Evolution of Multi Cloud Framework for Integrity, Confidentiality and Availability

D.V.Bhavsagar[1], Dr. Vinay Chavan[2], Dr. S. J. Sharma[3]
[1]*Assistant professor , Department of Computer Science, S. K. Porwal College, Kamptee(Nagpur)*
[2]*Associate professor And H.O.D , Department of Computer Science,S. K. Porwal College, Kamptee (Nagpur)*
[3] Dr. S. J. Sharma, P*rofessor And H.O.D , Department of Electronics R.T.M. Nagpur University, Nagpur*

Abstract— Data became most valuable corporate assets which demand to search safest place just like locker to store, manage and accessible whenever required. Many IT enterprises has burden of additional cost of management of storage resources for their valuable data. Exponential growth of data due to digitization for the business in last five years gave birth to problem of data availability, integrity and security. To resolving data growth many enterprises demand out source data storage on cloud which rise various data security issues i.e. vender lock-in, inter portability, availability, scalability, privacy, confidentiality and many more. The cloud provider intentionally not taking care of stored data or deliberately delete the rarely accessed file to save money and storage space which causes serious issues of availability of data in cloud. Data owner loss their control of data while outsourcing in cloud and incur privacy violations. Multi-tenancy feature in multi cloud architecture allowed customer to share various resources but rise flooding attacks, massive data and intense computations give raised to cross VM attacks, data beaches, enhanced computational and communications overheads. These issues strongly motivate to evaluate multi cloud framework for integrity, confidentiality and availability of data in cloud. In this paper we critically analysis HAIL architecture, single and distributed proxy architecture, DepSkp multi cloud system, Iris system architecture, cloud RAID architecture and NC Cloud to verify integrity, confidentiality and availability of data in multi cloud and admire that there is strong need to design mechanism to secure dada in multi cloud.

Keywords— *Multi cloud framework, Cloud integrity, availability, confidentiality, Scalability, privacy, data reliability, distributed data file, data transfer.*

I. INTRODUCTION

Digitization in every business has resulted into explosive growth of data. Big data is growing beyond our imagination owing to the paradigm shift in Technology. Cloud and Internet of Things (IoT) technologies have added to the fourfold growth of data. It is observed that the growth has tripled in just five years. This data is stored in various forms like structured unstructured and semi-structured. Recently the IBM survey has published that the 90% of this growth has occurred in just last two years. It is glaringly evident that the data is growing exponentially [1, 2, 3]. Apart from managing human resources of a company, there is a need to manage other tangible and intangible resources. Most valuable corporate asset is the data being generated through various devices. Data Storage is classified into three types like File Storage, Block Storage and Object Storage based on method of storing and accessing data. Since the dawn of the social media like Facebook, Twitter and Whatsapp most of the data is in digital form. This data in an unstructured format had issues of storing and retrieving the information. One of the methods is proposed to resolve these issues in [5]. This is the best method to store this type of data and was considered to be the object storage format. The difference between each type of storage is described in [6] as shown in fig 1

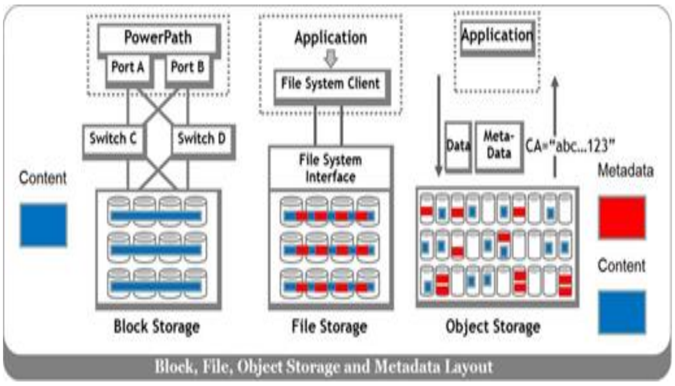

Fig 1 Block, File, Object Storage and Metadata Layout[6]

Bhuyya et al. have proposed methods to outsource the data storage on to the cloud to meet the growing demand for storage. However, the multi-tenancy and virtualization feature of cloud gave rise to tremendous security issues. Atenieseet al. have proposed solutions to address the problem of providing privacy to the clients, data stored on the cloud. The cloud provider in order to save money and storage space may neglect the data stored or deliberately delete the rarely accessed files [7,8,9]. This poses a serious problem to the users. Several attempts have been made to design solutions that meet these requirements. Outsourcing will make the owner of the data to lose control on his data and incur privacy violations. Multi-tenancy will allow multiple cloud customers to share various resources which would result in flooding attacks. Massive data and intense computations may give rise to cross VM attacks, data breach, enhanced computational and communication overheads. Zhu and Warneke [10,11,12] have proposed to verify the integrity of the data which is stored remotely by hashing the entire set which is practically impossible. Hence, there is a need to explore a methodology to improve data integrity, data confidentiality and data availability in cloud computing environment [13,14,15,16]. In order to resolve the vendor-lock-in issue there is a need to enhance the interoperability in cloud [17,18,19]. Protecting private information details like credit card information, health records of the patients in any health solutions from intruders or hackers is of prime importance [20,21,22] .

II. MULTI-CLOUD FRAMEWORKS

Multi-cloud solves all security issues related to data storage, and also address concerns about data availability and vendor lock-in [23,24,25,26]. Some of the multi-cloud solutions are RACS, DEPSKY, MCDB [27], cloud-RAID [13,14,28]. Although multi-cloud has evolved since 2014, few key reference papers, which are relevant to the work, are critically reviewed.

High-Availability and Integrity Layer

Bowers et al. in his paper [29] introduces HAIL (High-Availability and Integrity Layer) in the year 2015. In order to check remote file integrity assurance in a system it is required to check both within server redundancy and cross server redundancy. It makes use of PORs as building blocks in order to store resources, which can be tested and reallocated when any failure occurs in one of the servers. It relies on a single trusted verifier. Here a client or a service can act on behalf of a customer who is interested to interact with the server and verify the integrity of the stored files. HAIL is based on new protocol design called as TAR (Test and Redistribute). As explained before TAR uses PORs to detect any data or file corruption and reallocate the resources when needed. Once a fault is detected via challenge response in a given server the client communicates with the other server. This communications helps the client to recover the corrupted shares from the encoded files distributed on other servers for cross-server redundancy. It later resets the faulty server with the correct share. The main contribution of this paper HAIL is dispersal code. This code is used to spread the file blocks across the servers. In order to distribute the blocks a new cryptographic method is proposed which is a combinations of Pseudo Random Function (PRF) , Error Correcting Code (ECC) and Universal Hash Function (UHF).This primitive is called as integrity Protected Error Correcting Code (IP –ECC). In this, the storage overhead is minimal which is equal to one extra code word symbol.

Single and distributed proxy architecture

Hussain and Leionne have proposed proxy based data distributed architecture as shown in fig 2.2 for object storage [27]. Authors have proposed single proxy and distributed proxy architecture to implement multi-cloud for object storage as shown in fig 2.2 and fig 2.3 respectively [27]. It is cloud storage proxy. In this system the data is striped between multiple cloud storage providers. The main contribution of RACS is reductions of onetime cost of switching between storage providers at the cost of additional operational overhead. RACS exposes minimal interface like put, get delete and list of the applications

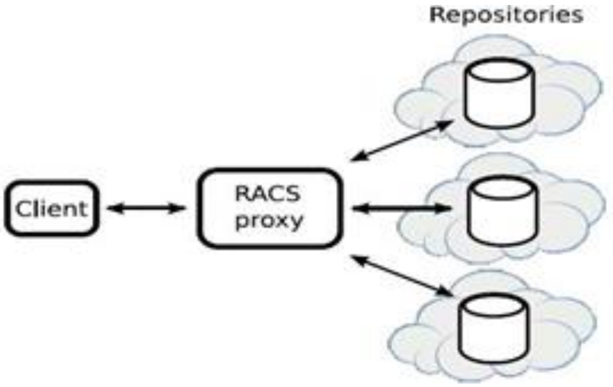

Fig 2 RACS Single- Proxy Architecture [16]

When the data needs to be stored on multiple storage providers it needs to pass through the proxy for encoding and decoding of the data. A single proxy could prove to be a bottleneck. In order to resolve this RACS is run as a distributed system with many proxies which simultaneously communicate to the same repository as shown in fig 2.3 [27]. One of the main problem observed in Amazon S3 is, simultaneous read operation may return some combinations of old and new shares. This problem is addressed in RACS by coordinating their actions with one writer and many readers synchronization for each pair of bucket and key. RACS makes use of Apache Zookeeper for the distribution of data on multiple storage repositories, as it provides distributed synchronization primitives. These primitives provide autonomic operations for Abstract Data types (ADT's) and for manipulating distributed tree structures. Only S3 cannot build this primitive due to its consistency deficiency

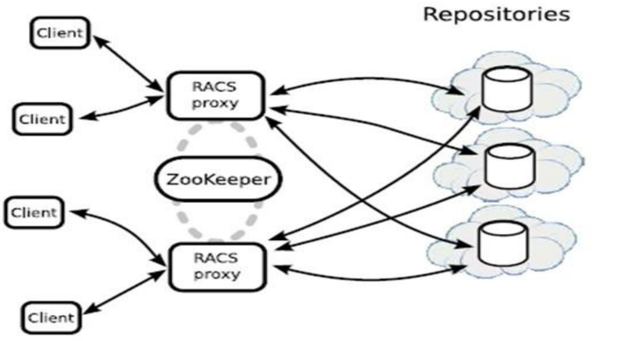

Fig 3 Multiple RACS Proxies coordinate their actions using Zookeeper Failure Recovery [16]

The price increase of any CSP can be an economical failure, which can be anticipated ahead of time. The administrators of the RACS will move the data from such a CSP much before the failure occurs. It redirects the put request from the failed repository to the new repository. It will not use the failed repository for get requests. It saves download and then upload of 1/m of the total object data. Thus, the authors claim that the system is economical.

C. DepSky Multi cloud system

Bessani et al. has proposed another Multi-Cloud system called as DepSky [28].This system primarily addresses only availability and confidentiality of the data stored in multiple cloud service providers which are part of a Multi-Cloud construction. The use of Byzantine quorum system protocol, erasure codes and cryptographic secret sharing algorithms ensure data security.

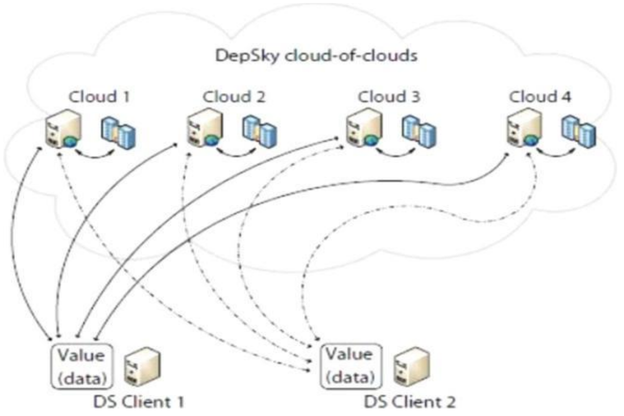

Fig 4 DepSky Architecture[28]

This architecture uses only four service providers which use their own particular interface. This software library will be installed on the client machine as shown in fig[28] which wishes to communicate with the cloud. There is no need to install anything on the clouds, as they are storage clouds. The software installed on the client's machine permits to carry out read and write operations on these storage clouds. This software handles the Application Program Interface (APIs) of individual clouds. Every cloud has its own API to handle its own format of data storage. Hence, DepSky consists of three levels namely conceptual data unit, a generic data unit and the data unit implementation. This system also has three parts namely readers, writers and storage providers. It is important to understand the role of these three parts. There are various means for the readers to fail and hence failure frequency is more in case of readers. On the other hand, the writers fail only due to crashing and hence their failure rate is comparatively less. Availability of data improves due to replication of information on multiple service providers using Byzantine quorum protocol. DepSky–A fails to address confidentiality. However, DepSky-CA proposes to solve this problem in the revised version

D. Iris System Architecture

Stefnovet al. designed an authenticated file system which supports workloads from large enterprises for storing the data in the cloud. It assures the users that the data is safe against the untrustworthy cloud service providers. This system guarantees strong integrity. Iris allows the enterprise to maintain a large file system in the cloud. In addition to data integrity, it provides data freshness and data availability in case of any CSP failure [30].

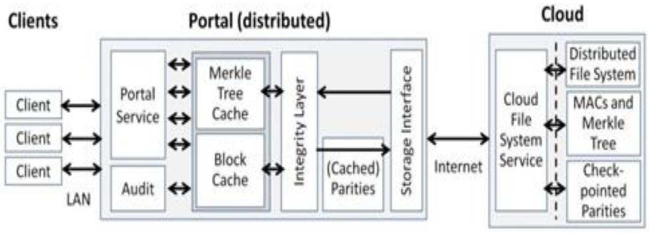

Fig 5 Iris System Architecture[30].

The recently accessed data by the enterprise clients is cached in this portal. This portal is also responsible for checking the availability of the data at any given time and its freshness. It also checks the integrity and the authenticity of the data, which is retrieved from the cloud. This also caches the error correcting information for the entire file system. Whenever corruption of data is identified by this portal through the auditing protocol, it enables the client to recover the modified or corrupted data.

E. Cloud-RAID Architecture

Schnjakinet al. proposed an approach that deals with above mentioned problems. This solution addresses privacy by separating the data into unrecognizable chunks and distributes it on multiple cloud service providers [31,14] . Initially in cloud RAID (Redundant Array of Inexpensive Disks), it was considered that a data object is completely transferred only when all the chunks are placed in the CSP's shown in fig 2.6[13].

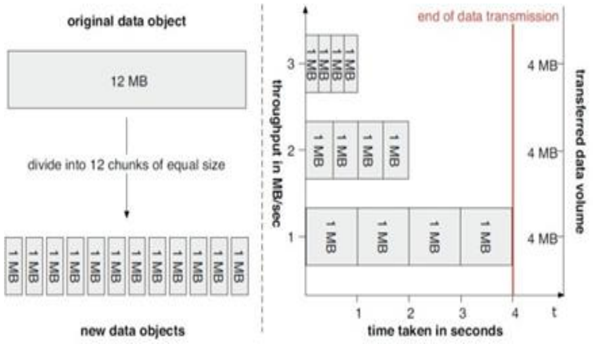

Fig 6 Cloud-RAID Architecture[13]

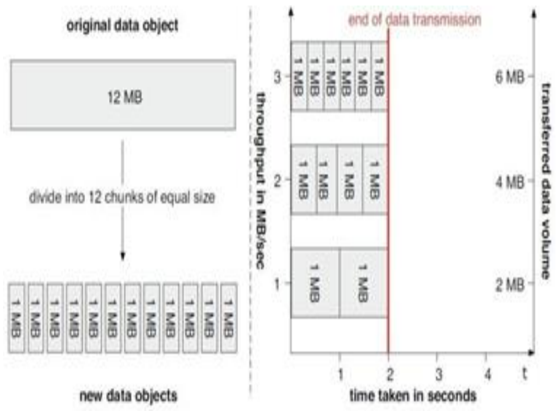

Fig 7 Optimization of Cloud-RAID[31].

This approach hampered the performance of the system by the slow providers. The optimization of this system was carried out by considering the maximum provider's throughput capacity. They assured that individual chunk transfer gets completely simultaneously. The approach used to achieve this was by distributing the load of transferring data across providers based on their capabilities as shown in fig 2.7[31]. The main aim of this approach is to strike a balance between pay per use feature of cloud and at the same time ensure the security of company's data. The goal is to achieve this by distributing the data on multiple Cloud Service Providers (CSP's). It is similar to service oriented version of RAID (Redundant Array of Inexpensive Disks). This approach manages redundancy across multiple hard drives. Like Raid 5 which stripes the data across an array of hard drives and maintains appropriate numbers of parity drivers to reconstruct the date in case of failures, cloud RAID makes use of erasure coding techniques. It makes use of AES encryption algorithm and Message Digest 5 (MD) and Secure Hash Algorithm 1 (SHA) as the cryptographic hashes. It facilitates the execution of encryption, decryption and provides access in parallel due to the model of one thread per providers per package.

F. NC cloud

NC cloud is built on top of Functional Minimum Storage Regenerating (FMSR) codes a network coding based storage scheme [32]. As in case of traditional erasure codes even this code maintains the same fault tolerance and data redundancy at a much lesser repair traffic. Due to this it incurs much lesser monetary cost in data transfer. The key feature of FMSR is that there is no need to encode the data storage nodes during repair. It is a proxy based Multi-Cloud storage system.

The interface between the cloud and the client applications is the proxy server as shown in fig2.8(a). In case of cloud failure, the proxy activates the repair action as shown in fig 2.8(b) [32]. Reconstruction of the data is done by the proxy, by reading the data from other surviving clouds. This recovered data is written to new cloud. There is absolutely no interaction between the other clouds.

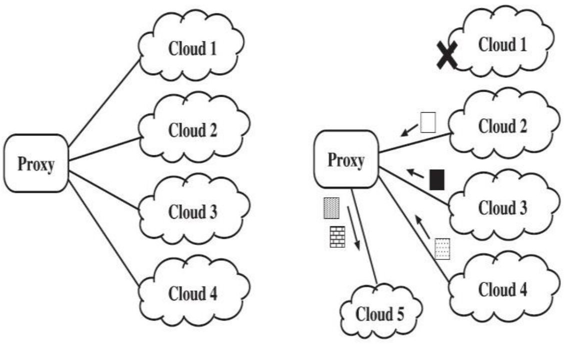

Fig 2.8 Proxy-based design for multiple-cloud storage(a) normal operation and (b) repair operation when cloud node 1 fails[32].

A file object that needs to be stored on the cloud is divided into chunks of equal size whose total size is equal to M. These are distributed on n nodes, which are greater than K. Hence the required data can be reconstructed using any K number of nodes. This property of handling failures is known as Maximum distance separable (MDS) property. Even the reconstruction of the data in the failed node can be achieved by downloading fewer amounts of data from the surviving nodes. There is no necessity to download the entire file for reconstruction. This is implemented using FMSR codes.

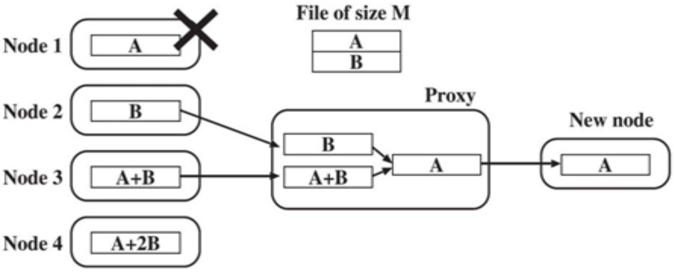

Fig 2.9 RAID 6 Codes[32]

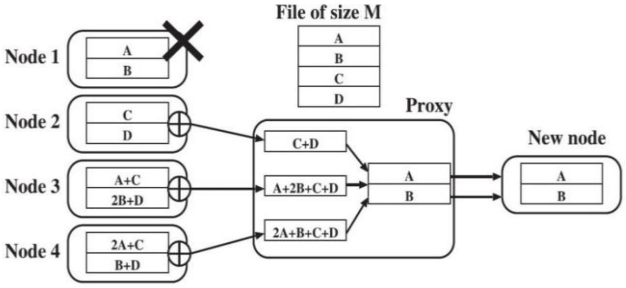

Fig 2.10 EMSR Codes[33].

In case of RAID -6 Exact Minimum–Storage Regenerating (EMSR) codes are used as shown in fig2.9. EMSR codes keep the same storage size. The revised version which divides the file into four chunks and stores the native and code chunks as shown in fig 2.10 [33]. The above fig shows how the data is reconstructed by using the other surviving nodes. When any one node fails to repair it, the surviving nodes send the XOR summation of the data chunks present in them. It is clearly seen that when the stored data size is 2M the repair traffic is only 0.75 M. Thus we can see that there is 25% saving.

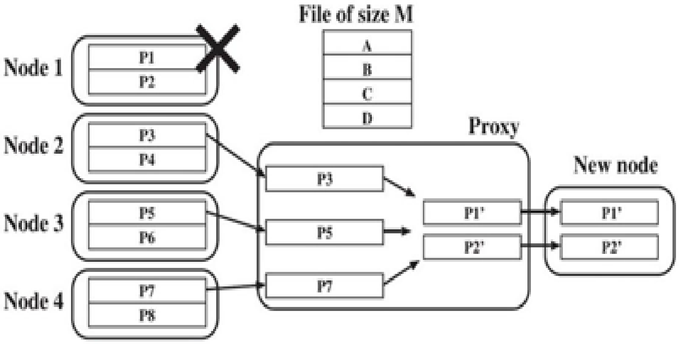

Fig 11 FMSR codes

Considering the double fault tolerant implementation of FMSR codes, as shown in fig 2.11, divide the files into 4 native chunks. These chunks are later constructed into 8 chunks namely P1…..P8 by using different combinations. The original four native chunks can be recovered from any two nodes each code chunk has the same size as M/4 equivalent to native chunk. Even though like EMSR the FMSR codes save 25% of repair traffic, the additional property of these codes is that nodes do not perform encoding during repair observation.

Though NC-Cloud uses FMSR codes, which have the same traditional erasure coding properties like Raid-6 with respect to fault tolerance and data redundancy yet this is economical.

Pustchi et. al. have focused on sharing and collaboration of the resources available on various clouds [34] and focused on the efficient use of resources like compute, storage and networking. The paper [35] in the year 2015 presents a framework Try ones for multi-cloud storage using erasure coding technique to formulate and optimize data placement. This work carries out both single objective optimization as well as multi objective optimization. Authors have proposed Euclidean distance measure through geometric space abstraction to optimize the results of the objective function. Later a model to utilize the free storage provided by various CSPs is proposed in [36]. Most of the CSPs have a provision of allowing their clients to make use of some amount of storage free of cost on their cloud. However this space is very less on individual CSP and hence the user may have to spread the application over large number of CSPs this implementation will need efficient data splitting and distribution on multiple cloud providers. Cyrus, client based architecture which allows individuals to distribute shared files to multiple CSPs [37]. This scheme ensures privacy by dividing the data among several CSPs, so that no single CSP can reconstruct the data. Part of the file is not stored on single CSP whereas it places on multiple CSPs to enhance reliability. It also optimizes the share download in order to minimize the data transmission time and promotes simultaneous upload to multiple CSPs. This may also promote competition among the CSPs as the users may have options to select the CSP before outsourcing the data. Managing Object storage is simpler as this storage architecture manages data as objects. Unlike the file systems which supports (Portable Operating System Interface) POSIX IO calls (open, close, read, write, seek), object storage supports only PUT and GET fundamental operations. By discarding the notion of files and directories, object storage achieves new heights of overall performance, substantially better scalability, resilience, and durability as compared to today’s most of the parallel file systems[1,38,39,40,41].

The simplicity of object storage semantics makes it extremely scalable, but it also extremely limits its utility due to following reasons:

- Object storage applications are limited to data archival because of Objects’ immutability, which restricts object storage to write-once, read-many workloads. This also implies, object storage cannot be used for scratch space or hot storage.
- As object ID and data, define the objects. Outside the object store, managing metadata for an object like logical file name, creation time, owner, access permissions is possible.

It is evident from the drawback mentioned that all the object stores need an additional data base layer called as proxy or gateway. This layer lies on top of the usual data base layer and provides a front-end interface to the users. It maintains metadata to provide the road map of the object ID. This metadata contains information of the object like access permission, owner information and many more.

The most traditional service type is shared file system, or called as “File Storage”, and the name itself says that multiple clients can access a single shared folder. The most popular shared file system protocols are Network file System (NFS) and Sever Message block (SMB) / Common Internet File System (CIFS) in Enterprise data storage. File storage is type of storage called as file- based storage or file-level storage. The data is stored in it in hierarchical structure. The saved data is in the file and folder and is presented to both systems as to storing data and retrieving data in the same way [38,4]. Even though Open Stack Cinder supports block storage format and open stack Manila supports file storage, it stores data on single cloud [42, 43]. The most suitable approach to achieve this would be File storage supported by Network Attached Storage (NAS) and erasure coding technique or replication.

Erasure coding technique distributes the data using redundancy where it is a forward error correction code which transforms a data of K blocks into a data of n blocks such that original blocks can be recovered from subset of n blocks. In replication K data blocks are replicated into another set of K blocks on a remote server. Thus replication requires at least double the capacity of original data. Data processing in redundancy is CPU intensive load whereas Disk intensive in case of replication. Thus the performance in case of replication is better compared to redundancy. Redundancy technique of distribution is better with respect to confidentiality of data as the data is divided into unrecognizable chunks.

III. CONCLUSION

High availability and integrity layer is distributed cryptosystem to protect integrity, rather than secrecy which aim is for more reactive than proactive. It worked by exploiting server redundancy and cross-server redundancy. HAIL provides integrity-protected error- correcting code (IP-ECC). RACS implements proxy server to distribute data among multiple CSPs. It used Apache ZooKeeper for distributed synchronization primitives of one-writer, many-reader for each (bucket, key) pair. It focused on economical failures and prevention of cost overhead. RACS fails to address the recovery of a repository from a transient outage and security of data. The DepSky Multi-Cloud system improve the availability and confidentiality of data stored on Multi-cloud storage providers system. It used combination of erasure codes, cryptographic secret sharing and Byzantine quorum system protocols enhances the security of the model. However, the cost of using this system will still be twice the cost of using a single cloud. This rise in cost is because of using the Byzantine fault-tolerant replication to store the data on multiple CSPs. Iris is implemented on distributed file system to support workloads from large enterprises storing data in the cloud and silent against potentially untrustworthy service providers. It used distributed caches to ensure the consistency of distributed data. The portal needs to be distributed internally in order to scale to the requirements of the enterprises having thousands of clients. It provides a mechanism for strong integrity, availability and recovery of corrupted data. However the data privacy issue is not being addressed. Cloud-RAID identifies a provider with maximum throughput and exploits its capability to complete the data transfer of all the chunks simultaneously. It improve reliability of data stored in clouds and confidentiality by using

Advanced Encryption Standard (AES) for symmetric encryption. The drawback of this system is that the CSP with better throughput keeps getting more loads. Economic aspects with respect to amount of data stored or transmitted not addressed. NC cloud FMSR code maintains the same fault tolerance and data redundancy at a much lesser repair traffic as in case of traditional erasure codes which incurs less monetary cost. It focused on reconstruction of failed CSP by economical way. However Security features like confidentiality, integrity and cross VM attacks require more research emphasis.

IV. REFERENCES

[1] C.Bandulet,Evalution of filesystem.[Online].Available: http://www.snia.org/sites/default /education/tutorials/ 2010 / spring/file/Christian _ Bandulet_SNIA Tutorial%20 Basics_Evolution FileSystems.pdf

[2] EMC.[Online].Available: http://www.emc.com/about/news/press/2011/20110628-01.htm

[3] EMC.[Online].Available: http://www.emc.com/about/news/press/2011/ 20110628-01.

[4] Siemon, Data Center Storage Evolution.[Online].Available:https://www.siemon.com/us/white _papers/14-07-29-data-center-storage- evolution.asp

[5] G. Ateniese et al., "Provable data possession at un-trusted stores," in Proceedings of ACM Conference on Computer and Communication Security, NY,USA, Oct 29 - Nov 02, 2007, pp.598-609.

[6] G. Ateniese, R. Pietro, L. Mancini, and G. Tsudik, "Scalable and efficient provble data possession," in Proceedings of Secure Comm 2008, Istanbul, Turkey, Sept 22-25, 2008, pp.9-14.

[7] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in Proceedings of the third symposium on Operating Systems Design and Implementation (OSDI '99)., CA,USA, Feb 22-25,1999, pp.173-186.

[8] Daniel Warneke and Odej Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud.," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 6, pp. 985 - 997, June2011.

[9] C. Wang and Y. Zhou, "A collaborative Monitoring Mechanism for making a Multitenant Platform Accountable," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud '10), Boston, MA, Jun 22-25, 2010, pp.18-18.

[10] Zhu Yan, Hu Hongxin, Ahn Gail-Joon, S Stephen, and Yau, "Efficient audit service outsourcing for data integrity in clouds," Systems and Software, vol. 85, no. 5, pp. 1083-1095., May 2012.

[11] M. Schnjakin and C. Meinel, "Evaluation of Cloud-RAID: A Secure and Reliable Storage Above the Clouds," in Proceedings of 22nd International Conference on Computer Communications and Networks (ICCCN), Nassau, Bahama, July 30 - Aug 2 , 2013, pp.1-9.

[12] M. Schnjakin and C. Meinel, "Scrutinizing the State of Cloud Storage with Cloud- RAID: A Secure and Reliable," in Proceedings of IEEE Sixth International Conference on Cloud Computing (CLOUD), Santa Clara, CA, Jun 28 - July 03, 2013, pp.309-318.

[13] A. Haeberlen, "A Case for the Accountable Cloud," in Proceedings of SOSP Workshop on Large Scale Distributed Systems and Middleware (LADIS), Big Sky,MT,USA, Oct 10-11, 2009, pp.1-6.

[14] Y. Singh, F. Kandah, andW. Zhang, "A secured cost effective multi-cloud (INFOCOM 2011), Shanghai, China, Apr 10-15, 2011, pp. 619 -624.

[15] F. Feldhaus, Brightalk.com. [Online]. Available: http://www.brightalk.com/webcast/9077/71455

[16] T. cowen. Brightalk.com. [Online]. Available:http:/www.brightalk.com/webcast/8067/61744

[17] G. Lewis, Role of Standards in cloud interoperability. [Online]. Available: https: //resources.sei.cmu.edu /asset_files /Technical Note /2012_004_001_28143.pdf

[18] M. Singhal et.al., "Collaboration in Multi-cloud Computing Environment: Framework and security Issues," IEEE Computer Society, vol. 46, no. 2, pp. 76- 84, Feb2013.

[19] R. Ko, and P. Jagadpramana, "Trust-Cloud: A Framework for Accountability and Trust in Cloud Computing," in Proceedings of IEEE World Congress on Services (SERVICES), Washington, DC, USA, July 4-9 , 2011, pp. 584- 588.

[20] H. Wada, J. Suzuki, and K. Oba, "Multi-objective Optimization of SLA-aware Service Composition," in Proceedings of IEEE Congress on Services, Honolulu, USA, Jul 6-10 ,2008, pp. 368 - 375.

[21] R. Buyya, C. Yeo, and S. Venugopal, "Market Oriented Cloud Computing: vision, hype, and reality for delivering it services as computing utilities.," in Proceedings of 10th IEEE International Conference on High Performance Computing and Communications, (HPCC '08), Dalian, China, Sept 25-27, 2008, pp.5-13.

[22] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599- 616, Jun2009.

[23] T. Restenpart, and E. Tromar, "Hey you got off my cloud : exploring information leakage in third party compute clouds," in Proceedings of 16th ACM conference oncomputer and communication security, ChicagoIL, USA, 2009, pp.21-26.

[24] Y. Singh, F. Kandah, andW. Zhang, "A secured cost effective multi-cloud (INFOCOM 2011),Shanghai, China, Apr 10-15, 2011, pp. 619 -624.

[25] A. Husssain, and P. Leionne, "Racs- A case study for cloud diversity," in Proceedings of the 1st ACM symposium on Cloud, (SoCC '10), Indianapolis, IN, June 10-11, 2010, pp.229-240.

[26] A Bessani, M Correia, B Quaresma, F Andre, and P Sousa, "DEPSKY: Dependable and secure storage in a cloud-of-clouds.," in Proceedings of Eurosys'11 Sixth EuroSys Conference, Salsburg, Austria, Apr 10-13, 2011, pp. 31-46.

[27] K. Bowers, A. Juels, and A. Opera, "HAIL: A high-availability and integrity layer for cloud storage," in Proceedings of 16th ACM conference on computer and communication security, Chicago IL, USA, Nov 9 - 13, 2009, pp.187-198.

[28] E. Stefnov, A. Dijk, A. Juels, and A. Opera, "Iris: a scalable cloud file system with efficient integrity checks," in Proceedings of 28th Annual Computer Security Applications Conference (ACSAS'12), Florida, USA, Dec 3-7, 2012, pp.229-238.

[29] M. Schnjakin, and C. Meinel, "Evaluation of Cloud-RAID: A Secure and Reliable Storage Above the Clouds,"in Proceedings of 22nd International Conference on Computer Communications and Networks (ICCCN), Nassau, Bahamas,July30- Aug 2, 2013, pp.1-9.

[30] C.Henry H. Chen, H. Yuchong, and P. Patrick, "NCCloud: A Network-Coding- Based, ,"IEEE Transaction son computers,vol.63,no.1,pp.31-46,Jan2014

[31] J. Luo, A. Dimakis, C. Huang, and D. Papailiopoulos, "Simple regenerating codes: Network coding for cloud storage," in Proceedings of IEEE INFOCOM,2012, Orlando, FL, Mar 25-30 , 2012, pp. 2801 -2805.

[32] N. Pustchi, R. Krishnan, and R. Sandhu, "Authorization Federation in IaaS Multi Cloud," in Proceedings of the 3rd International Workshop on Security in Cloud Computing (SCC '15), Singapore, Apr 14 -16, 2015, pp.63-71.

[33] M. Su, L. Zhang, and Y. Wu, "Systematic Data Placement Optimization in Multi- Cloud Storage for Complex

Requirements," in IEEE Transaction on Computers, vol. 64, no. xx, pp. 1-14, xxx2015.

[34] C. Theodore and L. Hyoteak, "Enhanced cloud data placement Strategy for Multiple cloud Storage Services on Mobile applications," in Information Science and Application.: Springer, 2016, pp.545-553.

[35] J. Chung, C. Joe-Wang, S. Ha, J. Hong, and M. Chiang, "CYRUS- Towards Client-Defined Cloud storage," in Proceedings ofthe Tenth European Conference on Computer (EuroSys '15), Bordeaux, France, Apr 21-24, pp. 1-16.

[36] Sandeep Dutta. How Storage Technologies are Evolving to Manage the data. [Online]. Available: http://www-03.ibm.com/systems/data/flash/in/resources/Sandeep_Datta_-_InformationWeek.pdf

[37] Performance in a Gluster System.[Online].Available: https://s3.amazonaws.com/aws001/guided_trek/ Performance_in_a_Gluster_Systemv6F.pdf

[38] Introduction to Gluster systems.[Online]. Available: http: // moo.nac.uci.edu/~hjm/fs/An_ Introduction _To_ Gluster_ArchitectureV7_11 0 708.pdf

[39] S. Weil, and S. Brandt, "Ceph: a scalable, high-erformance distributed file system," in Proceedings of the 7th symposium on Operating systems design and implementation, (OSDI '06), Seattle, WA, USA, November 6–8, 2006, pp. 307- 320.

[40] Communications Supply Corporation, SAN over 10G ip – Communications Supply Corporation. [Online] Available: http://www.gocsc.com/uploads/white_papers/B2F87B7BE1CD4979A90504AC2514DF8A.pdf

[41] T. Roblitz, "Towards Implementing Virtual Data Infrastructures – A case study with iRods", Computer Science, vol. 13, no. 3, pp.21-33, Sept 21-33, 2012.

[42] Openstack DefCore Committee.[Online]. Available: https:/ / wiki.openstack.org/wiki/Governance/DefCore Committee

[43] A. Rajasekar, R. Moore, C. Hou, C. Lee, R. Marciano, A. Torcy, M. Wan, W. Schroeder, S. Chen, S. Gilbert, L. Tooby, P. Zhu, eScience on Distributed Computing Infrastructure.: Morgan and Claypool Publishers.

D.V. Bhavsagar, Assistant Professor, Department of Computer Science, Seth Kesarimal Porwal College, Kamptee (Nagpur). Area of research is Cloud Computing.