A CONTROL SYSTEM FOR HYBRID MULTI-USER DATA REPLICATION AND ACCESS IN INFORMATION TECHNOLOGY FOR CLOUD DATA SECURITY

Panukorn Puripanyanan[1], Thaphat Chaichochok[2] and Tuannurisan Suriya[3]

[1]Department of Information Technology and Digital Innovation, Faculty of Science and Technology, Songkhla Rajabhat University, Thailand. E-mail: panukorn.pu@skru.ac.th

[2,3]Department of IoT and Information Technology, Faculty of Industrial Technology, Songkhla Rajabhat University, Thailand. E-mail: thaphat.ch@skru.ac.th[2], tuannurisan.su@skru.ac.th[3]

DOI Number: 10.14704/nq. 2022.20.6.NQ22702 **Neuro Quantology 2022; 20(6):14-23**

Abstract:

Due to the high computational cost and memory needs of real-time cloud computing, data replication-based cloud data access control measures are essential. In cloud-based applications, hybrid data replication techniques are crucial for data protection and recovery. Both the medical area and cloud computing applications require the use of machine learning tools and methodologies. Most typical machine learning models recover patterns from numerous virtual machines in a cloud computing environment using data partitioning and replication techniques that are static in nature. This study offers a hybrid data replication and multi-user data access method to enable robust data recovery and security in a cloud computing setting. For cloud data security, this work employs machine learning-based patterns for data partitioning and server security in the cloud. The experimental findings demonstrate that the hybrid data replication model outperforms the standard data replication methods in terms of data replication time and storage space in the cloud computing environment.

Keyword: data partitioning, cloud computing, data access, machine learning

1.Introduction

To maintain a balanced system load, the speed of data access should be enhanced. Scalability and availability are the two primary considerations for enhancing cloud performance. Replication generates several copies of an existing entity[1]. The generation of duplicates is an essential method for reaching this goal. Replication increases the availability of resources. Replication also enables minimal access fees, shared bandwidth use, and little time delays. In the event of a system breakdown, replication provides transparent, error-free access to resources. It is possible to extend replication over a computer network such that storage devices can be situated in physically distinct installations. In the case of data transmission failure, users retrieve neighboring copies to boost the throughput. Multiple sites are utilized to store the data, which has numerous advantages. In the event that a server fails to provide the necessary data, a system can continue to function utilizing duplicated data. This notion is still approachable. Data are stored in several locations. The required information is gathered from the request originator. This improves the performance of the system. The costs of creating, maintaining, and updating copies are necessary for the provision of replication advantages. Replication can considerably enhance performance [2]. It takes time for replication technology to recover data from other sites and restart services. There is a cost associated with performance. It is beneficial to accept the flaw and boost accessibility [3]. One issue with cloud computing systems is their susceptibility to malfunctions. In fact, the entire system's availability might be endangered if a single node fails [4]. Nonetheless, their dispersal improves the system's dependability. A tolerant fault system is a computer system setup that prevents the occurrence of an unforeseen problem. Fault tolerance is the ability of a system to continue to provide expected services despite faults that generate mistakes in the system's output. [5]. Its objective is to avoid malfunctions when they occur. A system that detects and corrects errors is fault tolerant. Permanent defects are identified and rectified while acceptable services are provided. It refers to all the approaches necessary for a system to tolerate any residual software errors after development. The state machine strategy is often referred to as active replication. The behavior of the copies is independent of one another. It is assured that all copies will arrive in the same sequence as the originals. Even in the event of a collision, this approach offers a rapid response time. The definition of a simple quorum is any significant node with Q>N=2, where N is the total number of copies (Q). Prior to the client's return, all actions should preferably be completed by a majority to guarantee that each customer always has access to the most recent system version. However, some systems enable you to optimize speed by configuring reading (R) and writing (W) quorums. To provide the same features as when using a majority, the quorum must intercept at specific nodes, such as W + R > N. Otherwise, quorums cannot overlap, thus it is possible to witness halted versions. First-tier backup replication: In the primary backup method, modifications to a single replica (the master) are implemented and transmitted to the remaining replicas. This dish may be made fast or gradually. With the impatient strategy, the client submits an update and the master reacts only after the update has propagated to all replicas. This procedure is exceptionally consistent. In the main backup strategy, regulating the competitiveness of updates is straightforward, and a single master might be considered a system bottleneck. It is anticipated that the master will react to the client when a local update has been done. When the client conducts a read operation on each replica, the stale data is exposed before to the distribution of the update. Chain Replication is a straightforward backup approach that prioritizes the creation of a high-performance data repository with simple access. This technique includes the placement of copies in a chain or series. Each node in a chain has a predecessor, with the exception of the initial (head) and terminal (dough) nodes. The fact that data owners no longer have direct access to their sensitive data, however, offers significant challenges to the data confidentiality and integrity requirements of cloud computing systems. Encrypting sensitive data prior to outsourcing the confidentiality problem to remote servers is possible.

Traditional cryptographic primitives based on hash and signature techniques cannot secure the integrity and accessibility of outsourced data without a local copy. The owners cannot download all of the stored data in order to verify its integrity, as doing so would demand expensive I/O operations and huge communication network overheads. In order to ensure the integrity of outsourced data that requires less transmission, calculation, and storage space, efficient approaches must be employed. Decisions include the processing or application of data and knowledge and are impacted by the qualities and conditions of decision-making as well as the relevant information/knowledge mix. In situations when decisions fall somewhere in the center, human decision-makers can be aided and enhanced by a decision-support system. [6]. In the event of replication or migration of common data blocks at random chip locations, a directory or broadcast method is utilized to look for and verify consistency, as the placement criteria in each block are likely to vary [7]. In many instances, a storage system's usability hinges on its scalability. When a large number of data items must be saved or the number of requests to store exceeds the capacity of autonomous systems, it is rational to distribute the data among numerous physical computers. The most sensible architectural option. If relatively few data items are utilized for several requests, replication should be performed [8]. It is ideal to replicate the inventory and customer data at these locations because it facilitates rapid access to local copies and disaster survival in the event that all physically situated equipment fails[9]. Initially, the Personalized Search team constructed a customer-side replication mechanism on the large table to guarantee that all clones would eventually be consistent. The present approach uses a replication mechanism that is embedded into servers[10]. They have devised replication algorithms for data access pattern[11]. Six distinct replication techniques with a multi-stage design have been developed. These solutions are primarily concerned with decreasing access latency and bandwidth utilization. Program replication options include "No Replica," "Best Client," "Cash Replication," "Fast Spread," and "Plain Caching." In their investigation, three types of access patterns were presented. The first kind is "temporary location," which indicates that the most recently requested file is being retrieved. Geographic location is the second method of entry. It indicates that the recently viewed file is likewise accessible to the nearby client in the same area. The fourth and third access pattern is "Spatial Location." It indicates that a file that was visited recently may be accessed again shortly. All six techniques were assessed using various approach patterns, and simulations revealed that cascade and fast spread performed the best. Replication of cascade worked well to minimize access latency, and rapid replication worked well when the primary goal is to reduce bandwidth usage, but it wastes a considerable quantity of storage [12]. The Dynamic Model Replication Strategy was proposed by [13]. Utilizing a peer-to-peer architecture, the replication choice is made decentralized in this technique. The primary objective of this method is to facilitate peer-to-peer data access. To increase availability, the peer can produce a replication automatically and decentralized. Due to the fact that each pair may independently determine the replication of the file, two pairs will be directed to generate replicas of the same file. Each peer is equipped with a set of tools to monitor file status and may opt to duplicate the available file level at a certain threshold. The advantage of this method is that there is no single point of failure, as the decision is made by an independent replication peer. It also results in the waste of storage space, as this method might make more copies than necessary. Other objectives include identifying the optimal host for new copies and determining the optimal number of replicas. Costs and benefits are the two characteristics used to choose the ideal client; costs and benefits should be considerably dissimilar. This technique is constrained by the fact that it assumes it is feasible to generate unlimited copies, but in reality, this is not achievable. Invoking the replica location service each time a replica is made is an additional cost [14]. In cloud computing, big systems are connected to public and private networks that deliver applications, data, and file storage with infrastructures that are dynamically expandable. Cloud computing offers enterprises with vital information from diverse storage resources without revealing their origin. Several challenges play a significant part in the administration of cloud applications in large cloud-based enterprises, including cloud backup and data retrieval, cloud interoperability & data analysis, cloud data loss concerns, cloud data integration, cloud energy consumption, cloud big data, etc. We primarily focus on data loss and cloud energy use. Statistical investigation of hardware failures in [15] indicates that a high number of cloud computing nodes makes hardware problems more probable than nontrivial. Some hardware faults can corrupt disk node data. Thus, programs that rely heavily on data cannot properly read data from CDs. The cloud computing system makes extensive use of data replication to prevent data corruption and guarantee high data availability. Data Replication duplicates and mimics replicas of a database in order to reflect changes made to a single replica. The replication enables a large number of users to work with a local copy of the database while updating it as if they were working with a centralized database. For replication applications when users are geographically dispersed, the most effective method of database access is frequently replication. Due to the heterogeneity of nodes in cloud computing, high-service-quality application data may now be played on a low-performance node. The low-performance node has a sluggish communications approach to latency and disk. If a node with good service quality experiences data corruption, application data is restored from a node with poor performance. Information about the QoS need and application is derived from the

eISSN 1303-5150

properties of the application. The advantages of the cloud replication approach include a quicker recovery time, the copying of offshore information in real time, the storage and replication of data without the use of external software, etc.

2.Related works

Replication based on the Bandwidth Hierarchy was a strategy that utilized network bandwidth and cloud measurements to replicate data. This replication approach utilizes network-level data access data. This strategy is primarily intended to reduce data access time by eliminating network congestion. The site is separated into many sections. Clearly, there is far greater network capacity inside a region than between areas. Thus, it is quicker to locate a file inside the same region than between regions. As in the same region, the number of routers between nodes is smaller and the network bandwidth is greater, but the number of routers and network bandwidth are bigger across regions. The BHR approach generates replicas in the region to reduce data access time and boost network locality. Region optimizer keeps track of the number of file visits and chooses whether or not to replicate the file based on the availability of storage. If a request is made for a file, it will be inspected at the requesting website first; if it is not located, the request will be collected from the other website, and the choice for replication will be taken after processing, taking local storage into account. If there is not enough space for the new file, the previous files will be deleted. First, the identical file cannot be saved at another location inside the same area. As a result, it is checked. The file will not be saved if an identical file already exists. If a file needs to be saved at the provided location, it must be deleted from the local storage of all other web pages. In the second stage, all files are saved in sorted order in the least-used storage element. After removing all files with a lower access latency than the duplicated file, it is possible to store new replicas. If adequate space is created, the replication procedure is executed. One process prevents data duplication, while the other replicates just the most popular files[17]. Branch Replication Scheme is described in [18]. replication method (BRS). The BRS architecture is hierarchical by nature. In the administration of grid data, the data replication system plays a significant role [19, 20]. Traditional data replication solutions demand vast amounts of storage space. BRS emphasises appropriate storage consumption. In contrast to normal replication data for the whole replica on each site, the BRS simply saves the subsection of the replica, and parallel access to that subsection is also allowed. This improves the efficiency of data access. [21]-[23]. As a result of data consistency difficulties, previous data replication strategies did not provide for efficient data modifications. The Branch Replication Scheme maintains consistency of replicas during updates. Additionally, this technique offers fault tolerance, enhanced speed, and scalability. In addition, the authors presented the system and the replica designation updating scheme. Simulation studies indicate that the BRS system outperforms competing systems, such as hierarchical replication systems and server-directed data replication schemes, in terms of data access performance and scalability. Branch Replication Scheme surpasses Hierarchical Replication Scheme in read and write operations for all file sizes. [24]. To add copies to hierarchical data, a mechanism for dynamic replication was presented. This approach, known as the PBRP strategy, is predicated on the file's popularity. The file's popularity is determined by keeping track of the customers' data access logs. It is anticipated that the popular files will be employed in the near future. The algorithm governs the dynamics of the access patterns at regular intervals. By strategically placing replicas, data accessibility and availability may be increased. By identifying the popularity threshold value, the PBRP algorithm has become effective. The primary objective of the method is to reduce job performance, correct bandwidth use, and optimize storage usage. The performance of this algorithm is superior to "best customer" and "caching" algorithms. It is located. Authors were also suggested the Adaptive-PBRP (APBRP) methodology. The pace of data request arrival may be used to determine the popularity threshold in this technique.

For load balancing reasons, the most frequently requested file is recognized by a larger weight value and reproduced at relevant places. The simulation findings are based on the average runtime, effective networking (ENU), and storage resource use compared to the least frequently used (LFU) and no replication. It illustrates that LALW's work operates concurrently with LFU optimizers in terms of network efficiency. [25] Dynamic Data Replication, also known as Upgraded Last Access Largest Weight in Data Grids, offered an enhanced LALW approach (ELALW). The position of the replica within the area is selected by ELALW. Storage is crucial while thinking about grids. Thus, replication and replica substitution are efficiently carried out. Similar to LALW's concept, the network architecture is separated into two layers and is based on centralized data replication management. The simulation findings indicate that ELALW has a shorter mean execution time than eight other dynamic replication schemes. The ELALW system is intended to minimize network traffic and bandwidth utilization. The low value for efficient network operation (ENU) suggested that the ELALW method effectively assigns replicas to cluster locations. Replication is conducted on a regular basis and replicas are kept in designated areas, hence decreasing undesired replication and lowering storage requirements. [26] investigates cost-performance tradeoffs between replicated and erasure-encoded storage systems[27]. We researched the process for data replication placement and developed a heuristic method for data replication placement. The simulation assesses if the algorithm demonstrates superior performance in a storage setting. Dynamic Provable Data Possession (DPDP), which is transparent, distributed, and replicated. It enables cloud storage providers (CSPs) to manage resources flexibly while concealing the underlying structure from clients, hence enabling them to provide customers proven services. The CSP has control over the amount and location of data storage. This work measures improved performance in simulation testing one to two orders, whereas replication systems enhance availability and dependability by distributing the load across various data centers. In addition, the optimal complexity of a centralized and distributed dynamic version control system is achieved by the use of permanent ranked authenticated skip lists.

3.Proposed Model

Data replication can occur when identical data is stored on numerous storage media, or when identical computer tasks are executed repeatedly. It is the process of distributing copies of data and database objects automatically and synchronously among SQL Server instances. Replication is the process of distributing data to enhance the dependability, fault tolerance, and accessibility of redundant resources, such as software or hardware components. If numerous storage media or data replication are utilized to replicate the same data, it is conceivable that the same computer task be executed many times. In this sort of setting, secure information sharing is a challenging issue. There are two primary types of replication protocols: active replication, in which all replica processes concurrently concur with all input messages, and passive replication. Access to and distribution of the data held by various sources is governed by policies that vary by source owner. The Database Research Community has focused on passive replication, in which just one replica processes all input messages and periodically sends their current state to the other copies for consistency. Data dispersion and replication give chances to enhance performance through the execution and loading of simultaneous queries and the expansion of data availability. In distributed database systems, data is routinely copied to increase dependability and accessibility, hence enhancing their reliability. According to the illustration, each user is originally allocated distinct virtual computers. Since each user has k virtual computers, the data partitioning technique is utilized to partition the data of each user. Each data portion in the data partitioning process is subjected to the integrity model prior to storage in the VM.

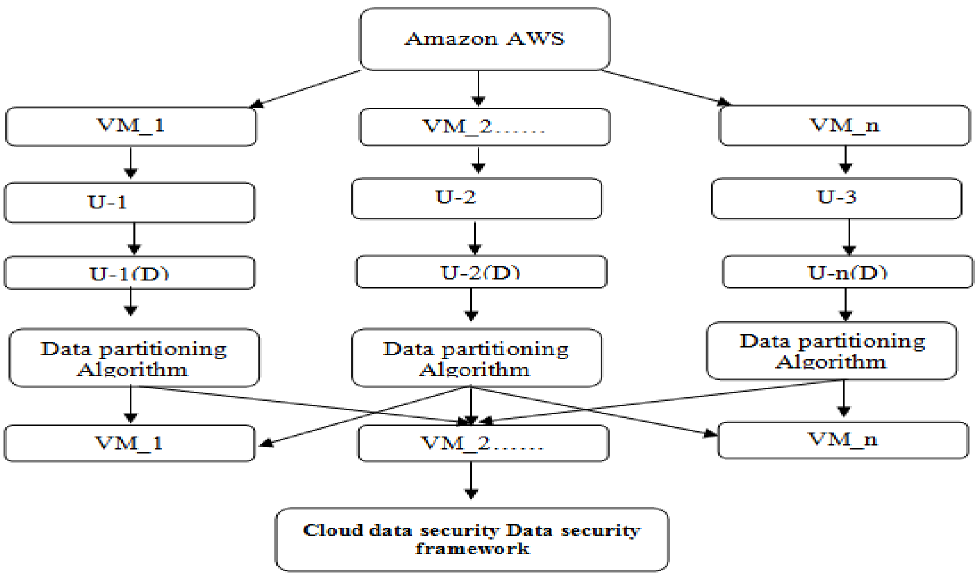

Figure 1: Proposed Multiple data partitioning load balancing parameter selection

- Algorithm for replicating data block by block:
- Input: cloud data files
- Data files with user access policies are the output.
- Procedure:
- Users' datafiles are appended to each cloud file.
- Create k chunks from the data.

1024 bits are assigned to each block B(i) in k blocks. Let V ID be the cloud virtual machine's identifier with accessible data zones. Calculate the user's access policy with algorithm 2 as U P(VM ID,B(i)). Calculate each user's secret nonce using the cyclic group parameters as the formula:

Let Zr, G1,G2 are randomized cyclic group parameter with Generator a

$GauDist(a) = k(1-k)^a$, a = 0,1,2, ...
$UniDist(a) = 1/(r1 - r2)$ for r1 $\leq a \leq r2$
CyclicElement p = bi-linearpair (Zr, $GauDist(a)$);
PrtKey.g = bi-linearpair (G1,$\sigma^2_{UniDist(a)}$);
PubKey.gp = bi-linearpair (G2,a);
MastKey.β = bi-linearpair (G2,a);
MastKey.g_alpha = bi-linearpair (PubKey.gp,$(p)^{Zr}$;
PubKey.h = bi-linearpair (PK.g,$(\beta)^{Zr}$);
PubKey.g_halpha = bi-linearpair (PubKey.g,MastKey.g_alpha);

Save each block in the η(VM_ID, B(i), U_P) by using the user's access policy.
Replicate the block to each VM in the VMList
done

This algorithm partitions each user's data file and replicates its corresponding block to different virtual machines for recovery purposes. In steps 1-3, each cloud user's data file is used as input and partitioned into 1024-bit blocks. In steps 4-7, each of the k blocks is used to compute the user's access policy and secret nonce using cyclic group metrics. Step 8 involves replicating each block to multiple virtual machines using a zone list. In this step, three parameters are used to replicate each block in multiple virtual machines: virtual machine ID, block data, and user access policy. Finally, in the step 9 each block is replicated in the available virtual machine.

Algorithm 2: User access policy generator

1: Initialize secret key K.
2: Partition the input data M into blocks with size 8.
3: while(len(M) >0)
Do
If(len(M)<8)
Pad message with sequence of …0000001;
else

Perform block processing using each block partition;

Done

4. Processing

Divide the block into 32 size sub-block for non linear transformation in the proposed model.

SP[] = BlocksPartition (SP/32)

5. Sub Block processing. For each byte in SP[i] DO.

$$\text{mat\_y} = |N| . \frac{e^{-|\sum K-\mu|/\rho}}{2\rho}; \rho > 0$$

$$\eta = \text{Norm(mat\_y)};$$

$$\text{gdf}(\eta) = \frac{\lambda^{\alpha} x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)}, \quad \text{for } x > 0$$

$$h1 = \text{sp(i)};$$

$$h2 = f(\text{sp[i]}) = \log\left(\frac{\lambda e^{-\lambda(\text{sp[i]}-\tau)}}{(1+e^{-\lambda(\text{sp[i]}-\tau)})^2} * \text{gdf}(\eta).\text{mean}\right)$$

$$h3 = \text{bytes(mat\_y)}$$

$$H[i] = h1 \wedge h2 \wedge h3$$

6 H= Concat (H0..H2..Hn.)

Using the policy generator, each user's access control policy is modified in algorithm 2. In this approach, a secret key and input data are partitioned as indicated in steps 1-3 to determine block-wise access control. In step 4, each block of data is sub-partitioned in preparation for step 5's computation of the access policy. In step 5, each sub-block partition's hash value is utilized to determine the user's access policy.

4.Experimental results

Experiment findings are achieved utilizing user datasets in a data cloud computing environment. This study utilizes the Amazon AWS cloud server to identify the block-wise replication mechanism in the accessible cloud virtual machines. In this experiment, each user's machine learning behaviours are utilized as replication input data. These patterns are extracted from medical datasets using classification algorithms based on filters.

Table 1: Performance analysis of proposed data replication model to the conventional models on cloud storage data(VM-1)

USER-1 DataSize	PDDRA	Dynamicweighted(DWDR)	linearDRA	Proposed
DataSize-1KB	3.96	4.42	3.33	2.3
DataSize-2KB	4.65	4.65	3.35	2.35
DataSize-3KB	4.32	4.54	3.18	2.39
DataSize-4KB	4.13	3.8	3.28	2.36
DataSize-5KB	4.73	4.4	3.22	2.33

eISSN 1303-5150

DataSize-6KB	4.33	4.32	3.38	2.34
DataSize-7KB	4.53	3.87	3.15	2.25
DataSize-8KB	4.2	4.44	3.13	2.31
DataSize-9KB	3.95	3.85	3.38	2.25
DataSize-10KB	3.83	4.52	3.32	2.36

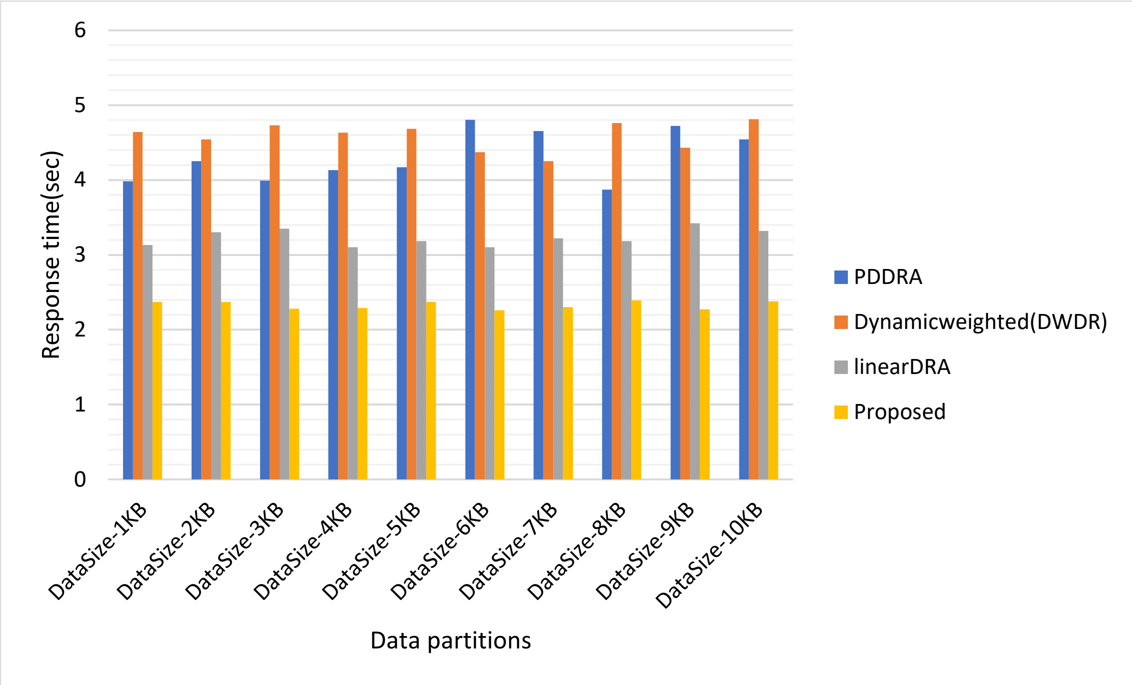

Figure 2:Performance analysis of proposed data replication model to the conventional models on cloud storage data(VM-2)

Table 2: Performance analysis of proposed data replication model to the conventional models on cloud storage data(Average of all VMs)

DataSize-1KB	4.23	4.26	3.29	2.39
DataSize-2KB	4.13	4.6	3.43	2.27
DataSize-3KB	4.52	4.55	3.18	2.36
DataSize-4KB	4.61	4.32	3.16	2.36
DataSize-5KB	4.12	4.73	3.17	2.31
DataSize-6KB	4.83	3.96	3.28	2.35
DataSize-7KB	4.34	4.26	3.15	2.34
DataSize-8KB	4.83	4.42	3.18	2.33
DataSize-9KB	4.22	3.98	3.15	2.4
DataSize-10KB	4.09	4.67	3.16	2.26

Table 3: Comparative examination of the storage space performance of the suggested data replication strategy vs the current approaches for cloud data storing (VM-1 and VM-2)

VM-1	PDDRA	Dynamic weighted(DWDR)	linearDRA	Proposed

DataSize				
DataSize-1KB	23.69	23.8	20.72	17.28
DataSize-2KB	27.36	29.96	27.06	18.66
DataSize-3KB	21.82	21.41	26.75	16.3
DataSize-4KB	27.3	25.85	27.18	16.77
DataSize-5KB	23.92	28.74	22.1	16.51
DataSize-6KB	25.6	29.15	20.23	18.45
DataSize-7KB	24.58	29.49	27.24	16.92
DataSize-8KB	23.63	27.26	20.91	16.22
DataSize-9KB	28.23	24.37	25.17	18.09
DataSize-10KB	20.05	24.4	22.92	18.58
VM-2				
DataSize-1KB	24.15	29.73	28.49	17.25
DataSize-2KB	26.66	27.36	20.55	17.22
DataSize-3KB	26.83	24.72	23.05	16.47
DataSize-4KB	27.04	20.17	27.05	18.17
DataSize-5KB	27.07	23.1	29.02	17.66
DataSize-6KB	24.57	24.58	21.78	16.46
DataSize-7KB	21.63	22.99	22.87	16.45
DataSize-8KB	28.86	20.3	21.07	18.61
DataSize-9KB	21.88	20.44	26.36	16.05
DataSize-10KB	22.28	22.88	29.9	18

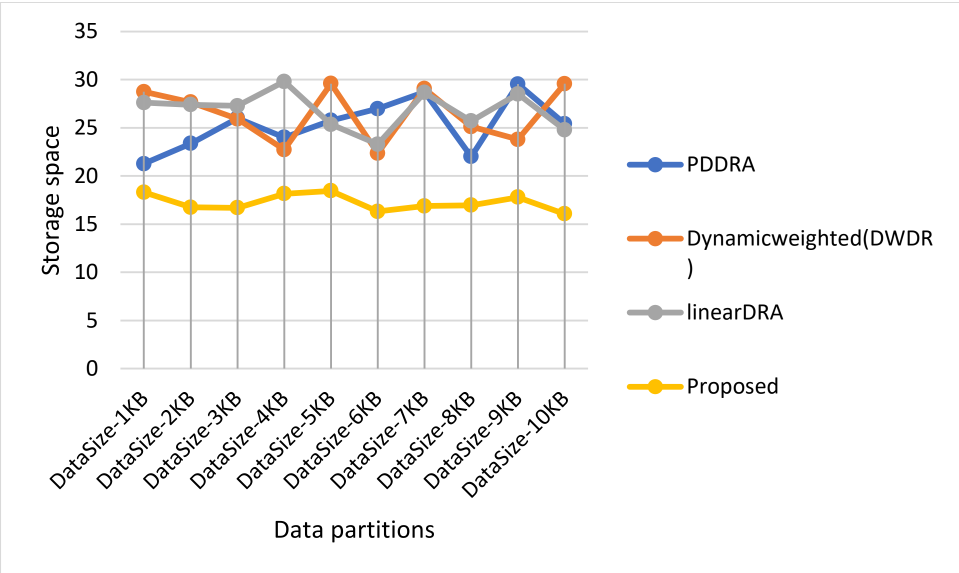

Table 3: Performance analysis of proposed data replication model average storage space to the conventional models on cloud storage data(VM-1 and VM-2)

5. Conclusion

This study designs and implements a hybrid data replication-based multi-user access control method for cloud servers. Due to high computational costs and memory limitations, it is impossible for the majority of traditional models to recover the user's machine learning habits. This study implements a hybrid data partitioning-based replication model based on the user's machine learning patterns for data recovery and decision-making.

References

[1]I. Casas, J. Taheri, R. Ranjan, L. Wang, and A. Y. Zomaya, "A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems," Future Generation Computer Systems, vol. 74, pp. 168–178, Sep. 2017, doi: 10.1016/j.future.2015.12.005.

[2]B. Alami Milani and N. Jafari Navimipour, "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions," Journal of Network and Computer Applications, vol. 64, pp. 229–238, Apr. 2016, doi: 10.1016/j.jnca.2016.02.005.

[3]R. Mokadem and A. Hameurlain, "A data replication strategy with tenant performance and provider economic profit guarantees in Cloud data centers," Journal of Systems and Software, vol. 159, p. 110447, Jan. 2020, doi: 10.1016/j.jss.2019.110447.

[4]S. Gopinath and E. Sherly, "A Dynamic Replica Factor Calculator for Weighted Dynamic Replication Management in Cloud Storage Systems," Procedia Computer Science, vol. 132, pp. 1771–1780, Jan. 2018, doi: 10.1016/j.procs.2018.05.152.

[5]N. K. Gill and S. Singh, "A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers," Future Generation Computer Systems, vol. 65, pp. 10–32, Dec. 2016, doi: 10.1016/j.future.2016.05.016.

[6]N. Mansouri, B. Mohammad Hasani Zade, and M. M. Javidi, "A multi-objective optimized replication using fuzzy based self-defense algorithm for cloud computing," Journal of Network and Computer Applications, vol. 171, p. 102811, Dec. 2020, doi: 10.1016/j.jnca.2020.102811.

[7]N. Mansouri and M. M. Javidi, "A new Prefetching-aware Data Replication to decrease access latency in cloud environment," Journal of Systems and Software, vol. 144, pp. 197–215, Oct. 2018, doi: 10.1016/j.jss.2018.05.027.

[8]P. Prajapati and P. Shah, "A Review on Secure Data Deduplication: Cloud Storage Security Issue," Journal of King Saud University - Computer and Information Sciences, Nov. 2020, doi: 10.1016/j.jksuci.2020.10.021.

[9]L. Helali and M. N. Omri, "A survey of data center consolidation in cloud computing systems," Computer Science Review, vol. 39, p. 100366, Feb. 2021, doi: 10.1016/j.cosrev.2021.100366.

[10]B. A. Milani and N. J. Navimipour, “A Systematic Literature Review of the Data Replication Techniques in the Cloud Environments,” Big Data Research, vol. 10, pp. 1–7, Dec. 2017, doi: 10.1016/j.bdr.2017.06.003.

[11]A. Briones, A. Mallorquí, A. Zaballos, and R. M. de Pozuelo, “Adaptive and aggressive transport protocol to provide QoS in cloud data exchange over Long Fat Networks,” Future Generation Computer Systems, vol. 115, pp. 34–44, Feb. 2021, doi: 10.1016/j.future.2020.08.043.

[12]Y. Mansouri, V. Prokhorenko, and M. A. Babar, “An automated implementation of hybrid cloud for performance evaluation of distributed databases,” Journal of Network and Computer Applications, vol. 167, p. 102740, Oct. 2020, doi: 10.1016/j.jnca.2020.102740.

[13]A. R. Setlur, S. J. Nirmala, H. S. Singh, and S. Khoriya, “An efficient fault tolerant workflow scheduling approach using replication heuristics and checkpointing in the cloud,” Journal of Parallel and Distributed Computing, vol. 136, pp. 14–28, Feb. 2020, doi: 10.1016/j.jpdc.2019.09.004.

[14]S. Skakun, E. F. Vermote, A. E. S. Artigas, W. H. Rountree, and J.-C. Roger, “An experimental sky-image-derived cloud validation dataset for Sentinel-2 and Landsat 8 satellites over NASA GSFC,” International Journal of Applied Earth Observation and Geoinformation, vol. 95, p. 102253, Mar. 2021, doi: 10.1016/j.jag.2020.102253.

[15]S. Arora and A. Bala, “An intelligent energy efficient storage system for cloud based big data applications,” Simulation Modelling Practice and Theory, vol. 108, p. 102260, Apr. 2021, doi: 10.1016/j.simpat.2020.102260.

[16]A. M. El-Shamy, N. A. El-Fishawy, G. Attiya, and M. A. A. Mohamed, “Anomaly Detection and Bottleneck Identification of The Distributed Application in Cloud Data Center using Software–Defined Networking,” Egyptian Informatics Journal, Jan. 2021, doi: 10.1016/j.eij.2021.01.001.

[17]J. Cha, S. K. Singh, T. W. Kim, and J. H. Park, “Blockchain-empowered cloud architecture based on secret sharing for smart city,” Journal of Information Security and Applications, vol. 57, p. 102686, Mar. 2021, doi: 10.1016/j.jisa.2020.102686.

[18]J. R. Gudeme, S. K. Pasupuleti, and R. Kandukuri, “Certificateless multi-replica public integrity auditing scheme for dynamic shared data in cloud storage,” Computers & Security, vol. 103, p. 102176, Apr. 2021, doi: 10.1016/j.cose.2020.102176.

[19]S. A. Bello et al., “Cloud computing in construction industry: Use cases, benefits and challenges,” Automation in Construction, vol. 122, p. 103441, Feb. 2021, doi: 10.1016/j.autcon.2020.103441.

[20]T. Mazzola, “Cloud dimensions from Jack Rabbit II field trial data,”

Atmospheric Environment, vol. 237, p. 117614, Sep. 2020, doi: 10.1016/j.atmosenv.2020.117614.

[21]R. Nachiappan, B. Javadi, R. N. Calheiros, and K. M. Matawie, “Cloud storage reliability for Big Data applications: A state of the art survey,” Journal of Network and Computer Applications, vol. 97, pp. 35–47, Nov. 2017, doi: 10.1016/j.jnca.2017.08.011.

[22]C. Li, M. Song, Q. Zhang, and Y. Luo, “Cluster load based content distribution and speculative execution for geographically distributed cloud environment,” Computer Networks, vol. 186, p. 107807, Feb. 2021, doi: 10.1016/j.comnet.2021.107807.

[23]G. Levitin, L. Xing, and Y. Dai, “Co-residence based data vulnerability vs. security in cloud computing system with random server assignment,” European Journal of Operational Research, vol. 267, no. 2, pp. 676–686, Jun. 2018, doi: 10.1016/j.ejor.2017.11.064.

[24]K. A. Torkura, M. I. H. Sukmana, F. Cheng, and C. Meinel, “Continuous auditing and threat detection in multi-cloud infrastructure,” Computers & Security, vol. 102, p. 102124, Mar. 2021, doi: 10.1016/j.cose.2020.102124.

[25]P. Yogendra Naidu, D. Rajalakshmi, S. Muthusundari, M. A. Berlin, and K. Manikandan, “Cost efficient tool for cloud storage services,” Materials Today: Proceedings, Jan. 2021, doi: 10.1016/j.matpr.2020.10.868.

[26]M. F. Mohamed, M. El-Gayyar, A. E.-R. Shabayek, and H. Nassar, “Data reduction in a cloud-based AMI framework with service-replication,” Computers & Electrical Engineering, vol. 69, pp. 212–223, Jul. 2018, doi: 10.1016/j.compeleceng.2018.02.042.

[27]S. Fu, L. He, X. Liao, and C. Huang, “Developing the Cloud-integrated data replication framework in decentralized online social networks,” Journal of Computer and System Sciences, vol. 82, no. 1, Part B, pp. 113–129, Feb. 2016, doi: 10.1016/j.jcss.2015.06.010.

[28]M. M. Alshammari, A. A. Alwan, A. Nordin, and A. Z. Abualkishik, “Disaster Recovery with Minimum Replica Plan for Reliability Checking in Multi-Cloud,” Procedia Computer Science, vol. 130, pp. 247–254, Jan. 2018, doi: 10.1016/j.procs.2018.04.036.