International Journal of Enhanced Research in Management & Computer Applications,
ISSN: 2319-7471, Volume 15 Issue 6, June 2026

To Strengthen the Cloud Computing Paradigm's Security Issues

Amit Gupta[1], Dr. Sanmati Jain[2]

[1]Research Scholar, Dept of Computer Science & Engineering, Vikrant University, Gwalior, Madhya Pradesh
[2]Supervisor, Dept of Computer Science & Engineering, Vikrant University, Gwalior, Madhya Pradesh

ABSTRACT

The best data storage plan in a cloud environment depends on more than just getting rid of redundant data. The effective storage and retrieval of data in relation to time and space also determines optimality. In addition to providing quick access to data hosted in the globally dispersed cloud data center, the optimal placement method also handles load balancing. Consequently, a unique optimal data placement technique is provided in the thesis' second phase to store the non-redundant data that results from the deduplication process. The ideal site is found using Floyd's algorithm and the Modified Distribution (MODI) method in this suggested model, which drastically lowers the time and cost of data transmission. In order to determine the best site, this model takes into account two primary factors: load balance and capacity limits. Floyd's technique is then used to determine the shortest path to the ideal location. Thus, the experiment results yielded a score that demonstrates how much the suggested approach reduces the time and cost of data transfer. Additionally, the suggested model aims to improve security while maintaining data integrity when obtaining data from the cloud environment. Thus, Merkle Array Hash Tree (MAHT), which is presented in the third phase of the proposed study, provides a secure way to maintain location information of data stored in geographically dispersed cloud data centers. The data found information must be retrieved from the MAHT in response to the user's request for data retrieval. Validation is required before the user's query, which initiates the data retrieval process, reaches the cloud database. Therefore, in the suggested method, the daemon not only verifies the query but also identifies and stops the SQL injection attack that uses a user query. SQL injection attacks are identified through the use of machine learning techniques, and the suggested scheme's enabled ANTLR parser stops them.

Keyword*: Model, Query, Transmission, Machine Learning Techniques*

1. INTRODUCTION

Burden adjusting administrations, which permit cloud assets or burden to be expanded or diminished dependent on request without respect for versatility, are regularly utilized by cloud suppliers. Space, CPU, network data transfer capacity cap, memory, and applications are instances of assets. Flexibility is one of the prerequisites for improving Cloud Computing tasks and utilizing assets in an adaptable manner. Virtual Machine asset load adjusting is an unquestionable requirement.

Inferable from the unpredictable plan of the cloud world, this is a troublesome work. The booking and conveyance of assets among cloud workers, which depends on the client solicitation and asset accessibility, are utilized to assign assets. Booking calculations are utilized to guarantee appropriate asset assignment, limit the quantity of Virtual Machine movements, lessen asset holding up occasions, and guarantee that assets are dispersed equitably through workers or Datacenters.

Booking is an assortment of systems utilized by a scheduler to guarantee that assets are distributed effectively. Both the cloud specialist co-op and the cloud client profit by ideal cloud asset booking. As far as asset use, the suppliers advantage.

Subjective measurements like Elasticity, Throughput, Fault Tolerant, Overhead, Resource Utilization, Response Time, Migration Time, and execution in a norm and versatile cloud climate ought to be focused on for better asset designation. Existing calculations don't zero in on flexibility measurements with regards to asset allotment. Relative examinations of current asset distribution calculations The Token Routing Algorithm (TRA) centers around token development in cloud frameworks; however it needs adequate data to disperse responsibility in a versatile cloud climate. This calculation gives a quicker and more exact steering choice. Cooperative effort (RR) is a straightforward

asset distribution strategy that uses the time cut rationale to oversee cloud clients' solicitations dependent on a particular time span or cut. Choked Load Balancer Algorithm (TLBA) doles out a solitary Virtual Machine to decided cloudlets, which contain work or errand relying upon the climate. Moreover, it utilizes a queueing framework when there are an enormous number of solicitations. occur simultaneously. Utilizing populace, hybrid, and transformation strategies, the Genetic Algorithm (GA) is utilized to limit Virtual Machine booking time and increment reaction time, yet relocation time turns into an issue. The strategy is dealt with by the Randomized Algorithm (RA), which is subject to the likelihood esteem. At the point when the assets are equally appropriated and there is no deterministic strategy, this calculation functions admirably. CQA exploits the standards of dynamic responsibility appropriation, line regulator, requester, and burden director.

By utilizing static and dynamic measurements in a cloud setting, the Gang Scheduling Algorithm (GSA) is answerable for choosing the best alluring apparatuses for Virtual Machine (VM) or cloudlet execution for planning equal positions. The Priority Based Scheduling Algorithm (PBSA) allots Virtual Machines dependent on responsibility weight, cloud client need, asset utilization, relocation, and execution time, with need changing powerfully to Virtual Machines utilizing maturing strategies.

Flexible Resource Scheduling is the standard and focal part for proficiently overseeing cloud assets dependent on client requests and cloud specialist co-op accessibility. VMs or occupations are doled out cloud administrations dependent on Service Level Agreements, which improves the Quality of Service. None of the current booking and burden adjusting calculations center around flexibility measurements to improve asset effectiveness, decrease Virtual Machine movement check, lower working expenses, and lower energy use for cloud specialist organizations and clients.

A multi-specialist framework (MAS) is a framework that comprises of different associating canny specialists inside a given setting. MAS can be utilized to take care of issues that an individual specialist or solid design would discover troublesome or difficult to tackle. Any methodic, viable, procedural, or algorithmic methodology can be viewed as knowledge. In MAS, specialists have the decision of collaborating, coordinating different specialists, and relocating from one machine to another.

Versatile specialists are generally utilized in different fields like circulated processing, lattice registering, man-made brainpower, remote sensor organizations, and semantic web because of the highlights and applications recorded previously. Joining versatile specialists builds the exhibition of the framework in an assortment of fields and concerns. The creators joined versatile specialists with distributed computing since conveying portable specialists diminished organization traffic while likewise lessening reaction time. In examination, in a regular cloud climate, each solicitation is sent straightforwardly to the WWW, bringing about expanded organization traffic. To address this, an amalgamated imitation of the versatile specialist was fostered that could be moved from one finish to the next.

2. REVIEW OF LITERATURE

Wen Xia et al. (2019) introduced a data deduplication technique in large-scale storage systems. This technique efficiently identifies and eliminates redundant data chunks and files. As per the author's statement, the utilization of secure-hash-based fingerprinting in the deduplication technique increases the time and computational requirements of storage systems. To prevent duplication, this work employs Content-Defined Chunking (CDC). This paper outlines the deduplication process, which consists of four distinct stages: fingerprinting, indexing, writing, and pipelined and parallelized data deduplication. To minimize the computational expense, these four stages are ultimately organized into a pipeline to enable parallel processing.

In this study, Fu et al. (2019) implement a deduplication strategy in cloud data centers to enhance storage resource efficiency. The author proposes a cloud-based framework for distributed deduplication that incorporates application awareness, data similarity, and locality. This framework aims to improve deduplication efficiency by implementing inter-node and intra-node application-aware deduplication. The proposed method utilizes large-scale data segmentation with a stateful data routing technique based on manual printing to ensure efficient global deduplication while evenly spreading the workload among nodes. This design optimizes RAM usage, achieves efficient load balancing, minimizes communication overhead, and lowers data redundancy across nodes.

Wen et al. (2015) introduced a Deduplication-Aware Resemblance detection and Elimination technique with little resource usage. This proposed solution in data deduplication-based archival storage systems utilizes pre-existing duplicate-adjacency information to achieve highly efficient similarity identification. In order for this strategy to operate, it is necessary for two data chunks to be comparable. The suggested approach enhances the efficiency of super-feature-based similarity recognition when neighboring data chunks in a deduplication system are identical.

Yan et al. (2016) provides a service provisioning by rearranging the diverse resources in the cloud environment. The author presents a concise elucidation on data storage and essential services provided by cloud computing. To ensure data security and privacy, data stored in the cloud is encrypted, which presents challenges related to cloud-based data

storage and processing. This work presents a deduplication strategy for encrypted data hosted in the cloud, using ownership challenge and proxy re-encryption as the underlying techniques. This suggested approach combines cloud data deduplication with access control to enable efficient data update and sharing, even while the data holders are offline.

The authors (Chia-Mu et al., 2020) have examined the deletion of redundant data and its impact on maintaining single instance storage, while also considering security measures. The ZEUS framework, acronym for zero-knowledge deduplication response, is suggested by the author of this research. This study proposes two deduplication strategies that prioritize privacy. Although ZEUS+ provides greater privacy assurances at a higher cost of communication, it is more economically efficient than ZEUS in terms of privacy guarantees. The privacy-aware deduplication strategy prevents the attacker from accessing the existence status information through duplicate checks.

3. OBJECTIVES OF THE STUDY

1. To improve dependability and availability.
2. To strengthen the cloud computing paradigm's security issues.

4. RESEARCH METHODOLOGY

Scalability, resource provisioning, stability, fault tolerance, and sustainability are just a few of the computational services provided by cloud computing. To ensure that all of these characteristics are applicable, cloud systems must be evaluated before being implemented or deployed in the real world. The main explanation for this is that it is very difficult to change the constraints that exist during real-time execution. It can result in provider overhead, increased costs, and time waste. Simulation is the most effective way to escape any of these frustrations. Another advantage of simulation tools is that they can be used to test user-defined hypotheses prior to software creation. It also discusses the efficiency and bottlenecks of the built framework before it is deployed in a real-world cloud environment. Both cloud customers and cloud providers benefit from these simulation tools.

Cloud Sim, the first simulation platform, was published in 2010 by the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne's Computer Science and Software Engineering Department. Following CloudSim, many other simulators such as Cloud Analyst, Emu Sim, DC Sim, and iCan Cloud have developed over time to make custom application development easier. In the following section, a few main simulators related to this project will be discussed.

5. RESULTS AND DATA INTERPRETATION

Algorithm- IDS based on CSA with PSO

The planning and implementation of intrusion detection systems employing the CSA algorithm are examined in this section. The intrusion detection system's (IDS) architecture and how CSA improves it are discussed in this chapter. PSO maximises the usage of IDS memory.

Algorithm

Step 1: Begin
Step 2: Upload the dataset.csv file in step two.
- Data Categorisation and Numbering
- Specify the function for attack names and numbers as well as attack labels.
- Use attack numbers and names.

Step 3: Cuckoo Structure Initialisation
Step 4: Establish the Fitness Function
Step 5: Use the Cuckoo Search Algorithm to Choose Features
Step 6: SVM and Naive Bayes Classifier Training
- Divide the data into test and train sets.
- Learn SVM and Naïve Bayes

Step 7: Assessing SVM and Naive Bayes Models
- Analyse the model's performance and metrics.

Step 8: Enhance Replication of Cloud Storage
- Use PSO to Choose Features
- Use specific features to train the NB and SVM classifiers.
- Assess the models' cloud storage replication performance.

Step 9: Determine the Memory Usage Prior to and Following PSO Application
- Show the amount of memory used in kilobytes.

Step 10: Show the Outcomes
Step 11: Finish

Algorithm-IDS based on ABC with PSO

This section examines the intrusion detection system's (IDS) planning and implementation using the ABC algorithm. This chapter examines the intrusion detection system's (IDS) fundamental components and shows how ABC enhances intrusion detection capabilities. It is discussed how the IDS's memory usage can be optimised with the use of G-PSO.

Algorithm

Step 1: Begin
Step 2: Upload the dataset.csv file in step two.
- Data Categorisation and Numbering
- Specify the function for attack names and numbers as well as attack labels.
- Use attack numbers and names.

Step 3: Bee Structure Initialisation
Step 4: Establish the Fitness Function
Step 5: Use the Bee Algorithm to Choose Features
Step 6: SVM and Naive Bayes Classifier Training
- Divide the data into test and train sets.
- Learn SVM and Naïve Bayes

Step 7: Assessing SVM and Naive Bayes Models
- Assess the model's performance and metrics

Step 8: Enhance Replication of Cloud Storage
- Use PSO to Choose Features
- Use specific features to train the NB and SVM classifiers.
- Assess the models' cloud storage replication performance.

Step 9: Determine the Memory Usage Prior to and Following PSO Application
- Show the amount of memory used in kilobytes.

Step 10: Show the Outcomes
Step 11: Finish

Resident Set Size

A storage metric called the Resident Set Size (RSS) quantifies the portion of a process's memory that is kept in RAM, or random access memory, and is not moved to disc. We call this area of memory the "resident cache." Put differently, the Remote System Support (RSS) indicates the amount of tangible storage that an activity is actively utilising. The idea is explained as follows:

A process or program interacts with the user's computer's memory while it is operating. This memory is arranged using pages, which can be stored on disc or in RAM. The phrase "Resident Set" describes the group of all the pages in the process's address space that are presently being kept in RAM. This set consists of the portions of the process's memory that are currently being used. When a computer system uses virtual memory, unused pages can be moved to disc to make room in the random-access memory (RAM) for other operations. This is done in order to efficiently manage the limited quantity of physical memory.

The RSS is a number that shows how much physical memory a process is consuming. Only pages that are now in RAM are taken into account; pages that are moved to disc are not. To improve system efficiency, system administrators may find it essential to keep an eye on an operation's RSS. This is due to the fact that an activity's RSS can be used to gauge how much memory it consumes. The Python computer language's psutil package can be used to obtain information about a process, particularly its RSS.

```
# Get the RSS of the current process
rss_memory = psutil.Process().memory_info().rss
```

Results and Discussion- CSA (Storage)

The intended intrusion detection system (IDS) that used SVM-based CSA achieved a finer level of detection than the proposed IDS that used NB, according to the Enhanced CSA approach with storage.. We conducted ten trials for this study project, each of which included a different amount of data. It has been noted that the superior CSA based on NB was attained by the predicted IDS CSA based on SVM.

Table 5.1: Comparison of Accuracy among Proposed Ids CSA (Storage) Based On SVM and NB

Sr. No.	Data	NB(Storage)	SVM(Storage)
1	500	84	95
2	1000	71	83
3	1500	67	81
4	2000	65	80

5	2500	72	84
6	3000	78	85
7	3500	74	75
8	4000	71	75
9	4500	69	74
10	5000	66	69

CONCLUSION

Cloud dealer's connection clients and cloud specialist co-ops. As an outcome, the client's essential concern is the dependability of the cloud representative. Be that as it may, there is no uniform design set up to guarantee this right now.

Following an extensive examination of the previously mentioned weaknesses, the accompanying versatile, dependable, and adaptable structures were created to guarantee ideal asset assignment at the least conceivable expense, just as hearty information security.

An Agent-based Automated Service Composition (A2SC) Algorithm, which includes demand preparing and mechanized assistance creation stages, has been proposed. This instrument is answerable for finding broad offices, yet in addition for decreasing the cost of virtual machines that are just utilized for on-request benefits.

A proposition has been made for an Autonomous Agent Based Load Balancing (A2LB) Algorithm that gives dynamic burden adjusting to cloud conditions. This specialist based instrument conveys proactive burden computation of all VMs and looks for and oversees reasonable choices from adjoining server farms when a solicitation is taken care of by a server farm. This strategy altogether diminishes the time it takes to reply just as the expense of examining.

A Hybrid Two-Tier Security Engine (HT2SE) has been proposed to guarantee the security of client information put away in the cloud. This system utilizes a mix of symmetric and uneven key calculations to give effective and severe security to client information. Moreover, this technique gives the client full oversight over information unscrambling, taking out the chance of information abuse by the cloud specialist organization.

A Scalability-Providing Agent-Based Framework A specialist based engineering for giving adaptability in distributed computing conditions has been proposed. It incorporates calculations for finding another cloud if the receptive cloud gets overburdened, just as a technique for finding the closest datacenters with the fastest virtual machine reaction time.

REFERENCES

1. Dhumwad, S, Sukhadeve, M, Naik, C, MKN & Prabhu, S (2017), ‘A Peer to Peer Money Transfer Using SHA256 and Merkle Tree’, 23RD Annual International Conference in Advanced Computing and Communications (ADCOM), pp. 40-43.
2. Fan, Y, Chen, J & Wang, L (2016), ‘Energy-Efficient and Latency-Aware Data Placement for Geo-Distributed Cloud Data Centers’, 11th International Conference on Communications and Networking in China. Xi an: Springer, pp. 465-474.
3. Fu, JS, Yun Liu, Han-Chieh Chao, Bharat K. Bhargava & Zhen-Jiang Zhang (2018), ‘Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing’, IEEE Transactions on Industrial Informatics, vol. 14, issue 10, pp. 4519-4528.
4. Fursan Thabit, Abdulrazzaq HA Al-Ahdal & Sudhir Jagtap (2021), ‘A New Lightweight Cryptographic Algorithm for Enhancing Data Security In Cloud Computing’, Global Transitions Proceedings, vol. 2, no. 1, pp. 91-99.
5. Fursan Thabit, Sharaf Alhomdy & Sudhir Jagtap (2021), ‘Security Analysis and Performance Evaluation of a New Lightweight Cryptographic Algorithm for Cloud Computing Environment’, Global Transitions Proceedings, vol. 2, no. 1, pp. 100-110.
6. Gaetan de Rassenfosse & Florian Seliger (2021), ‘Imputation of missing information in worldwide patent data’, Data in Brief, vol. 34, 106615
7. Guljar Shaikh (2015), ‘A Survey on Deduplication Strategies and Storage Systems’, IOSR Journal of Computer Engineering, pp. 85-90.
8. Guohua Tian, Hua Ma, Ying Xie & Zhenhua Liu (2020), ‘Randomized de duplication with ownership management and data sharing in cloud storage’, Journal of Information Security and Applications, vol. 51, 102432.
9. Haifeng Gu, Jianning Zhang, Tian Liu, Ming Hu, Junlong Zhou, Tongquan Wei & Mingsong Chen (2020), ‘DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data’, IEEE Transactions on Reliability, vol. 69, no. 1, pp. 188-202.

10. Hao Feng, Jaime Llorca, Antonia M Tulino & Andreas F Molisch (2018), 'Optimal Dynamic Cloud Network Control', IEEE/ACM Transactions on Networking, vol. 26, no. 5, pp. 2118-2131.
11. Hongwei Li, Rongxing Lu, Liang Zhou, Bo Yang & Xuemin (Sherman) Shen (2014), 'An Efficient Merkle-Tree-Based Authentication Scheme for Smart Grid', IEEE Systems Journal, vol. 8, no. 2, pp. 655-663.
12. Hu, Z, Li, B & Luo, J (2017), 'Time-and cost-efficient task scheduling across geo-distributed data centers', IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 3, pp. 705-718.
13. Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf & Muhammad Ovais Ahmad (2020), 'Fake News Detection Using Machine Learning Ensemble Methods', Complexity, vol. 2020, pp. 1-11.
14. Indrani Balasundaram & Ramaraj, E (2012), 'An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching', International Conference on Communication Technology and System Design, Procedia Engineering, vol. 30, pp. 183-190.
15. Jan Stanek & Lukas Kencl (2018) 'Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage', IEEE Transactions On Dependable And Secure Computing, vol. 15, issue 4, pp. 694-707.
16. Javier Panadero, Mennan Selimi, Laura Calvet, Joan Manuel Marquès & Felix Freitag (2021), 'A two-stage Multi-Criteria Optimization method for service placement in decentralized edge micro-clouds', Future Generation Computer Systems, vol. 121, pp. 90-105.