Exploiting Rate less Codes in Cloud Storage System

S Sudhakar,
Department of Computer Science and Engineering,
National Institute of Engineering (NIE), Mysuru, Karnataka, India

Mr. S LOKESH, Assistant Professor,
Department of Computer Science and Engineering,
National Institute of Engineering (NIE), Mysuru, Karnataka, India

***Abstract*: Block-Level Cloud Storage (BLCS) offers to users and applications the access to persistent block storage devices (virtual disks) that can be directly accessed and used as if they were raw physical disks. In this paper we derive ENIGMA, architecture for the back-end of BLCS systems able to provide adequate levels of access and transfer performance, availability, integrity, and confidentiality, for the data it stores. ENIGMA exploits LT rate less codes to store fragments of sectors on storage nodes organized in clusters. We quantitatively evaluate how the various ENIGMA system parameters affect the performance, availability, integrity, and confidentiality of virtual disks.**

These evaluations are carried out by using both analytical modeling (for availability, integrity, and confidentiality) and discrete event simulation (for performance), and by considering a set of realistic operational scenarios. Our results indicate that it is possible to simultaneously achieve all the objectives set for BLCS systems by using ENIGMA, and that a careful choice of the various system parameters is crucial to achieve a good compromise among them. Moreover, they also show that LT coding-based BLCS systems outperform traditional BLCS systems in all the aspects mentioned before.

***Keywords*: Engima, LT Rate less codes, Virtual disks.**

I. INTRODUCTION

The typical architecture of a BLCS system features front-end, providing users/applications with a set of abstract disk operations, and a back-end, that implements these operations over a set of physical storage resources. Traditional back-ends are typically implemented by aggregating a pool of volume servers, and by provisioning each VD on a single server.

This solution, however, is affected by performance and availability problems, since storage servers are a single point of failure, and may easily become a bottleneck. In this paper we investigate how a family of codes, known as Luby Transform (LT) codes, can be used as the enabling technology for BLCS systems able to solve the problems mentioned above. In coding-based storage system, the data units are first split into k equal size information fragments. These are subsequently encoded into n equally sized coded fragments (n k) such that a suitably-chosen subset of them suffices to reconstruct the sector.

Finally, the coded fragments are spread across a set of independent physical nodes. LT codes are rate less, that is the ratio k=n (the rate of the code) is not fixed at design-time, but can be instead adjusted at run time. Compared to alternative coding schemes (e.g., Reed-Solomon) LT codes are better suited to the needs of a BLCS system because of their much lower complexity, and their ability to adjust redundancy at run time.

Thanks to these properties, the usage of LT codes in the back-end of a BLCS system provides many benefits. In particular, in this paper we show that: low sector access time and high transfer throughput can be achieved by exploiting the simultaneous fetch of sector fragments from independent storage resources, thus exploiting the availability of many independent network paths; there is no single point of failure, as the coded fragments are stored on various independent storage resources, and lost fragments (caused by a failed resource) can be regenerated on-the-fly without the constraint of recreating exactly the lost data; suitable levels of confidentiality can be achieved by keeping secret the random generation process Using coding for cloud storage is not a new idea, and several works have studied in depth solutions to achieve each one of the goals of BLCS systems (i.e., adequate performance, availability, integrity, and confidentiality).

However, to the best of our knowledge, none of the existing works attempts to achieve all these goals at the same time. Simultaneously achieving all these goals, however, is not trivial, as they typically conflict with each other. For instance, as shown later, higher levels of

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 07
June 2017

availability require larger values of n, while higher levels of performance require lower values of n.

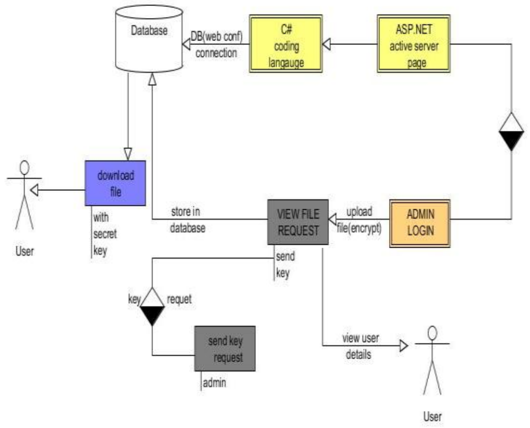
Therefore, several trade-offs emerge when all these objectives must be simultaneously achieved. To understand and evaluate these trade-offs, we quantitatively evaluate how the system parameters of ENIGMA affect the performance, availability, integrity and confidentiality of virtual disks.

These evaluations are carried out by using both analytical modeling (for availability, integrity, and confidentiality) and discrete event simulation (for performance), and by considering a set of realistic operational scenarios. Our results indicate that it is possible to simultaneously achieve all the objectives set forth for BLCS systems using ENIGMA, and that a careful choice of the various system parameters is crucial to achieve a good compromise among them. Moreover, they also show that coding-based BLCS systems outperform traditional BLCS systems in all the aspects mentioned before.

II. PROPOSED SYSTEM

In this paper we derived a suitable architecture for the back-end of BLCS systems that achieves adequate levels of access and transfer performance, availability, integrity, and confidentiality, for the data it stores. We exploited LT Rate less codes and showed how beneficial they are to all system properties we considered. The Rate less property allows to blindly spreading coded fragments to storage nodes in a cluster with the level of redundancy achieving the desired availability. Moreover, we devised a particular encoding strategy that for small block sizes guarantees zero decoding failure probability and improves availability for large values of the grouping factor x. Confidentiality is obtained by keeping the coding key secret, i.e., we assume that proxies are trusted and cannot be compromised. Furthermore the lower the values of x the higher the confidentiality.

Rate less codes allow for detection of polluted sectors and accurate and fast identification of malicious storage nodes. Furthermore, we show that after identification high recovery probability can be achieved especially for low values of x. As for the performance, we showed that much better performance than an equivalent-cost baseline system can be achieved even when caches are small and independently of the position of proxies in the network.



SYSTEM ARCHITECTURE

III. SDLC METHDLOGIES

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.

A preliminary design is created for the new system.

- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

- A second prototype is evolved by a fourfold procedure:

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 07
June 2017

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning and designing the second prototype.
4. Constructing and testing the second prototype.

- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

Conclusion

In this project I have derived a suitable architecture for the back-end of BLCS systems that achieves adequate levels of access and transfer performance, availability, integrity, and confidentiality, for the data it stores. We exploited LT rate less codes and showed how beneficial they are to all system properties we considered. In particular: the rate less property allows to blindly spreading coded fragments to storage nodes in a cluster with the level of redundancy achieving the desired availability. Moreover, we devised a particular encoding strategy that for small block sizes guarantees zero decoding failure probability and improves availability for large values of the grouping factor x. Confidentiality is obtained by keeping the coding key secret, i.e., we assume that proxies are trusted and cannot be compromised. Furthermore, the lower the values of x the higher the confidentiality. Rate less codes allow for detection of polluted sectors and accurate and fast identification of malicious storage nodes. Furthermore, we show that after identification high recovery probability can be achieved especially for low values of x. As for the performance, we showed that much better performance than an equivalent-cost baseline system can be achieved even when caches are small and independently of the position of proxies in the network. The future development foreseen for the current work is a thorough evaluation of the malicious storage nodes identification technique presented in to assess accuracy, robustness, and reactivity.

REFERENCES

[1] H. Dewan and R. Hansdah, "A survey of cloud storage facilities,"in IEEE SERVICES, jul 2011, pp. 224 –231.

[2] M. Bishop, Introduction to Computer Security. Addison-Wesley,2005.

[3] Jessica Bosari "The developing role of social media in the modern business world ", 2012.

[4] Toni Ahlqvist, Asta, Minna Halonen & Sirkka Heinonen "Social Media Roadmaps Exploring the futures triggered by social media", VTTTIEDOTTEITA- Research Notes 2454, ESPOO 2008.

[5] V. Shrividhya , R.Anitha "Evaluating Preprocessing Techniques in Text Categorization" International Journal of Computer Science & Application Issue , 2010.

[6] Jones S. and Paynter G. "Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications" Journal of the American Society for Information Science and Technology Volume 53, Issue 8, pages 653–677, 2002.

[7] Gutwin C, Paynter G, Witten I, Nevill-Manning C and Frank E. "Improving browsing in digital libraries with keyphrase indexes", Decision Support Systems 27(1–2), 81–104, 1999.

[8] Hulth A "Combining machine learning and natural language processing for automatic keyword extraction" Stockholm University, Faculty of Social Sciences, Department of Computer and Systems Sciences, 2004.