage method automatically and thus transparently for the user.

In this initial version, the best Multi-cloud method is chosen in according to user profiles. An user profile is a structure composed by fields representing levels; for instance, the desired availability is a field with a percentage value. The profile also contains information on the cost a user is able to pay for the service, based on previous information. Alongside, the set of security rules capture (as security levels) the security policy required by the user. Redundancy needs are also represented by a levelled structure. Thus, when the *framework* has to decide on the best technique, it compares the user needs with the expected service provision offerings using Euclidean distance over the fields. A closer distance represents a better match. Preliminary evaluation compared the costs and performance of the Rabin and Reed-Solomon algorithms.

Our approach differs from existing by focusing on selection of a dispersal method which most appropriately addresses the user's needs. There are several approaches to aggregate cloud storage providers.Usually they provide a restricted set of options, forcing the user to make manual choices or restricting the properties available to them. E.g. iDataGuard [5] provides confidentiality but lacks features to handle redundancy. DepSky [6] provides confidentiality and redundancy features, but generates a large overhead. Considering cloud service selection related work [7], [8] is limited to selection of individual cloud services, and hence not suited to multi-cloud methods.

Following, we present the framework, consider the experimental evaluation and identify the next steps for our work.

II. Multi-cloud Storage Selection Framework

In our work the term *dispersion method* refers to a combination of a dispersal algorithm, settings for the algorithm (how

[1]The authors would like to thank FAPESP, CNPq and CAPES for the financial support. Part of the work was conducted while S. Reiff-Marganiec was on Study Leave supported by the University of Leicester.

will a file be split and recovered), location for metadata and the encryption level (none, metadata-only, etc). In the selection process, MSSF considers properties such as availability, security, access pattern, file size, performance requirements, cost and storage space overhead. In MSSF we have three main elements: redundancy algorithms, security rules and a user profiles, see Fig. 1. When a user demands a service the user profile describes required levels of service, either from historic data or automatic computation of reasonable values through security rules based on policies chosen by the user. A matrix with properties of the algorithms based on measures of storage and processing-times captures aspects of each service.

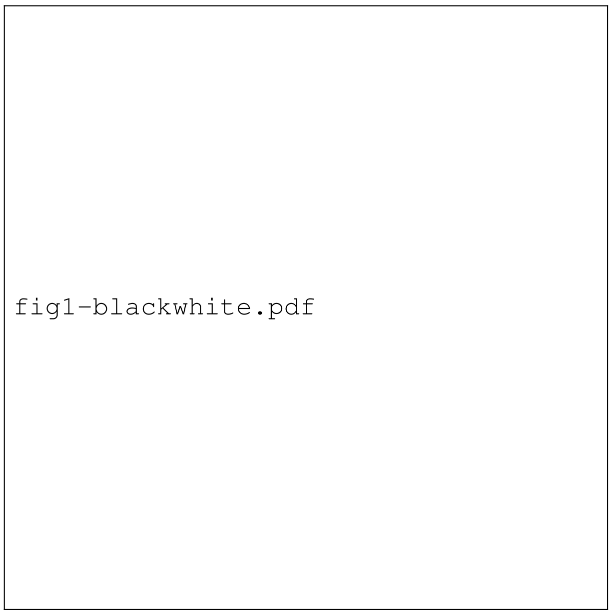

Fig. 1. Main elements of MSSF.

Thus the user profile is a vector $Y = \{y_1, y_2, ..., y_i\}$ and the matrix of values for the algorithms can be sliced into k vectors $X_l = \{x_{l,1}, x_{l,2}, ..., x_{l,i}\}$, where each y_i and $x_{l,i}$ have the same domain. MSSF will first discard methods whose cost c_l exceeds the possible payment p and then select the algorithm which satisfies the condition $min(\sum_{i=1}^{k} \sqrt{(x_i - y_i)^2})$ as part of the dispersion method.

III. Experiments

Our experiments compare the Rabin and Reed-Solomon algorithms. The experiment design considers three factors with two levels each and perform a complete factorial experiment. The selected factors are dispersal/retrieval time, the relation of parts (file split in n parts, with m needed for reconstruction), and obviously the algorithm. Our experiments show that, the algorithm and the configuration of the storage influence the time processing (Figure 2a and 2b) and storage overhead (Figure 2c). The Reed-Solomon (RS) algorithm is always faster than the Rabin algorithm, but it creates more space overhead. Overall, Rabin demands 57% less space overhead than Reed-Solomon at the cost of needing 9 times as much processing-time.

Such results allow us to complete a matrix of cost and benefits which can be used by an automated selection method.

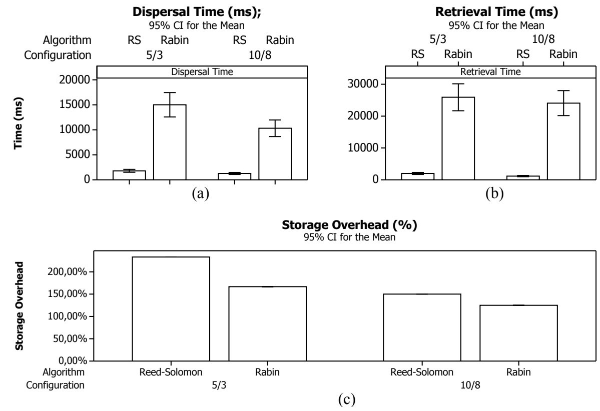

Fig. 2. Algorithm and Configuration influence on dispersal time and storage overhead. (a) Dispersal time (b) Retrieval time and (c) Storage overhead.

A simple linear equation using market values for storage costs and processing time in addition to the results matrix allows to compute and compare costs.

Selection requires the data presented in the previous paragraph and then needs to match this against user requirements. Recall that a user profile shows the user's requirements for speed and availability as well as the available amount for payment. Matching the demands and offers, MSSF selects the adequate dispersion method. In our tests, for profiles which require speed MSSF generally selects the Reed-Solomon approach, while for expirements where storage cost saving is needed the Rabin approach is selected.

IV. Discussion and Recommendations

We reported on a work-in progress effort to select the fittest dispersal method based on matching user demands with service offerings. MSSF is part of a larger project which aims to develop a highly flexible multi-cloud storage system using cryptographic techniques and redundancy configurations. Based on the literature review, we briefly described our proposed technique and some initial experiments. MSSF uses an Euclidean-distance measure to estimate which method will provide the best solution for a user's need. We believe that MSSF has great potential to help in automatic, user-friendly multi-cloud data dispersal.

Other techniques such as classifiers based on rules and multi-criteria decision making methods could also be promising. Future work will investigate further aggregation functions. We will also consider more complete sets of quality parameters as well as more differentiated profiles (such as differentiated handling of different file types and different access patterns). This will allow allowing more fine grained tuning of algorithm selection as well as the algorithm's input parameters.

References

[1] M. Li, "On the confidentiality of information dispersal algorithms and their erasure codes," *CoRR*, vol. abs/1206.4123, 2012.

[2] A. Juels and A. Oprea, "New approaches to security and availability for cloud data," *Commun. ACM*, vol. 56, no. 2, pp. 64–73, Feb. 2013.

[3] M. O. Rabin, "Sequences," R. M. Capocelli, Ed. New York, NY, USA: Springer-Verlag New York, Inc., 1990, ch. The information dispersal algorithm and its applications, pp. 406–419.

[4] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. Wiley. com, 1999.

[5] R. C. Jammalamadaka, R. Gamboni, S. Mehrotra, K. Seamons, and N. Venkatasubramanian, "A middleware approach for outsourcing data securely," *Computers & Security*, vol. 32, no. 0, pp. 252 – 266, 2013.

[6] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "depsky."

[7] C. Chen, S. Yan, G. Zhao, B. S. Lee, and S. Singhal, "A systematic framework enabling automatic conflict detection and explanation in cloud service selection for enterprises," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, June 2012, pp. 883–890.

[8] Z. ur Rehman, O. Hussain, and F. Hussain, "Multi-criteria IaaS service selection based on QoS history," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, March 2013, pp. 1129–1135.