ENHANCED TPA BASED DATA INTEGRITY MECHANISM FOR OBJECT STORAGE ARCHITECTURE ON INTEGRATEDIOT CLOUDSMART ENVIRONMENT

Mr. Sunil Raj Y*, Dr. Albert Rabara S & Mrs. Helen Parimala E

Assistant Professor, Department of Computer Science, St. Joseph's College (Autonomous), Trichy – 620002.
Associate Professor, Department of Computer Science, St. Joseph's College (Autonomous), Trichy – 620002.
Research Scholar, Department of Computer Science, St. Joseph's College (Autonomous), Trichy – 620002.

ysrsjccs@gmail.com, a_rabara@yahoo.com, helenandrew07@gmail.com

Abstract— Cloud storage has become a part and parcel of everyday life and is used largely at this IoT age of computing. Facts are dumbed over storage whose physical location is not bothered by the very users. Storage may hold sensitive information, which may be more private have to be kept secure. Cloud service providers ought to ensure various security concerns such as integrity, availability, privacy and confidentiality to users. In current scenario of computing cloud owners must not possess physical data storage. As TPA provides transparency and fairness in performing auditing while serving as a bridge between CSU and CSP. Granting quite a lot of plusses, it is an untrusted unit, also it can initiate attacks or indorse threats by not ignoring them. For complete lifelike security purpose, also CSUs to attain extreme benefits on cloud at least computational cost, TPA is required. The intent of the paper is to analyze the existing object storage architecture and propose object storage with enhanced auditing method for CSUs/CSPs for guarantee the integrity of TPA while minimizing possibility of malevolent activities, TPAs are verified.

Keywords— Cloud Storage, Object Storage, Secure Storage, Cloud Security, Third Party Audit

1. INTRODUCTION

In the recent age of computing, cloud computing [5] has also emerged as one of the major shifts. It is a strong and low-cost technology that provides as a service various services such as network, platform, and applications. This capacity would allow all organizations to develop system quickly and economically that can scale up or down based on environmental changes. Data is growing day by day, and cloud computing provides easy access to web-based data at low cost. Through offering networking, processing, data storage, archiving and other resources such as automation, enterprise applications, IOT and mobile services, cloud computing supports organization in many areas. Cloud computing provides customers with personal, government, hybrid and collaborative cloud services

Big cluster databases and infrastructure are being used as cloud computing continues to evolve. Cloud infrastructure is the best solution for big data to store large volumes and various data varieties. Cloud computing has a wide pool of resources, space and networking which offers an optimal way to meet the demands of big data. Cloud storage today is amorphous. It delivers demand-based storage across a network. In cloud, you need to pay for the consumption of only the amount of data. Next to traditional remote access protocols or virtual or physical server hosting, cloud storage offers block, file storage and object-based objects.

In recent times, cloud item stores are gaining tremendous popularity because they provide considerably more cost-effective processing and pay as you go to conventional on-site space alternatives. Commodity processing of software size coupled with economies of scale has rendered virtual object stores a feasible option for many providers of cloud infrastructure [8]. This cloud object stores ' price models push expanded penetration with conventional on-site processing stacks. Improved momentum is benefiting from hybrid cloud infrastructures to greater data object storage convergence. Although these options to storing cloud items may equal conventional on-premise processing in terms of bandwidth, latencies are significantly shortened. Average latencies for storing cloud artifacts are usually 3-5X lower than their equivalents for storing on premise block / file. Cloud storage services are now offering different edge caching options focused on geo-location as cost additions and sometimes as a main feature differentiator [7] to overcome these limitations. For these caching services, pricing models vary from storage services for vanilla cloud objects.

Cloud computing has a problem that data centers don't always have proper control over stored data [7]. Data-controlled service providers can perform any tasks such as copying, destroying and modifying. Cloud computing ensures that the virtual machines are controlled at certain levels. Because of this loss of information access, security issues are higher than the standard paradigm for cloud computing. The only authentication does not give complete control over the data stored, but it does bring more than plain data. Cloud computing's features are virtualization, so multi-tenancy also has different attack possibilities than in the standard type of cloud.

As storage being a necessary and an important component in cloud and IoT integrated environment. It is a necessary to analyze the types of storage system being used. This analysis may help upgrading the system or to improve the efficiency of the existing system.

The types of storage system available can be put together as in Fig 1.

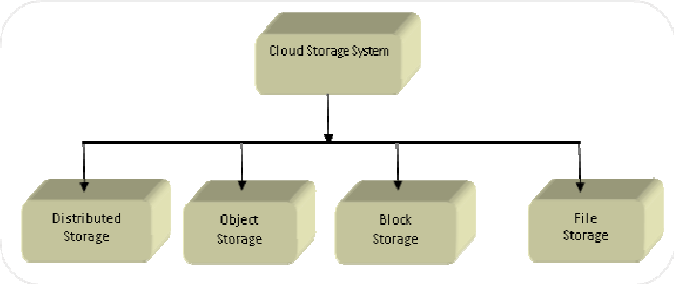

Fig. 1. Various Available Options of Cloud Storage System

i) Distributed System

Several ways to store and recover massive amounts of data have been suggested. In a cloud computing environment, some of these solutions were applied. However, issues hinder the practical implementation of such approaches, including the ability of current cloud technologies to provide the necessary capacity and high performance to address massive data volumes [11], how data can be stored in such a way that it can easily be retrieved and migrated between servers.

ii) Block Storage

Block storage [12] is an information storage where files are separated into small blocks and used in network (SAN) processing environments as in Fig. 2. Every block operates as a single hard drive and is installed by the storage manager and managed by internal database operating system. Although solutions for block processing appear to be more complicated and costly than other storage systems. Blocks are managed by the operating system and are typically used to access data from processing using protocols such as Fiber Channel (FC), Fiber Channel over Ethernet (FCoE) and iSCSI. Block storage is well equipped to processing software server and updating information regularly. For its persistent I / O operations, many applications preferably use block storage.

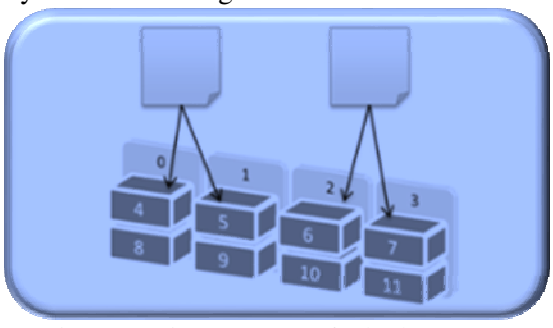

Fig. 2. Basic Structure of Block Storage

iii) File Storage

Fig. 3, shows the basic organization of file storage system often used in cloud to hold any type of data, including unstructured or semi-structured. Data File Level [13], the data disk is designed to a standard such as NFS and SMB / CIFS and the files are processed and read from it in bulk. Data at the directory level is simple to use and enforce. This contains files and folders, so transparency is the same for the accessing users and the processing process. Such space level is not costly to maintain as opposed to processing at block scale. Networked storage systems are largely dependent on the storage type of the file system. Storage at the file level can handle access control, integrate with corporate directories, etc.

In the file system, the locking function is used to provide read and write control. Meta data processing in the file system is handled independently and when the process grows it is very repetitive. Scale -out NAS has no limitations to scaling and does not decline as scale grows.

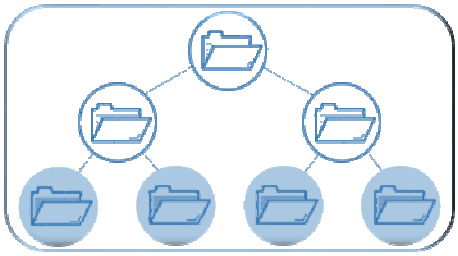

Fig. 3. Basic Organization of File Storage

iv) Object Based Storage

Object-based storage system [14] is a system that forms data as objects as blocks and files compared to other modern distributed storage system. The Fig. 4, describes the object storage as the data and memory are translated to Objects [5]. The entity is not a fixed size like blocks variable-length and can be used to hold all kinds of data, such as documents, server information, audio/ video, image, medical records. To store a complete file system or database, a single object could be used. The request of processing decides what is contained in an object. Unlike block I / O, a rich interface similar to a file system accomplishes the creation of objects on a storage device.

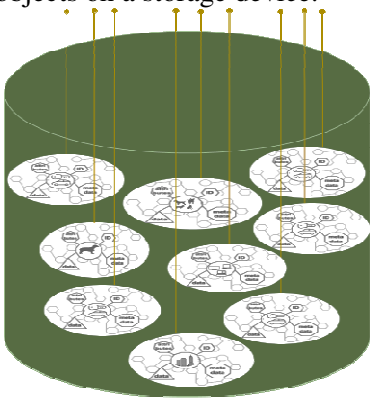

Fig. 4. Basic Structure - Object Storage

Object storage holds digital data, data from computers, sensor data, and directories, documentation, and very high scalability is easily managed. Using these metadata descriptors and policies to monitor information artifacts makes entry, creation, duplication, delivery, and persistence much more realistic than traditional approaches.

2 REVIEW OF LITERATURE

As the importance of storage mechanism is analyzed, this works presents an analysis on a major storage called object based storage. While using various parameters such as security, algorithms, protocols and other mechanisms used, the advantages and disadvantages can be very well understood by presenting an effective architecture with better data security.

2.1 Unstructured Data

The task of handling data provided by different devices during the present IoT age could be handled by object storage [1]. Because storage systems need to expand and Object Storage needs fewer metadata, they reduce overhead by storing them as objects to hold and access information. This could be continuously multiplied by incorporating nodes, and by replicating artifacts through multiple servers, consistency could be accomplished. The interface helps any client to quickly and easily access data.

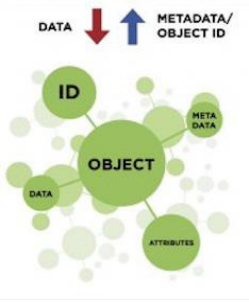

Fig. 5. Object Storage Architecture for Handling Unstructured Data

The predominant type of data contained is unstructured data [2], while innumerable programs take functions of files, images, and document management. Every implementation on the client will build a writable file. Such data can be relatively large, using more bandwidth and computing power to influence server requirements for high performance. To identify data objects, object storage would use unique identifier addresses.

This could provide almost infinite spaces for addresses. Because of its stateless architecture, commonly used cloud APIs are REST [10], SOAP [9] for accessing applications and widely used. To endorse both HTTP and HTTP(s) protocol, these make requests for service via the Web or CPS API.

Object Based Storage offers storage that is very robust and each object has its own hash to find duplicates of discard. Data protection is accomplished by duplication, erasure coding process in object base space.

This storage system uses Continuous Data Protection [15] which allows recoverability at the point of each write made to the disk by a process. Continuous data security, which contributes to restricted usage, is very cost-effective to use in storage system. A Continuous Data Protection framework will effectively combine cloud entity stores with caching to deliver low cost, high capacity, low latency, or high processing performance requirements. Through authentication, object versioning, and replication, data protection of object storage is fully achieved.

Duplication is a simple process for preserving data protection when making full, accessible, standalone access copies. For each element, three copies of duplication are placed in data space. Erasure coding can be useful with large quantities of data and any applications or systems that need to tolerate failures, such as disk array systems, data grids, distributed storage applications, object stores and archival storage.

2.2. Object Storage

The [2] system includes modules for clients, COP loggers that subscribe to COP service. The edge buffer cache serves as a staging area and from foreground applications masks network latencies. The system consists of destager, restore units, respectively, to store and recover information. A specific element of the design manager serves as a mapper between modifications of the file system and objects.

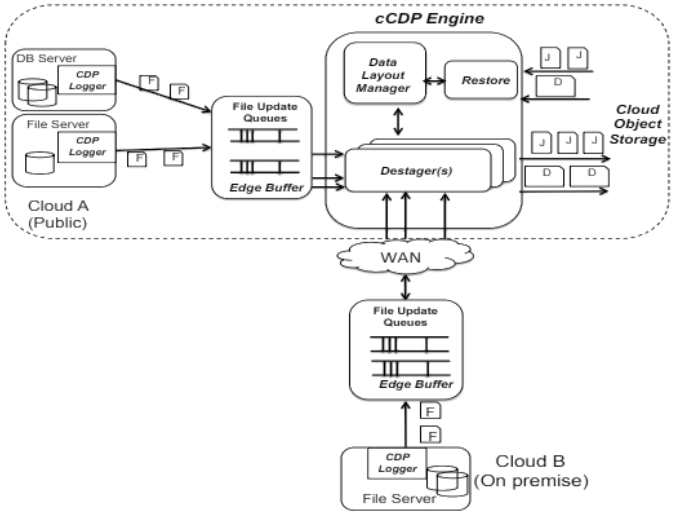

Fig. 6: COP System Overview

The operation of file system must be logged as COP logger module records operation timestamp for ordering and

consistency, dynamically selecting a queue at runtime based on a simple round robin load balancing strategy and returning an acknowledgement to the application. All of these queues are operated by a dedicated destage worker procedure that dequeues updates and executes the required packaging and writes packed object(s) to cloud store.

Model bundles metadata journal as well as data journal into individual objects, entry (J Ei) related to an atomic file system operation consisting of two types of metadata. Metadata describing file system operation, and internal layout metadata pointing to specific data object (D 0i) and data object offset corresponding to journal entry. Numerous journal records and associated information are combined / coalesced together and processed as larger journal items or data objects to avoid creating a large number of small artifacts or working in a more secure region of swift bandwidth. By coalescing operations, destager aims to maximize swift write efficiency.

Physical design will have a considerable impact on the performance of restoration / learning and can also affect the production of writing absorption. Naive format of two fast containers-one for all journal artifacts and one for all information items. From the viewpoint of reading / restore, locating data based on temporary constraints includes the following steps, listing all journal items in journal container, extracting data from each journal item, sorting journal entries based on specified time period.

This can take considerable time based on the number of journal items in the process. In addition, easily paginates these list requests to limit the impact of results (with standard 10k entries paging). A basic format requires a complete listing and reading of all journal items regardless of the length of time range to restore / read. Journal container could have billions of artifacts over standard lifespan of cdp and this simplistic format could severely affect recovery time and is the least realistic of the alternatives. One enhancement for this basic format is to encrypt the journal object's temporal metadata in its description. For example, encoding the lowest and highest timestamp of journal entries as the journal object's title can provide considerable pace. A pre-filtering stage based on the journal object title can also be included in the second method. From the point of view of reading, this basic format has the least write output overhead. To transfer to their respective bins, journal and information artifacts can be written at wire speed. However, this simplistic format does not make it easy to parallelize due to the lack of order semantics of list operations and the need for strict sorting.

2.3 Storage Architecture for Cloud Environments

Object storage have to with stand various issues such as storage issue, connection issue, policy issue to provide a virtuous infrastructure for cloud-based storage. As government agencies capable of obtaining information from a third party, cloud services may even be forced to submit data violating breaches of copyright and other relevant information [4]. All of these primarily affect customer data privacy, software is designed to solve these challenges.

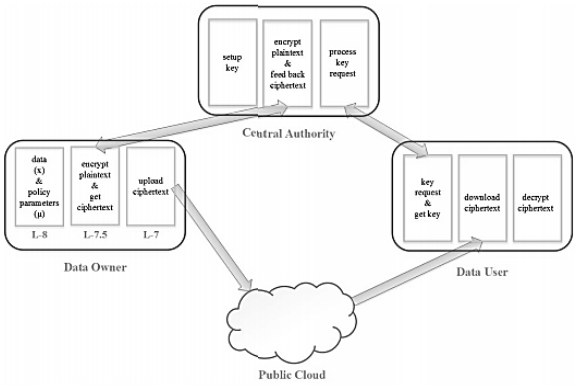

Fig. 7. Storage Architecture of Public Cloud

The software falls out with the following members including Software Manager, Data Consumer, Central Administrator, Public Cloud, among others. It would be appropriate for data users to perform information functions belonging to individual data holders. Although data owners can upload their data to the cloud, the policy parameters are attached [4]. A central authority must be fully trusted in which information holders may rely on it to guarantee that different policies are followed.

Information holder uploads data with certain policy criteria, it is transferred via central authority where plaintext is translated to cipher text and submitted to the public cloud. As the client wishes to process a cipher message, the central authority gets a "key-request," retrieving the rule variable attached to the document, the authority chooses whether or not to give a password. Client may encrypt cipher text downloaded from cloud only if key is released.

3 PROPOSED ARCHITECTURE

The main focus is on enhancing security to maintain customer data more private and highly unchanged. The major role players here is depicted as Users, Access Log, TPA and Owners. To ensure the reliability of storage during node failure, erasure code technique can be used, where duplication of objects kept in various other nodes. This could be done by breaking the objects in to equal partitions, based on size of data within. The user will be provided with the initial portion of the Object ID as they requests access.

3.1. System Model and Assumptions

According to the proposed architecture, the cloud being provided with the service, intended to go through two levels of authentication validation. Though the process seems to be a little larger, it could be understood the security provided to the storage shall be strong.

Cloud owners being the major contributors of entire services and making things more dynamic, given same kind of authentication and biometric key generation mechanism, providers can keep the storage in more secured way. Owners could only enter into the storage after key is generated based on the biometric data. Also based on the Registry customer will be provided with the object Id and associated access privileges. Now customers enter in to the storage and do the intended work and encrypt using the key that is provide by TPA.

A central authority called Third Part Auditors Management will be managing the whole process. Where TPA's may not be aware of actual data, since TPA would maintain the Data Dictionary and Users Privileges with respect to the object they request for access. Also this TPA would also continually monitor the Access Logs, which will be kept at the Service provider end. It is very clear that TPA's will no way have exact data in their hands instead only the data about the data.

TPA's are divided in to three different modules, which adds integrity to the data. A TPA management module for users is introduced, will take care of the users totally and provides key after consulting the TPA Activity Manager module. The TPA Owner Management module could control the customers regardless of the process they carry out with storage.

Both CSU, CSP is provided with mechanisms to audit TPA, thus verifying TPAs activities like performing any assigned auditing tasks with respect to an object id. The CSP also verifies whether the TPA performs its job within the assigned keys. For ensuring correctness, simulation was made with 5 instances of object storage, two cloud users and cloud owners.

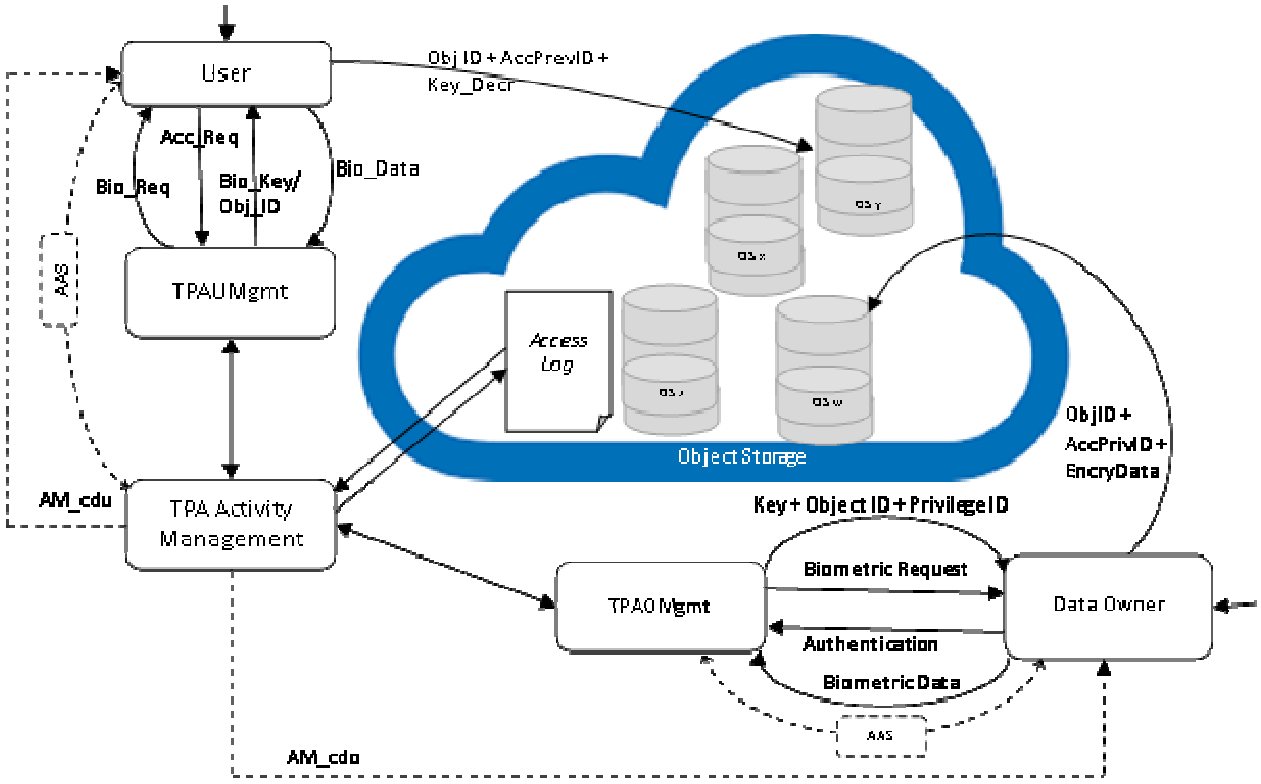

Fig 8. Secure Object Storage Architecture Based on TPA (SOSA-TPA)

Third party audit management module for both user and owners were added. The authentication mechanism used was found to be better and key generated is used to encrypt the data by the data owners. The key generated by TPA was used by cloud users to decrypt intended data, form the provider. As for the smaller set of input it was found that the system performed better.

Employing third-party cloud users for managing and verifying closure audit verification. Audit process is validated as,

$$val\_stat = \sum ((TPA_{log_{--}}). (DOW_{notice}).(TPAMgmt_{audit_{--}})) \text{ -------1}$$

The integrity of the data is confirmed, by using the above audit mechanism. The process followed is by retrieving the data from Third party auditor log while comparing it with data owner notice along with the third party audit manager. This is done before and after the access of every user on the cloud, data owner and the concerned third party.

4 RESULT AND ANALYSIS

Justification of any novel proposal is very much essential, therefore the integrity mechanism proposed is being validated by developing a prototype and performance of the setup is evaluated.

4.1 Proposed Architecture (SOSA-TPA)

The entire flow is evaluated using a dynamic model, with the help of wamp open source tool and visual studio code editor. The prototype ensures that the proposed architecture is functional, thereby ensures the integrity of storage. The user requests are monitored by user auditor, data owner is monitored by owner monitor and to enhance the assurance of data integrity activity management monitors the activities of both auditors. This could ensure the integrity of data by keeping the data safe and secure, from malicious intruders in the form of service providers or third party auditors.

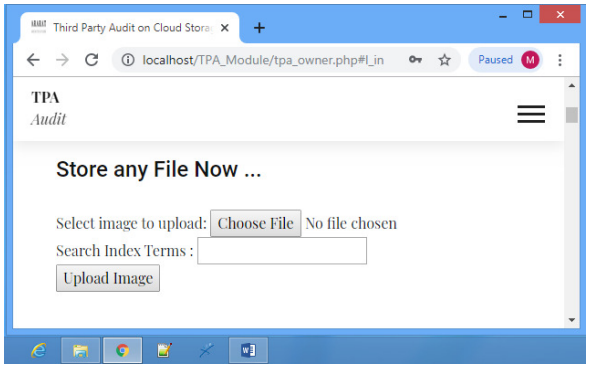

Fig. 9. Interface for Data Owner

As the registration module enable users to enter the system by providing details that are specific to the service, the data owners can enroll and can receive the generated id. As TPA owner auditor provides this, by taking first three alphabets of the name, along with associated random characters and finally a ten digit phone number of owner itself. As for the demonstration system is checked with storing images and documents, so as to ensure the integrity.

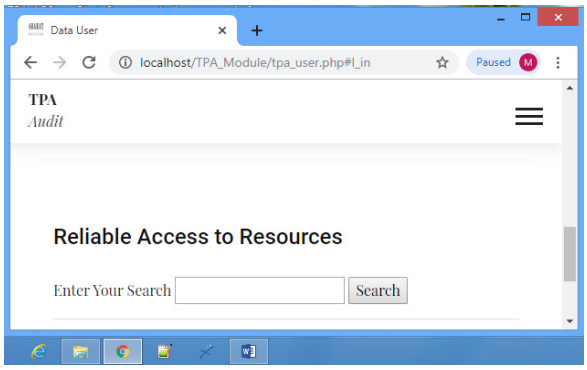

Fig. 10. Interface for Data Users

As the registration module allows users to enter the system by providing details that are specific to the service, the data

users can enroll and can receive the TPA generated id. As TPA owner auditor provides this, by taking first three alphabets of the name, along with associated random characters and finally a ten digit phone number of owner itself. As for the demonstration system is checked with storing images and documents, so as to ensure the integrity..

4.2 Performance Evaluation

To study the performance of the proposed architecture, the experiment is conducted using simulator. Proposed scheme was tested based on required auditing task involving CSP, CSU, and TPA Auditors including TPA Manager. The configurations made consists of 25 data centers, 6 user base and other parameters such as users and TPA's. Results observed were analyzed and found that the performance of proposed architecture is better with the specified configuration.

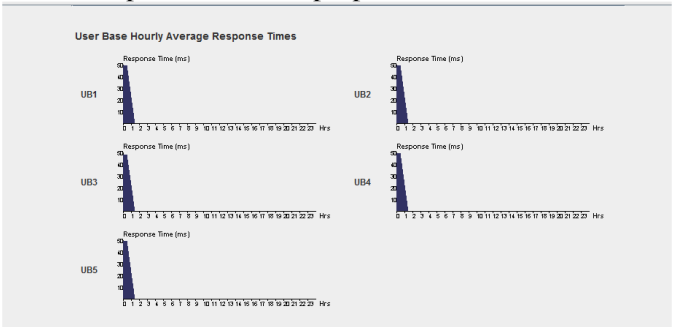

Fig. 11. User Base Hourly Average Response Time

Fig 11, depicts the average response time of user base in the frequency of hours. Six user base were generated and they are named as UB1, UB2, UB3, UB4, UB5 and UB6. The average response time of UB1 50.2ms, and that of UB2 is 49.9 ms, UB3 is 49.9ms, UB4 is 50.2 ms, and UB5 is 50.1ms. The overall average response time of user base is 50 milliseconds, the minimum response time is found to be 37ms and maximum response time is 67ms.

Fig. 12. User Nodes Generated on Simulation

The fig. 12, show the data centers used as samples for analyzing the results. For the evaluation twenty five data centers for DC1 to DC25 were generated, and the results were monitored. The result depicts the amount of requests processed in hours and it was noted as it was between 2000 requests per hour and 1000 requests per hour, which was based on the number of resources available and the processing capability of the resource.

	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	287.78	35.54	672.79
Data Center Processing Time:	0.41	0.01	1.11

Fig. 13. Overall Response Time

Fig.13, shows the overall response time of the system that is simulated. The results generated are based on the overall response time and data center processing time. The overall response time is noted as 287 ms as an average response time. And the data center processing time at a maximum of 1.11 ms and the average processing time is noted as 0.41ms.

5 CONCLUSION

This paper proposes an architecture of object storage, after analyzing existing works based on various parameters, specifically based on the security aspects. It was found that object storage could perform well and critical for effective infrastructure scaling. Object storage uses unique identifier addresses to identify data objects, could provide near-infinite address spaces. As CSU and CSP are auditing TPA back, also getting proper intimation for the activities done on the Object related to respective CSU or CSP. Erasure coding mechanism is used to provide security to the data. Data protection in object storage is highly achieved by using biometric based encryption along with third party audit. The prototype created performed well depicting the integrity of data and the performance analysis confirmed that the system could perform well in the cloud environment.

REFERENCES

[1] *Samundiswary.S, Nilma M Dongre, "Object Storage Architecture in Cloud for Unstructured Data", International Conference on Inventive Systems and Control, 2017*

[2] *NagapramodMandagere, RamaniRoutray, Yang Song, David Du, "Cloud object storage based Continuous Data Protection (cCDP)", IEEE, 978-1-4673-7891-8, 2015.*

[3] *RishikaKedia, AnishaLunawat, "Artificial Intelligence based Storage Management Architecture", IEEE International Conference on Cloud Computing in Emerging Markets, 2018.*

[4] *Chuan Fu, Jun Yang, Zheli Liu, ChunfuJia. "A Secure Architecture for Data Storage in the Cloud Environments", 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2015.*

[5] *SamhitaKanthavar, Manish Chawla, Saniya Simeen, Sarayu, "Design of an Architecture for Cloud Storage to Provide Infrastructure as a Service (IaaS)", IEEE , 978-1-5386-4318-1/17, 2017*

[6] *Ibrahim AbakerTarigoHashem "The Rise of Big data on cloud Computing: Review and open research issues", Elsevier, 2014.*

[7] *Naresh, Rao, "A Study on Data Storage Security Issues in Cloud Computing", 2nd International Conference on Intelligent Computing, Communication & Convergence, Procedia Computer Science 92, pp 128 – 135, 2016.*

[8] *Sharon Moses, Dinesh Babu "An Efficient user oriented Permanent deletion for Cloud Object Storage", IEEE, 2015.*

[9] *Julan YI, "Key Technology Research for Unstructured Data Cloud Storage: New exploring", 2nd International Workshop on Materials Engineering and Computer Sciences (IWMECS), 2015.*

[10] *Mike Mesnie, Gregory R Ganger, "Object Based Storage", IEEE, 2003.*

[11] *"Enabling Digital Transformation With object storage As Service", EMC, Aug 2016.*

[12] *Sandor, Mark, Peter, "Block Level storage Support for Open source IAAS clouds", IEEE 21st International conference on Parallel, Distributed and Network based processing, 2013.*

[13] *Neal Ekker, "File and object storage for dummies", A Willey Brand.*

[14] *MohdBazli Karim "Improving performance of database appliances on Distributed Object Storage", IEEE International Symposium on Cloud and Grid Computing, 2014.*

[15] *Yusuki, Seiya, "A High Performance, QOS Enabled, S3Based Object Store", International Conference on Cloud computing Research and innovation.*

[16] *Balusamy, Venkatakrishna, Vaidhyanathan, Ravikumar, Devi Munisamy, "Enhanced security framework for data integrity using third-party auditing in the cloud system", Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, Springer India, 2015, vol. 325, pp. 25–31.*

[17] *Sunil Raj, Albert Rabara, "An Integrated Architecture for IoT Based Data Storage In Secure Smart Monitoring Environment", International Journal Of Scientific & Technology Research, Vol 8 (10), 2019, pp.2213-2216.*

[18] *Sunil Raj Y,Lucase L, Helen Parimala, "DNA Based Hybrid Approach for Data Integrity on Integrated Cloud Based Smart Environment: An Analysis", IJSDR, 2020.*