International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Assessing the Performance of Raaes: Reliability-Assured and Accessibility-Enriched Storage Structure in Cloud Environment

Dr. S.Annal Ezhil Selvi[1],*,

[1] Associate Professor, PG. Department of Computer Science,
Bishop Heber College, Affiliated to Bharathidasan University
Trichirrapalli, TamilNadu, India - 620017.
Email: annalabel.cs@bhc.edu.in

Abstract—Cloud computing has revolutionized the world of distributed computing in recent decades, offering new dimensions and enticing opportunities. Presently, numerous organizations and individuals are increasingly transitioning to cloud storage services for their business and personal needs. This shift is primarily driven by the appealing concepts of "on-demand" and "Pay per Use," which provide flexible and cost-effective solutions. Perhaps, that's why it provides even more services to billions of users in every moment from the most popular companies such as Google, Microsoft, and Yahoo. However, the numerous cloud services are offered by a multitude of data centers equipped with power, network infrastructure, and backup systems. These services cater to users' demands for high availability and swift response times, necessitating the mirroring of each service across multiple geographically dispersed data centers. Previous research addressed this issue by focusing on the perspectives of vendors and users, aiming to develop economical and improved cloud storing solutions that fulfill stability and obtainability requirements for overall storage process. Furthermore, there is a suggestion to enhance the RAAES framework, aiming for efficient cloud storage by significantly reducing space and cost while still meeting reliability demands. Therefore, this study evaluated the performance of the RAAES through this paper, improved the availability requirements, ultimately reduced the response time, and actively promoted the development of the cloud with an efficient impact stock. As a result, it fosters the advancement of cloud technology by facilitating efficient storage, thereby generating a positive impact. [1]

Keyword*s*— Cloud, Storage, File Replication, Availability, Reliability, Replica Management.

I. Introduction

In recent years, cloud computing has garnered significant attention from users across various domains. It is a framework for distributed computing based on the concepts of "requested" and "pay-as-you-go". Within cloud computing, users share computing resources such as memory and data [1, 2, 3 and 4]. Service level agreements (SLAs) establish the connection between users and service providers, outlining key Quality of Service (QoS) metrics like scalability, storage, trustworthiness, consistency, stability, reliability, storage and cost [5, 6 and 7].

Cloud loading serves as a virtual repository for files and data, where digital records are logically stored. Hosting companies manage data across multiple data centers, ensuring redundancy and efficient storage management [8]. These cloud storage providers bear the responsibility of maintaining data availability and accessibility. Consequently, many individuals and organizations have transitioned to utilizing cloud storage solutions, finding satisfaction in their reliable and embedded nature.

Every day, massive internet companies like Yahoo, Google, and Facebook allure thousands of people with their appealing contributions. These services are hosted in massive data center's that include hundreds of servers, backup power sources, and solid network architecture. Each service is replicated over numerous far-reaching data center's to meet customer expectations of high availability and rapid response rates [5, 9]. File reproduction, copying of information, and external storage duplication are other terms for this type of service [5].

The storage replication service offers a highly efficient solution for ensuring valuable redundancy in the event of a failure in the primary storage system for backup data. This service enables cloud consumers to avail copied information promptly, minimizing interruption and related overheads [17]. By producing duplicate copies of the full backup file on regular intervals, proper implementation of the service can significantly enhance the disaster recovery process, resulting in a more streamlined approach [10].

To boost functionality and accessibility, several copies of the information file are stored in a data centre. Because the cloud is a streaming service, users have the option of paying for storage space in the cloud or use free online storage facilities. Cloud service providers (CSPs) that prioritize maximum demand are often favored by users, leading to the adoption of replication for increased availability [10, 13, 15, and 16]. However, it is important to weigh the benefits of replication against the associated costs.

For achieving cost-effective cloud storage, the number of copies should be considered by the consumers and the employed space they require, while also ensuring data accessibility and scalability over the cloud's lifecycle. This paper combines the features of DRRRA [7], DRCAES [6] a comprehensive approach for managing large volumes of cloud data files. The paper evaluates and provides detailed explanations of the RAAES system, utilizing cloud storage technologies for validation purposes.

1

The paper's structure is as follows: Section 2 examines related efforts and the existing framework, while Section 3 introduces the proposed RAAES system, including a detailed explanation of the cost estimation model with worked-out demonstrations. The RAAES algorithm, which is detailed in full, combines the distinctive characteristics of the DRRRA and DRCAES techniques. Evaluations of efficiency and validations are also included in this part of the manual. The results and evaluations of RAAES are reviewed in depth in Section 4, emphasizing the essential elements of the technique that is suggested. Finally, Section 5 summarizes the paper's results and conclusions.

II. Literature Review

The functionality of the contemporary Cloud Storage System (CSS) was evaluated with respect to the reviewer's perspective, as outlined in [7 and 9]. To address data redundancy issues, it is crucial for research to thoroughly examine the manner in which prevalent services are organized and the programmes employed for providing operations on different types of storage [1, 3, 9, and 10]. These applications typically operate as several hierarchical applications within software frameworks that operate globally. The popularity of files is computed based on the number of requests in multi-tier applications when a user submits a request, according to the current storage strategies in use [5]. However, the existing research on data replication in cloud servers has limitations, as it frequently relies on speculative explorations with no realistic constraints or experiential-based computations with no demonstrable guarantees of performance. The procedure of data reproduction and request-response on remote servers, which provides an optimization concern for effectively controlling stationary assets and consumer accessibility to cloud storage facilities, is an important part of replication [8]. Studies have proved that this optimization problem is NP-hard and involves delays, implying that no polynomial method currently provides an accurate resolution. Furthermore, the static approach has limitations as it cannot dynamically adapt to changing user access patterns [13 and 15].

Implementing centralized integer programming processes on cloud based servers poses different challenges. In a similar way distributed implementation of the approximated advancements in the perfect solutions for linear mathematical models is difficult [16 and 21]. Addressing this, a polynomial time-domain approximation algorithm is proposed in this work, specifically designed for a distributed environment like a cloud server, while maintaining uniform data sizes [9]. The replication selection and long-term optimization, driven by file access patterns, utilize the auction protocol for request/response and resource sharing. To solve replicating information for accessibility in the context of inconsistent procedures, a utility-based duplication technique on cloud-based servers is presented, which distinguishes it from the optimization task at hand [15 and 17].

In addressing the cost associated with replication creation and maintenance, previous research discussed performance enhancement techniques and cost optimization techniques [13 and 15]. However, the practice of arbitrarily accumulating identical objects disregards server heterogeneity, where different servers possess varying data request processing and network capabilities [18]. In search engine application production clusters, nearly half of the cross-rack traffic was generated due to the creation of replicas. While networks within a cluster often remain underutilized, imbalances in network utilization can lead to congested connections [9]. To tackle the challenge of multi-facility cloud resource allocation, solutions suitable for parallel implementation are primarily explored, driven by several reasons [18]. To begin, as mentioned in [19], the cloud allocating resources issue is fundamentally a sophisticated asymmetrical resource optimization problem with millions or more variables. In handling such massive amounts cloud computing challenges, centralized algorithms for cloud server distribution of resources are extremely ineffective [11, 17, and 21].

The CSS consists of two key components such as user interface and the server. Information replica plays a crucial role in ensuring data reliability and availability in these systems. Major cloud storage platforms such as As noted in the beginning, Google's Cloud File System, the case of Hadoop Microsoft Azure, and the Microsoft File System use multi-replica information replication technologies. By default, three copies (which incorporates the primary copy) of all data are kept in those networks. While a maximum of three clones for every element of data can be generated upon demand, the 3-replica mode is the most generally utilized strategy. As a result, this study refers to this particular data replica method as [9].

III. Materials and Methods

An improved framework named RAAES is proposed aiming for efficient cloud storage by significantly reducing space and cost while still meeting reliability demands. Therefore, this study evaluated the performance of the RAAES through this paper, improved the availability requirements, ultimately reduced the response time, and actively promoted the development of the cloud with an efficient impact stock.

A. RAEES Algorithm

The RAAES algorithm introduced in this study combines the key features of the algorithms such as DRRRA [7] and DRCAES [6] algorithms which is mentioned in the below algorithm and the presentation of Mathematical model for Calculation of Utilization Cost (CUC) and its algorithm. Through incorporating these characteristics, the proposed RAAES model includes a comprehensive cost calculation component.

RAAES Algorithm

Input : User activity
Output : Optimized Replica and optimized Cost.

```
Begin
    For every week from file uploaded time
        if Time_Th staisfied
            Call_DRRRA()  //defined in [7]
    For all user activity
            Call_DRCAES()  // defined in[6]
            Call_CUC()
    End
End
```

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

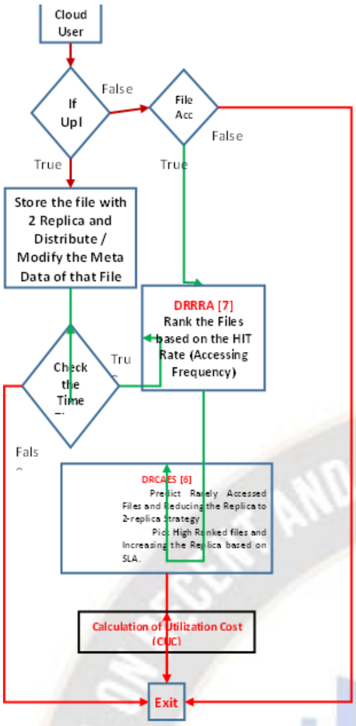

Figure 1. RAAES Flow diagram

The Figure 1 depicts the overall flow of the Proposed RAAES algorthm. It consists of three components. First one is DRRRA [7] algorithm, next DRAES [6] Algorithm, then Proposed CUC algorithm.

The DRRRA [7] algorithm processes files in multiple data centers based on user actions and generates a ranked result set. When a user interface triggers the process, it starts by setting some initial values then whenever the file accessed, the accessing value will be increased for computing the ranking of the file.

Next, The DRAES [6] algorithm is designed to dynamically increase file replicas in a distributed storage system based on user's need. The algorithm's ultimate goal is to distribute replicas of frequently accessed files efficiently across suitable data centers, ensuring both redundancy and resource optimization.

The proposed Calculation of Cost (CUC) Algorithm is presented in the next section.

B. Mathematical Model for calculation of Utilization Cost

The provided mathematical model involves various variables notations and equations used to assess a storage system's performance in the context of file uploads and user utilization costs. Here's a concise representation of the symbols as follows:

n: Number of uploaded files.

m: Number of Users.

Ui : i^{th} User, $i \leftarrow 1,\ldots,m$.

Fj : j^{th} file, $j \leftarrow 1,\ldots,n$.

FSi,j: File Size of i^{th} User in j^{th} File. It is stored in (m X n) matrix representation.

NoRi,j : Number of Replication of i^{th} User in j^{th} File. It is stored in (m X n) matrix representation.

OSi,j : Occupied Space for a File of i^{th} User in j^{th} File is calculated using the following eq.(1) and the result represented in m X m matrix.

$OS_{i,j} = (FS)_{i,j} * (NoR)_{i,j}$ where $i \leftarrow 1,\ldots,n \wedge j \leftarrow 1,\ldots,m$ (1)

OSi,j : User' Occupied Space (UOS) computed using the following eq. (2) as,

$OS_{i,j} = \sum_{j=1}^{m} OS_{i,j}$, where $i \leftarrow 1,\cdots,n \wedge j \leftarrow 1,\cdots,m$. (2)

UCdi,j: Utilization Cost per day (UCd) by the User is calculated using eq(3)

c: cost incur from the user per day for 1GB.

$UCd_{i,j} = OS_{i,j} * c \quad i \leftarrow 1,\ldots,n \wedge j \leftarrow 1,\ldots,n.$ (3)

OC : Opening Cost

BCi : Balance Cost in i^{th} day. Where $i \leftarrow 1,\ldots,30$ is calculated using the following eq.(1) as,

$BC_i = \{OC$ if $i=1$, $BC_i - UCd_{i,j}$ if $(1< i\ 30)$ $BC_i > 0$ (4)

C. CUC Algorithm

Input : F_i, U_i , $FS_{i,j}$, $OS_{i,j}$, $OS_{i,j}$, $=0$ $NoR_{i,j}$, Ucd_i , c=0.00067, OC=2, $BC_{i,j}$.
Output : UCd , BC_i

1. For each U_i do where i=1, …, n
2. For each F_j do where j= i
3. $OS_{i,j} = FS_{i,j} * NoR_{i,j}$
4. for each Ui do
5. for each Fi do
6. $OS_{i,j} = OS_{i,j} + OS_{i,j}$
7. for each Ui do
8. for each Fi do
9. $UCd_{i,j} = OS_{i,j} * c$
10. for i=1 to 30 do
11. if i=1 then
12. $BC_i = OC$
13. Else if (i<30 and i>1 and BC_i >0)
14. $BC_i = BC_i - UCd_i$
15. End

The algorithm takes a several inputs, including lists of Fi, Ui, FSi,j, NoRi,j, and BCi,j, as well as constants like c, OC, and NoRi,j. It aims to compute the values UCd and BCi.

It begins by iterating through each element in the Ui list (indexed by i) in a nested loop. Within this loop, it further iterates through the Fi list (indexed by j), where j is equal to i, creating a nested loop structure. Inside this nested loop, it calculates OSi,j by multiplying FSi,j and NoRi,j.

The algorithm then iterates through Ui and Fi again, incrementing OSi,j with itself in each iteration. Next, it calculates UCdi,j by multiplying OSi,j by the constant c.

The algorithm enters a loop that runs for 30 iterations (i varies from 1 to 30). If i is equal to 1, it sets BCi to the constant OC. For subsequent iterations where i is greater than 1 but less than 30 and BCi is still positive, BCi is decremented by UCdi.

Thus, the algorithm performs calculations on OSi,j, UCdi,j, and BCi, utilizing nested loops and conditional statements. It

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

repeats these calculations for a maximum of 30 days while adjusting the value of BCi. The final values of UCd and BCi is optimized depend on the input parameters.

D. Performance Evaluation of RAAES

The implementation and validation process utilized Java technology and the CloudMe cloud storage platform. Validation is performed on the local_host throughout this stage, whereas the documents are transmitted to the CloudMe storage service. Figure 2 depicts the process of execution flow in tremendous detail. When a user seeks access, the system validates their identity by checking the login information that they supplied. During the beginning or registration procedure, this authentication method is allocated. This approach is accessible to two categories of people: service providers (Admin) and service consumers (users).

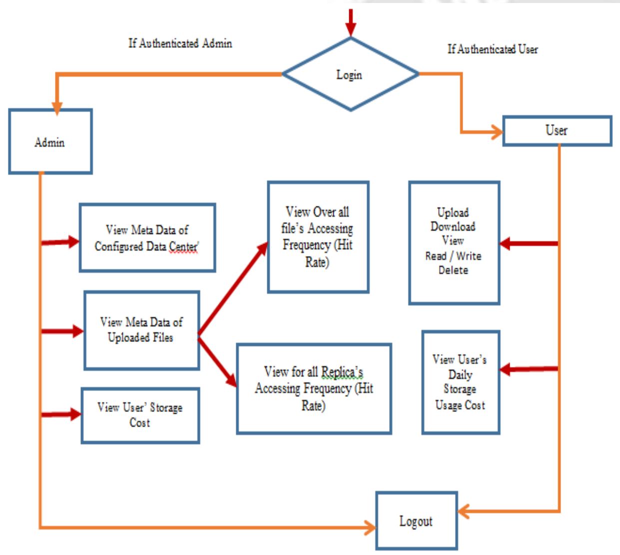

Figure 2. RAAES Implementation

The administrator possesses the capability to monitor various user activities and access all users' pricing models. They have visibility into the metadata associated with configured datacenters and uploaded files. This metadata includes information such as the availability of storage area of the datacenter, Available Space (AS), Occupied Space (OS), size and type of uploaded files, upload time, frequency of replicas, admission request frequency, and access type.

Additionally, the administrator can access details regarding all users' consumption costs, which include primary cost, outstanding cost and rationality cost [4].

IV. RESULTS AND INTERPRETATIONS

In Table 1, the proposed RAAES approach is assessed based on various parameters. This table provides an overview of the value assessments for four key parameters: NoR (Number of Replica), OS, Cost, and Request-Response Time Delay (RRTD). These assessments are compared with the existing PRC [8] for reference.

It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

TABLE I. COMPARISONS OF THE PROPOSED RAAES AND EXISTING PRC

S.NO	FAF	EXISTING (PRC) CONVENTIONAL 3 REPLICA STRATEGY (MINIMUM REPLICA IS 1 MAXIMUM REPLICA IS 3)				PROPOSED RAAES			
		NoR	OS= FS * NR	COST= OS * 0.00067	RR_TD	NoR	OS= FS * NR	COST= OS * 0.00067	RR_TD
1	<10	3	2.31	0.0015477	3	2	1.54	0.0010318	2
2	15-40	3	2.31	0.0015477	3.5	3	2.31	0.0015477	2.5
3	41-60	3	2.31	0.0015477	4.2	4	3.08	0.0020636	3
4	61-80	3	2.31	0.0015477	7	5	3.85	0.0025795	3.7

The modifications made to the NoR value will be directly manifested in the values of the OS (Operational Stability), Cost, and RR TD (Recovery Time Distribution) parameters. Currently, the NoR value in the system is established based on the disc fiasco degree scale of NoR at 3-replica, and it is considered as the standard approach.

The file with a File Size (FS) of 0.77 in GB, for example, is uploaded and retrieved in various contexts, as shown in the table above. The NoR is determined in the proposed system using the suggested DRRRA and DRCAES techniques.

The OS is determined in the following manner:

File Size (FS) * Number of Replicas = Occupied Space (OS) (NoR)

Here's how to figure out how much it'll cost:

Cost = OS * 0.00067 (0.00067 - 1 GB cost per day, which is used as an example).

TABLE II. COMPARISONS OF THE PROPOSED RAAES AND EXISTING PRC

Methods	Number of Replicas	Occupied Space	Cost	Reliability	Availability	Request-Response Time Delay
Existing PRC [Wen, 16]	1 or 2 or 3 Decide Based on Disk Failure Rate	Minimized Based on number of Replicas	Minimized Based on number of Replicas	No reliability with no replica 95% Assured with 2-replica 99% for 3-replica	Not Considered	Increased for more request
Proposed RAAES	Min replica as set; max determined by FAF, SLA.	Augmented	Augmented	95% Assured with 2-replica [Wen, 16]	Enriched	Reduced

The chart clearly demonstrates the effectiveness of the suggested RAAES in facilitating efficient data storage within the cloud environment. This storage system proves to be cost-effective and highly efficient, prioritizing data reliability and availability [1].

The figures presented in the chart are derived from different File Accessing Frequencies (FAF), and the corresponding values are graphically represented in the table.

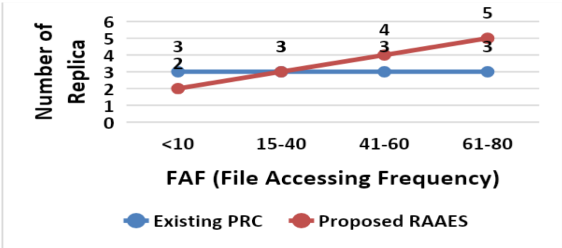

Figure 3. FAF vs. No. of Replicas (RAAES)

Figure 3 presents a comparison of changes in the characteristics of the Number of Replicas (NoR) across various FAF. The graph clearly illustrates that NoR is the standard in the current PRC. However, there are exceptions to this standard. The lowest number of replicas is two, while the highest number depends on the specific FAF and Service Level Agreement (SLA) requirements.

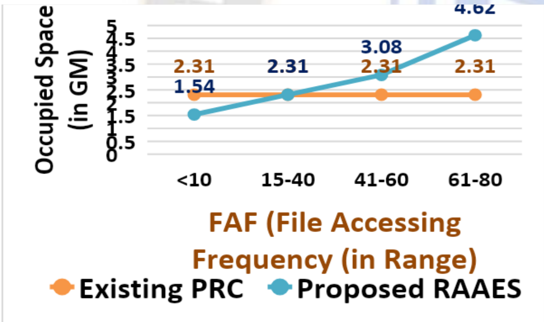

Figure 4. FAF vs. Ocupied Space (OS) (RAAES)

Figure 4 showcases a comparison of Occupied Space (OS) across various ranges of FAF. The graph provides valuable insights into the impact of changes in NoR on OS. However, it is important to note that the RAAES is optimized based on the specific FAF, ensuring efficient utilization of resources.

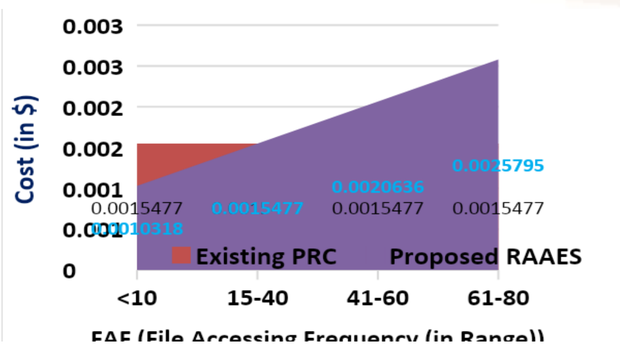

Figure 5. FAF vs. Cost (RAAES)

Figure 5 presents a comparison of costs in dollars ($) across different ranges of FAF. The graph highlights how the cost varies due to shifts in OS. The cost aligns with the existing PRC when the OS is standard. However, the RAAES approach optimizes the cost based on the specific FAF. For infrequently accessed files, the cost is minimized, while for frequently accessed files, it is maximized.

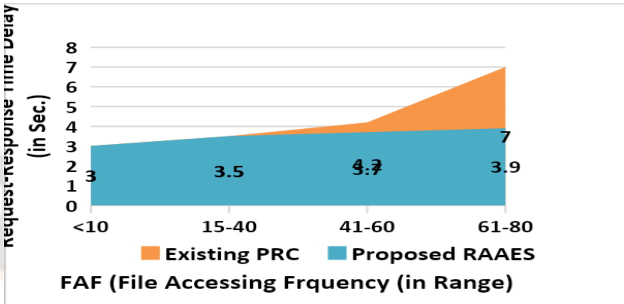

Figure 6. FAF vs. Request-Request Time Delay (RAAES)

Figure 6 illustrates the RRTD in seconds for various ranges of FAF. The graph depicts the values of RRTD on basis of chosen NoR, as determined by MATLAB tool results and reflections on the graph. In the existing PRC, the RR-TD is at its worst for the high FAF range due to the standard NoR. However, the proposed RAAES approach reduces the RR-TD by optimizing the NoR according to the FAF.

In this case, the reliability remains unchanged or unimproved. However, the cost is reduced by minimizing the frequency of replications while maintaining the same level of reliability concerns.

RAAES act as a Role-based Intelligent System, is a cutting-edge System that serves as an intelligent solution for cloud storage providers. This innovative system operates by assessing the usage patterns of uploaded files, primarily ranking them according to their Accessing Frequency, or Hit Rate. It offers several dynamic features that enhance storage efficiency and cost optimization.

One of RAAES's key functionalities is the ability to predict and identify rarely accessed files. These files are then subject to a reduction in the number of replicas if they meet certain criteria and time limits. Importantly, RAAES ensures that these operations adhere to set time constraints, safeguarding against service disruptions.

Furthermore, RAAES predicts the least accessed replicas within rarely accessed files and determines the data center with the minimum available space for removal. Conversely, for frequently accessed files, it dynamically creates new replicas and strategically places them in data centers with ample available space. This process also considers load balancing, considering various factors like available space, service level agreements, and accessing frequency.

RAAES's impact extends to the cloud storage market. It addresses the shortcomings of leading providers like Amazon, Microsoft Azure, and Google, offering optimized cost solutions and efficient storage management. By adopting RAAES, cloud storage providers can enhance their offerings, boost profitability, and meet user demands for reliable, cost-effective, and high-availability cloud storage solutions. Users, in turn, benefit from reduced costs, increased reliability, and decreased latency, aligning with their specific needs and service level agreements. In essence, RAAES is a game-changer in the cloud storage landscape, benefitting both providers and users alike.

V. CONCLUSION

The performance evaluation presents a comprehensive analysis of RAAES, a cutting-edge and cost-efficient mechanism designed to ensure data reliability and availability in the Cloud, especially when dealing with a vast number of data files. The proposed RAAES is thoroughly assessed by conducting a comparative analysis with the conventional 3-replica data storage approach commonly employed in the industry. The essential elements of the proposed RAAES are then listed, including the model's performance and cost-effectiveness. The cost-effective mechanism of data storage on Cloud is still in the testing phase, with techniques based on experimental environments. This endeavor will prominence on prompting a prototype of the solution in the Cloud soon. Then, when replicas of Cloud data should be generated or lost, there is a requirement for an efficient and effective data storage method that can retain the data replication level at a profitable range for the provider. The process of transferring data to appropriate storage devices, known as the data transfer activity, appears to be the most crucial aspect of the data generation and recovery stages in the lifecycle of Cloud data. Consequently, optimizing data transfer within Cloud networks could prove to be a financially advantageous strategy. By enhancing data transmission, the expenses associated with data creation or recovery can be substantially reduced. Future work on the following issues can be undertaken based on the existing work of this publication. These approaches will also tackle the challenges of time and space complexity, while simultaneously taking into account the importance of disaster recovery.

REFERENCES

[1] S Annal Ezhil Selvi, “Implementation of Raaes: Reliability-Assured and Availability-Enhanced Storage System”, JOURNAL OF ALGEBRAIC STATISTICS, vol. 13, no. 2, pp. 3711-3720, 2022.

[2] N. Mansouri, M. M. Javidi, and B. M. H. Zade, “Hierarchical data replication strategy to improve performance in cloud computing,” Front. Comput. Sci. vol. 15, pp. 152501, 2021.

[3] L. Chunlin, S. Mingyang, Z. Min, and L. Youlong “Effective replica management for improving reliability and availability in edge-cloud computing environment,” Journal of Parallel and Distributed Computing, Volume 143, pp. 107-128, 2020.

[4] A. Lazeb, R. Mokadem, and G. Belalem, “Towards a new data replication management in cloud systems,” *International Journal of Strategic Information Technology and Applications (IJSITA)*, vol. 10, no. 2, pp. 1-20, 2019.

[5] S. Annal Ezhil Selvi and R. Anbuselvi, “Popularity (Hit Rate) Based Replica Creation for Enhancing the Availability in Cloud Storage,” International Journal of Intelligent Engineering and systems (IJIES), ISSN: 2185-3118, vol. 11, no. 2, pp. 161-172, 2018.

[6] S. Annal Ezhil Selvi and Dr. R. Anbuselvi, “Optimizing the Storage Space and Cost with Reliability Assurance by Replica Reduction on Cloud Storage System,” International Journal of Advanced Research in Computer Science (IJARCS), ISSN: 2394-3785, vol. 8, no. 8, pp. 327-333, 2017.

[7] S. Annal Ezhil Selvi and Dr. R. Anbuselvi, “Ranking Algorithm Based on File’s Accessing Frequency for Cloud Storage System,” International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), ISSN: 0976-5697, vol. 4, no 9, pp. 29-33, 2017.

[8] N. Bonvin, T.G. Papaioannou, and K. Aberer, “A self-organized, fault-tolerant and scalable replication scheme for cloud storage,” In Proceedings of the 1st ACM symposium on Cloud computing, pp. 205-216, 2010.

[9] W. Li, Y. Yang, and D. Yuan, “A novel cost-effective dynamic data replication strategy for reliability in cloud data centres,” In *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing,* pp. 496-502, 2011.

[10] C. Cachin, B. Junker, and A. Sorniotti, “On limitations of using cloud storage for data replication,” In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops* (DSN 2012), pp. 1-6, 2012.

[11] D. Gupta, and D. Singh, “User preference based page ranking algorithm,” In *2016 International Conference on Computing, Communication and Automation (ICCCA),* pp. 166-171, 2016.

[12] N. K. Gill, S. Singh, and S. C. Satapathy, “Dynamic Cost-Aware Rereplication and Rebalancing Strategy in Cloud System,” © Springer International Publishing Switzerland 2015. In *Proc. of the 3rd Int. Conf. on Front. of Intell. Computer.(FICTA) 2014– Vol. 2, Advances in Intelligent Systems and Computing*, vol. 328, 2015.

[13] Z. Huang, J. Chen, Y. Lin, P. You, and Y. Peng, “Minimizing data redundancy for high reliable cloud storage systems,” *Computer Networks*, vol. 81, pp. 164-177, 2015.

[14] Y. Mansouri, A. N. Toosi, and R. Buyya, “Cost optimization for dynamic replication and migration of data in cloud data centers,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 705-718, 2017.

[15] M. A. Haque, H. Aydin, and D. Zhu, “On reliability management of energy-aware real-time systems through task replication,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 3, pp. 813-825, 2016.

[16] Wenhao Li, Y. Yang, and D. Yuan, “Ensuring cloud data reliability with minimum replication by proactive replica checking,” *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1494-1506, 2016.

[17] R. Li, Y. Hu, and P. P. Lee, “Enabling efficient and reliable transition from replication to erasure coding for clustered file systems,” *IEEE Transactions on parallel and distributed systems*, vol. 28, no. 9, pp. 2500-2513, 2017.

[18] H. Nishiyama, A. Takahashi, N. Kato, K. Nakahira, and T. Sugiyama, “Dynamic replication and forwarding control based on node surroundings in cooperative delay-tolerant networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 10, pp. 2711-2719, 2014.

[19] U. Tos, R. Mokadem, A. Hameurlain, T. Ayav, and S. Bora, “A performance and profit oriented data replication strategy for cloud systems,” In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld),* pp. 780-787. 2016.

[20] M. Hadji, “Scalable and cost-efficient algorithms for reliable and distributed cloud storage. In Cloud Computing and Services Science: 5th International Conference, CLOSER 2015, Lisbon, Portugal, May 20-22, 2015, Revised Selected Papers vol. 5, pp. 15-37, Springer International Publishing, 2016..

[21] S. Kianpisheh, M. Kargahi, and N. M. Charkari, “Resource availability prediction in distributed systems: An approach for modeling non-stationary transition probabilities,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, pp. 2357-2372, 2017.