M.Nithya Int. Journal of Engineering Research and Applications www.ijera.com
ISSN : 2248-9622, Vol. 4, Issue 12(Part 6), December 2014, pp.183-187

RESEARCH ARTICLE OPEN ACCESS

A Study of A Method To Provide Minimized Bandwidth Consumption Using Regenerating Codes In Cloud Storage

[1]M.Nithya, [2]K.Sasireka
[1]Master of Computer Science and Engineering, Muthayammal Engineering College, Rasipuram, India
[2]Assistant Professor, Department of Computer Science and Engineering, Muthayammal Engineering College, Rasipuram, India

Abstract-
Cloud storage systems to protect data from corruptions, redundant data to tolerate failures of storage and lost data should be repaired when storage fails. Regenerating codes provide fault tolerance by striping data across multiple servers, while using less repair traffic than traditional erasure codes during failure recovery. In previous research implemented practical Data Integrity Protection (DIP) scheme for regenerating-coding based cloud storage. Functional Minimum-Storage Regenerating (FMSR) codes and it construct FMSR-DIP codes, which allow clients to remotely verify the integrity of random subsets of long-term archival data under a multi server setting. The problem is to optimize bandwidth consumption when repairing multiple failures. The cooperative repair of multiple failures can help to further save bandwidth consumption when multiple failures are being repaired.

Key Terms: Cloud computing, Minimum storage, Bandwidth consumption.

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the "Software as a Service" (SaaS) computing architecture, are transforming data centres into pools of computing service on a huge scale. Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centers. Although envisioned as a promising service platform for the Internet, this new data storage paradigm in "Cloud" brings about many challenging design issues which have profound influence on the security and performance of the overall system. One of the biggest problem in existing method is it takes more bandwidth For repairing multiple failures. so overcome this problem we use functional minimum bandwidth cooperative regenerating method for providing minimized bandwidth consumption. Consider the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files.

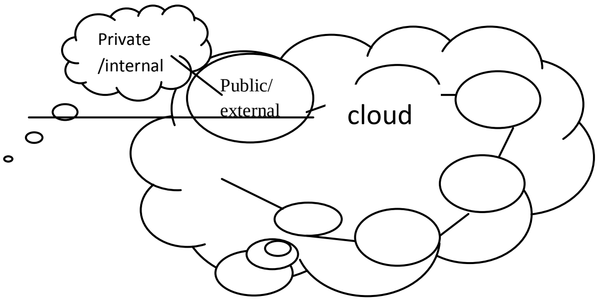

Fig.1.1 Cloud Infrastructure

In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability. Although schemes with private auditability can achieve higher scheme efficiency, public auditability allows any one, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent Third Party Auditor (TPA), without devotion of their computation resources. In the cloud, the clients themselves are unreliable or may not be able to afford the overhead of performing frequent integrity checks. Thus, for practical use, it seems more rational to equip the verification

Hybrid

protocol with public auditability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. Moreover, for efficiency consideration, the outsourced data themselves should not be required by the verifier for the verification purpose.

1.1 PRIVATE CLOUD

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users “still have to buy, build, and manage them” and thus do not benefit from less hands-on management, essentially “[lacking] the economic model that makes cloud computing such an intriguing concept”.

1.2 PUBLIC CLOUD

A cloud is called a “public cloud” when the services are rendered over a network that is open for public use. Public cloud services may be free or offered on a pay-per-usage model. Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called “AWS Direct Connect” and “Azure ExpressRoute” respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

1.3 HYBRID CLOUD

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. A hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers. A hybrid cloud service crosses isolation and provider boundaries so that it can’t be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

II. LITERATURE SURVEY

In this paper, they develop a new cryptographic building block known as a proof of retrievability (POR). A POR enables a user (verifier) to determine that an archive (prover) “possesses” a file or data object. More precisely, a successfully executed POR assures a verifier that the prover presents a protocol interface through which the verifier can retrieve a file in its entirety. Of course, a prover can refuse to release a file even after successfully participating in a POR. A POR, however, provides the strongest possible assurance of file retrievability barring changes in prover behavior. As they demonstrate in this paper, a POR can be efficient enough to provide regular checks of file retrievability. Consequently, as a general tool, a POR can complement and strengthen any of a variety of archiving architectures, including those that involve data dispersion

In this paper, we explore a unification of the two approaches to remote file-integrity assurance in a system that we call *HAIL (High-Availability and Integrity Layer)*. HAIL manages file integrity and availability across a collection of servers or independent storage services. It makes use of PORs as building blocks by which storage resources can be tested and reallocated when failures are detected HAIL does so in a way that transcends the basic single server design of PORs and instead exploits both within server redundancy and cross-server redundancy.HAIL relies on a single trusted verifier—e.g., a client or a service acting on behalf of a client—that interacts with servers to verify the integrity of stored files. (We do not consider a clientless model in which servers perform mutual verification, as for distributed information dispersal algorithms)

Our PDP schemes provide data format independence, which is a relevant feature in practical deployments and put no restriction on the number of times the client can challenge the server to prove data possession. Also, a variant of our main PDP scheme offers public verifiability To enhance possession guarantees in our model, the system define the notion of robust auditing, which integrates forward Error-Correcting Codes (FECs) with remote data checking. Attacks that corrupt small amounts of data do no damage because the corrupted data may be recovered by the FEC. Attacks that do unrecoverable amounts of damage are easily detected, since they must corrupt many blocks of data to overcome the redundancy. The system identify the requirements that guide the design, implementation, and parameterization of robust auditing schemes. Important issues include the choice of an FEC code, the organization or layout of the output data, and the selection of encoding parameters. The forces on this design are subtle and complex. The integration must maintain the security of remote data checking, regardless of the adversary's attack strategy and regardless of the access pattern to the original data. The integration must also maximize the encoding rate of data and the I/O performance of the file on remote storage, and minimize storage overhead for redundancy and the I/O complexity of auditing remote data.

III. METHODOLOGY

Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications. CloudSim goal is to provide a generalized and extensible simulation framework that enables modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services, allowing its users to focus on specific system design issues that they want to investigate, without getting concerned about the low level details related to Cloud-based infrastructures and services.

3.1 EXISTING SYSTEM

Proof of Retrievability (POR) and Proof of data possession (PDP) is to verify the integrity of a large file by spot checking only a fraction of the file via various cryptographic primitives and originally proposed for the single-server case.Data integrity checks to a multi server setting using replication and erasure coding, respectively. Reed- Solomon codes has a lower storage overhead than replication under the same fault tolerance level. Regenerating codes have recently been proposed to minimize repair traffic the amount of data being read from surviving servers. It achieve this by not reading and reconstructing the whole file during repair as in traditional erasure codes, but instead reading a set of chunks smaller than the original file from other surviving servers and reconstructing only the lost or corrupted data chunks. Functional minimum-storage regenerating (FMSR) codes and it construct FMSR-DIP codes, which allow clients to remotely verify the integrity of random subsets of long-term archival data under a multi server setting. FMSR-DIP codes preserve fault tolerance and repair traffic saving as in FMSR codes.The Problem is to optimizing more bandwidth consumption when repairing multiple failiures.

3.2 PROPOSED SYSTEM

In this work extended FMSCR (Functional Minimum Storage Co-operative Regenerating) –DIP codes one newcomer during repair, the corresponding families of cooperative regenerating codes are specifically termed as MSR and MBR codes as well. Though MBCR codes achieve the minimum repair bandwidth, they sacrifice the storage efficiency since each coded block contains more than 1k of the original data, while MSCR codes belong to the family of MDS codes by achieving both the optimal storage efficiency and the recoverability property. The original storage-bandwidth tradeoff that defines the family of regenerating codes is derived under the assumption that repairs are considered individually, node failures are repaired one by one in different rounds of repairs. The repair bandwidth can have a better lower bound with multiple newcomers that cooperate with each other. the storage have independently identified the storage-bandwidth tradeoff with multiple cooperative newcomers during repair and the erasure codes achieving such repair bandwidth in this tradeoff are termed as cooperative regenerating codes.Suppose that there are t newcomers during repair, the storage and the repair bandwidth per newcomer of Functional minimum-storage cooperative regenerating (MSCR) codes and minimum-bandwidth cooperative regenerating (MBCR) codes.

3.2.1 Creation of Cloud Environment

Cloud Storage contains three stages Data owners (owner), the cloud server (server), and the third-party auditor (auditor). The owners create the data and host their data in the cloud. The cloud server stores the owners' data and provides the data access to users (data consumers). The auditor is a trusted third-party that has expertise and capabilities to provide data storage auditing service for both the owners and servers. The auditor can be a trusted organization managed by the government, which can provide unbiased auditing result for both data owners and cloud servers.

3.2.2 FMSR- DIP Code Implementation

A systematic adversarial error-correcting code (AECC) to protect against the corruption of a chunk, In conventional error-correcting codes (ECC), when a large file is encoded, it is first broken down into smaller stripes to which ECC is applied independently. AECC uses a family of PRPs as a building block to randomize the stripe structure so that it is computationally infeasible for an adversary to target and corrupt any particular stripe. Both FMSR codes and AECC provide fault tolerance. The difference is that apply FMSR codes to a file striped across servers, while we apply AECC to a single code chunk stored within a server.

The cryptographic primitives stated and define per-file secret keys kENC, kPRF, kPRP, and kMAC for the encryption, PRF, PRP, and MAC operations, respectively. The usage of these keys should be clear from the context and are omitted below for clarity. Also, implement AECC as an (n,k) error-correcting code, which encodes k fragments of data into n fragments, such that errors up to b(n – k)/2, or erasures up to n - k can be corrected. A row as a collection of all bytes that are at the same offset of all native chunks or FMSR code chunks. The rth rows of the native chunks and the FMSR code chunks correspond to the bytes

$$\{F_{jr}\}_{1\leq j\leq k(n-k)} \text{ and } \{P_{ir}\}_{1\leq i\leq n(n-k)}$$

respectively. See that the rth row of each of the FMSR code chunks is encoded by the rth row of the native chunks. That is, can construct the rth row of the FMSR code chunk Pi, Each FMSR code chunk Pi from NCCloud is encoded by FMSR-DIP codes into Pi . The rth row of the FMSR-DIP code chunks corresponds to the bytes

$$\{P'_{ir}\}_{1\leq i\leq n(n-k)}.$$

3.2.3 FMSCR and FMBCR Implemention

An instance of [n, k, d] exact MSR codes (d ! k + 1), we can instantly construct an instance of [n, k, d−1, t = 2] exact MSCR codes. This means that we have broadly expanded the parameters of exact MSCR codes that have explicit constructions. To our best knowledge, there exists a construction of exact MSR codes as long as d ! 2k − 2 [12]. Therefore, if n ! d + 1 and d ! k, the system can get [n, k, d, t = 2] exact MSCR codes as long as d ! 2k − 3. On the other hand, the system also show that given a scalar construction of linear exact MSCR codes, the system can derive a construction of exact MSCR codes. Since we know that there exists no scalar construction of [n, k, d] linear exact MSR codes when d < 2k − 3 and k > 3 [17], it is impossible to have scalar construction of [n, k, d, t = 2] exact MSCR codes when d < 2k − 4 and k > 3. To summarize, discuss the construction of [n, k, d, t = 2] exact MSCR codes for all possible values of [n, k]. The existence or the non-existence of linear scalar constructions of all possible values of d, except the only open case of d = 2k − 3, where k > 3.Because the FMSCR codes constructed in derived from MSR codes, we can directly inherit advantages of the corresponding instance of MSR codes. The most straightforward and important advantage inherited from MSR codes is that the derived instance of MSCR codes can be systematic, i.e., the original data are embedded into coded blocks. Systematic MSCR codes can help to significantly reduce the access latency and the exact repair makes the repaired data remain systematic after any rounds of repair procedures.

3.2.4 Dynamic Operation

The basic file operations Upload, Download, and Repair of NC Cloud with the DIP feature. During upload, FMSR-DIP codes expand the code chunk size by a factor of n/k (due to the AECC). During download and repair, FMSR-DIP codes maintain the same transfer bandwidth requirements (with up to a small constant overhead) when the stored chunks are not corrupted. Also, we introduce an additional Check operation, which verifies the integrity of a small part of the stored chunks by downloading random rows from the servers and checking their consistencies.

IV. CONCLUSION

In this work extended FMSCR (Functional Minimum Storage Co-operative Regenerating) –DIP codes are used to provide minimum bandwidth and storage space when newcomer during repair, the corresponding families of cooperative regenerating codes are specifically termed as MSR and MBR codes as well. Though MBCR codes achieve the minimum repair bandwidth, they sacrifice the storage efficiency since each coded block contains more than 1k of the original data, while MSCR codes belong to the family of MDS codes by achieving both the optimal storage efficiency and the recoverability property.

REFERENCES

[1] AtenieseG, Burns.R, Curtmola.R, Herring.J, Khan.O, Kissner.L, Peterson.Z, and Song.D, “Remote Data Checking Using Provable Data Possession,” ACM Trans. Information and System Security, vol. 14, article 12, May 2011.

[2] Bowers.K, Juels.A, and Oprea.A, “Proofs of Retrievability: Theory and Implementation,”

Proc. ACM Workshop Cloud Computing Security (CCSW ’09), 2009.

[3] Bowers.K, Juels.A, and Oprea.A, “HAIL: A High-Availability and Integrity Layer for Cloud Storage,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS ’09), 2009.

[4] Dimakis.A, Godfrey.P, Y. Wu, M. Wainwright, and K. Ramchandran, “Network Coding for Distributed Storage Systems,” IEEE Trans. Information Theory, vol. 56, no. 9, 4539-4551, Sept. 2010.

[5] Krawczyk.H, “Cryptographic Extraction and Key Derivation: The HKDF Scheme,” Proc. 30th Ann. Conf. Advances in Cryptology (CRYPTO ’10), 2010.

[6] Shacham.H and Waters.B, “Compact Proofs of Retrievability,” Proc. 14th Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, J. Pieprzyk, ed., pp. 90-107, 2008.

[7] Schroeder.B, Damouras.S, and Gill.P, “Understanding Latent Sector Errors and How to Protect against Them,” Proc. USENIX Conf. File and Storage Technologies (FAST ’10), Feb. 2010.

[8] Vrable.M, Savage.S, and Voelker.G, “Cumulus: Filesystem Backup to the Cloud,” Proc. USENIX Conf. File and Storage Technologies (FAST), 2009.

[9] Wildani.A, Schwarz T.J.E., Miller E.L., and Long D.D, “Protecting Against Rare Event Failures in Archival Systems,” Proc. IEEE Int’l Symp. Modeling, Analysis and Simulation Computer and Telecomm. Systems (MASCOTS ’09), 2009.

[10] E. Naone, “Are We Safeguarding Social Data?”http://www.technologyreview.com/blog/editors/22924/, Feb. 2009.

[11] J.S. Plank, “A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-Like Systems,” Software - Practice & Experience, vol. 27, no. 9, pp. 995-1012, Sept. 1997.

[12] M.O. Rabin, “Efficient Dispersal of Information for Security, LoadBalancing, and Fault Tolerance, ” J. ACM, vol. 36, no . 2, pp. 335-348, Apr. 1989.

[13] I. Reed and G. Solomon, “Polynomial Codes over Certain Finite Fields,” J. Soc. Industrial and Applied Math., vol. 8, no. 2, pp. 300-304, 1960.

[14] B. Schroeder, S. Damouras, and P. Gill, “Understanding Latent Sector Errors and How to Protect against Them,” Proc. USENIX Conf. File and Storage Technologies (FAST ’10), Feb. 2010.

[15] B. Schroeder and G.A. Gibson, “Disk Failures in the Real World:What Does an MTTF of 1,000,000 Hours Mean to You?” Proc. Fifth USENIX Conf. File and Storage Technologies (FAST ’07), Feb. 2007.

[16] T. Schwarz and E. Miller, “Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage,”Proc. IEEE 26th Int’l Conf. Distributed Computing Systems, (ICDCS ’06), 2006.

[17] H. Shacham and B. Waters, “Compact Proofs of Retrievability,”Proc. 14th Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT ’08),2008.

[18] “TechCrunch,” Online Backup Company Carbonite Loses Customers’Data, Blames and Sues Suppliers, http://techcrunch.com/2009/03/ 23/online – backup – company – carbonite - loses customers-datablames- and-sues-suppliers/, Mar. 2009.

[19] M. Vrable, S. Savage, and G. Voelker, “Cumulus: Filesystem Backup to the Cloud,” Proc. USENIX Conf. File and Storage Technologies (FAST), 2009.

[20] “Watson Hall Ltd,” UK Data Retention Requirements,https://www.watsonhall.com/resources/ downloads / paper - uk-dataretention - requirements.pdf, 2009.

[21] A. Wildani, T.J.E. Schwarz, E.L. Miller, and D.D. Long, “Protecting Against Rare Event Failures in Archival Systems,” Proc. IEEE Int’l Symp. Modeling, Analysis and Simulation Computer and Telecomm.Systems (MASCOTS ’09), 2009.