DOI : https://doi.org/10.56452/7-2-559

ISSN: 0974-5823 Vol. 7 No. 2 February, 2022

International Journal of Mechanical Engineering

Enabling Secure Data Integrity in Regenerating-Coding-Based Cloud Environment

Ayushi Jain

Asst. Professor, Department of Comp. Sc. & Info. Tech., Graphic Era Hill University, Dehradun, Uttarakhand India 248002

Abstract

A well-liked approach to distributed, effective, and profitable computation is called "cloud computing," in which shared, programmable resources are made available as a service through the Internet. The capacity to offer processing, storage, networks, along with other essential computing resources is referred to as cloud infrastructure as a service (IaaS) and is defined by the U.S. National Institute of Standards and Technology. The Future Internet (FI) architectures provide structures for assembling applications atop federated service primitives in order to handle networked cloud computing environments (NCE). This research introduces a hierarchical framework to investigate inter-domain resource mapping in a networked cloud environment. It suggests an Iterated Local Search (ILS) metaheuristic-based request splitting method and a Networked Cloud Mapping (NCM) method depending upon the approaches used to solve the intra-domain VNE problem. The suggested framework may be used to achieve a high degree of productivity across a significant volume of VN requests across networked cloud sizes including the least amount of processing time, according to thorough assessments.

1. INTRODUCTION

Cloud computing has been a well-liked approach for distributed, effective, and commercial compute during the past ten years. A computer paradigm known as "the cloud" makes use of the Internet to deliver shared customizable resources (such as networks, servers, processing resources, as well as applications) as a service. Notwithstanding the several paradigms that have arisen regarding the business philosophy embraced also relative research value, the fundamental objective is to build a fluid pool of virtual resources spanning PCs, servers, and data centres that allow users to access apps plus stored data whenever they need to.

The cloud Infrastructure as a Service (IaaS), which is the capacity to supply networks, processing, storage, and other crucial computer resources while allowing the consumer to deploy also run arbitrary software, including operating systems as well as applications, is one of the cloud service models described by the U.S. National Institute of Standards and Technology. The user does not have any administration or control over the underlying cloud infrastructure, although they do have some limited control over specific networking parts, operating systems, storage, even installed applications.

A networked cloud computing environment results from include intra- or inter-cloud communication in the resource mix (NCE). The Future Internet (FI) architectures, which provide frameworks for building applications atop federated service primitives, are addressed in the context of networked clouds. Typically, in such a setting, we make a distinction between a transit network provider and a cloud service provider, the latter of which offers inter-cloud connection.

It is crucial to effectively handle the resource mapping problem if cloud IaaS is to be provided with the least amount of administrative effort. The Virtual Network Embedding (VNE) problem, when seen from the perspective of a network virtualization environment, is the issue of mapping substrate (i.e. physical) resources (computing and communication) to Virtual Network (VN) demands. The resource mapping approach involves routing virtual links across physical links and mapping virtual nodes to physical host nodes. Nevertheless, the associated academic community has only just begun to examine the VNE problem in terms of various administrative domains. . In short, it creates new difficulties with I allocating the necessary resources along with numerous physical resources owned by various Cloud service Providers and (ii) connecting these resources together using suitable inter-cloud virtual network services.

When a hierarchical structure was implemented, we examine inter-domain resource mapping in a networked cloud environment in this research. The following are the main contributions of this study. In particular, a request splitting solution using the Iterated Local Search (ILS) metaheuristic is originally developed and geared to address the problem's intrinsic complexity and scalability. After that, a Networked Cloud Mapping (NCM) strategy built on the ideas used to solve the intra-domain VNE problem was chosen. Contrary to the majority of recent research, which assume either constant or random costs, we describe the related resource provisioning costs according to the amount of resources, such as due to resource limitations as well as the average use across a time window. This allows for a more accurate but also comprehensive synthesis of the accompanying resource mapping issue. The suggested framework may be used to achieve a high degree of effectiveness over several VN queries with networked cloud sizes leveraging the least amount of processing time, according to thorough assessments.

2. LITERATURE SURVEY

This article argues in favour of striping user data across many cloud storage providers and other RAID-like strategies. It describes RACS, a proxy that transparently distributes the storage workload across numerous providers, and assesses a system prototype. It also demonstrates how RACS may lower the cost of switching storage suppliers for a big organisation like the Internet Archive by seven times or more by altering erasure-coding parameters through the use of trace-driven simulations. RACS is a straightforward technological application that modifies market structure by letting customers choose how much they value vendor mobility vs overhead costs. Erasure coding has been used in RACS to address a different kind of failure than it is typically used for in storage systems. We have created a functional implementation, conducted microbenchmarks, and modelled bigger trials using actual trace data. Customers of cloud storage may better take advantage of new innovations in the quickly evolving cloud storage industry by exploring overhead and mobility trade-offs using RACS [1].

The use of cloud computing is anticipated to increase, therefore developers should consider it. In contrast to single node performance, horizontal scalability of virtualized resources should be the primary focus of cloud providers. Pay-per-use licencing should be used for both applications and infrastructure software. Hardware systems should be created at a container-scale, and operating costs should be in line with performance and purchase price. The memory hierarchy should include flash memory, processors should operate well with virtual machines, and LAN switches and WAN routers' cost and bandwidth performance must increase [2].

This article offers a methodology for distant data verification called proven data possession (PDP). By selecting random groups of blocks from the server, it creates probabilistic proofs of possession while substantially lowering I/O costs. By sending a modest, regular amount of data, the challenge/response protocol minimises network communication. The model has features for mitigating arbitrary levels of data corruption, is lightweight, and supports massive data sets in distributed storage systems. In comparison to earlier approaches, two provably secure PDP techniques are more effective and have less server overhead. The effectiveness of PDP has been tested by experiments utilising our implementation, which also shows that disc I/O, not cryptographic computing, is what limits PDP's speed. The tradeoffs in speed, security, also space overheads when including resilience into a remote data verification technique are investigated through a thorough experimental examination [3].

A group of servers can use the distributed cryptography system HAIL (High-Availability and Integrity Layer) to demonstrate to a client that a stored file is legitimate and retrievable. It is quickly computed, compact, cryptographically checks file shares, reallocates them as needed, and is resistant to a persistent, moving attacker. HAIL consolidates, explicitly integrates, and simplifies many distributed-systems and cryptography techniques. It also suggests a rigorous analysis and parameter selection process as well as a robust, formal adversarial model for HAIL. HAIL enhances the effectiveness and security of current technologies, such as Proofs of Retrievability (PORs) installed on specific servers [4].

A compact demonstration that a target file F is intact is known as a proof of retrievability (POR), and it is an appealing building component for high-assurance remote storage systems. The constructs of Juels-Kaliski and Shacham-Waters that were previously suggested are improved in this paper's theoretical framework for the design of PORs, which also clarifies the conceptual shortcomings of earlier theoretical models. With the exception of the requirement that the adversary's error rate be constrained when the client tries to extract F, it supports a completely Byzantine adversarial paradigm. Our methods enable effective protocols across the whole range of 2, up to 2 non-negligibly near to 1.0, and they even show usable encoding for files F with sizes larger than client main memory [5].

3. PROPOSED SYSTEM

It's crucial to store all data on a single server when dealing with the single point-of-failure issue. Get the data from the other servers if the server has failed. Likewise, you should rebuild the damaged data from the failed server and upload it to a new one. For the same fault tolerance level, erasure coding uses less storage than replication. The server suffers a permanent loss of data as a result. In archival storage, even small failure rates can cause considerable data loss. These techniques are limited to data corruption detection. The original data is not recovered. When a server has a data loss, clients cannot access the information. Finding the original data takes a lot of time.

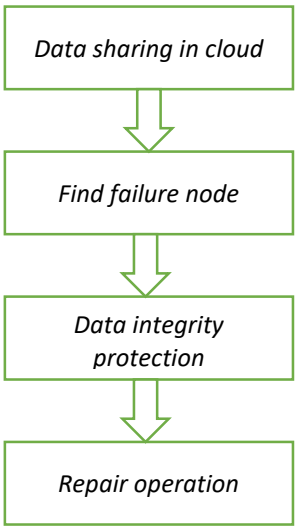

Fig 1: System Architecture

In order to reduce network repair traffic, regenerating codes are employed. During the repair period, it does not read and recreate the entire file. To recreate just the missing data, it reads a collection of chunks from the other servers that are smaller than the original file. FMSR codes, or functional minimum-storage regenerating codes, enable clients to remotely check the accuracy of the data on the server. All that is required of the servers is basic read and write functionality. It makes it possible for cloud storage to provide integrity protection, fault tolerance, and effective recovery. Data integrity is checked using the MAC technique. The following are some of the suggested system's many benefits:

- It offers a cheap way to keep the data stored in the cloud up to date.
- On the cloud, there is extremely little time for data integrity checks.
- The mathematical model examines the security.
- Clients can quickly access data from the server.

The phases of implementing the suggested strategy are explained in the sections below:

1. Data sharing in cloud

The project's initial phase is the creation of the customer. The data are uploaded to the cloud by utilising the client. The client then chooses the specific data that needs to be stored. The client uploads data, which is then received by the cloud. After separating, the data is then stored on cloud servers. This is carried out for security reasons. This data may be obtained by the customers on any time. This data can include the client's private information. As a result, data security is crucial. The divided data is kept on many cloud servers.

2. Find failure node

Different client data is stored on each cloud server. Moreover, the servers' capacities vary as well. The data stored on the server is lost if any of the servers fail. There is no way to retrieve the data from a failed server. The drawback of the current system is that. Finding the failed node is crucial in our proposed system, though, as it allows for the retrieval of the data. Details about the failing server are sent to the other cloud servers. You can obtain the data by identifying the failing node.

3. Data integrity protection

It's crucial that the data on the cloud server be accurate. This is a different kind of security. Integrity confirms that the data variations exist. The Message Authentication Code verifies each piece of data's integrity. The secure hash algorithm (SHA-1) is employed in the message authentication code. Compared to MD5, it is superior. The cloud server can determine whether the attacker changed any data by looking for these changes. Due to the message authentication code, this has happened. in order for it to be utilised for security purposes.

4. Repair operation

The data on the malfunctioning server is recovered during the repair process. It can be acquired by obtaining the same information from other servers. Since the data is kept on several cloud servers. Not all of the data is kept on different servers. The divided data is kept on many cloud servers. The server will then check the data's integrity after that. And give the information to the client who requests it. Integrity verification is crucial in this regard.

4. RESULTS

A well-liked approach to distributed, effective, and profitable computation is called "cloud computing," in which shared, programmable resources are made available as a service through the Internet. In a networked cloud environment, this research investigates inter-domain resource mapping using a hierarchical framework. Clients can benefit from outsourcing their data to the cloud, but they must ensure the accuracy of their data. Mathematical modelling improves the security strength of FMSR-DIP codes, which are employed for their fault tolerance and traffic-saving qualities.

DOI : https://doi.org/10.56452/7-2-559

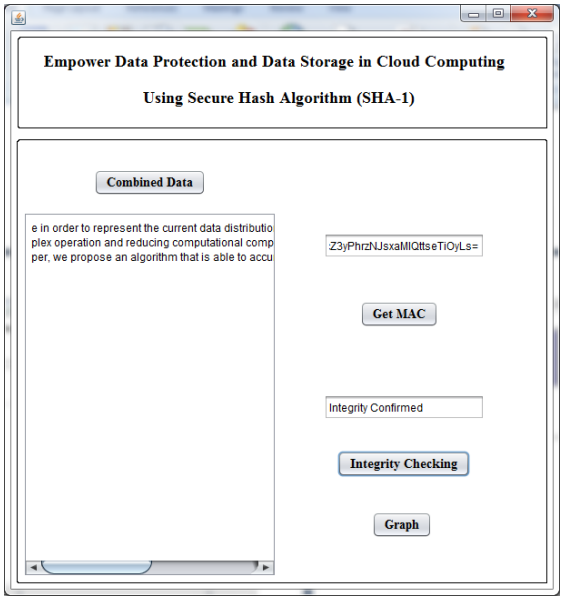

Fig 2: Integrity Confirmation

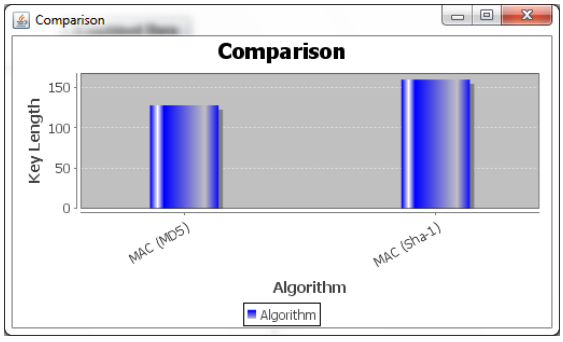

Fig 3: Comparative Analysis

5. CONCLUSION

Clients can benefit greatly from outsourcing their data to the cloud, but they must ensure the security of that data. The multiserver system uses the Data Integrity Protection Scheme for FMSR codes. For FMSR codes' fault tolerance and traffic-saving repair capabilities, FMSR-DIP codes are employed. The FMSR-security DIP's capabilities have significantly enhanced. The mathematical modelling evaluates it.

6. FUTURE ENHANCEMENT

In the message scheduling of its compression function, SHA-1 and SHA-0 only vary by a single bitwise rotation. This change was made to address a bug in the original algorithm that decreased its cryptographic security. Both SHA-0 and SHA-1 have been found to have vulnerabilities, although SHA-1 seems to be more resistant to assaults, corroborating the NSA's claim that the upgrade boosted security.

REFERENCE

[1] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: ACase for Cloud Storage Diversity," Proc. First ACM Symp. CloudComputing (SoCC '10), 2010.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A.Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M.Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53,no. 4, pp 50-58, 2010.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L.Kissner, Z. Peterson, and D. Song, "Remote Data Checking UsingProvable Data Possession," ACM Trans. Information and SystemSecurity, vol. 14, article 12, May 2011.

[4] K. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availabilityand Integrity Layer for Cloud Storage," Proc. 16th ACM Conf.Computer and Comm. Security (CCS '09), 2009.

[5] K. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability:Theory and Implementation," Proc. ACM Workshop Cloud ComputingSecurity (CCSW '09), 2009.

[6] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote DataChecking for Network Coding-Based Distributed Storage Systems,"Proc. ACM Workshop Cloud Computing Security (CCSW '10),2010.

[7] H.C.H. Chen and P.P.C. Lee, "Enabling Data Integrity Protectionin Regenerating-Coding-Based Cloud Storage," Proc. IEEE 31stSymp. Reliable Distributed Systems (SRDS '12), 2012.

[8] L. Chen, "NIST Special Publication 800-108," Recommendation forKey Derivation Using Pseudorandom Functions (Revised), http://csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf, Oct.2009.

[9] R. Curtmola, O. Khan, and R. Burns, "Robust Remote DataChecking," Proc. ACM Fourth Int'l Workshop Storage Security andSurvivability (StorageSS '08), 2008.

[10] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP:Multiple-Replica Provable Data Possession," Proc. IEEE 28th Int'lConf. Distributed Computing Systems (ICDCS '08), 2008.