Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 2, April 2024, pp. 812~820
ISSN: 2302-9285, DOI: 10.11591/eei.v13i2.5181

A survey to build framework for optimize and secure migration and transmission of cloud data

Ravinder Bathini[1,2], Naresh Vurukonda[2]
[1]Department of Information Technology, Vardhaman College of Engineering, Hyderabad, Telangana, India
[2]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur, Andhra Pradesh, India

Article Info

Article history:

Received Nov 8, 2022
Revised Sep 18, 2023
Accepted Oct 5, 2023

Keywords:

Cloud computing
Cloud technology
Data migration
Framework
Load balancing
Secure transmission

ABSTRACT

In the recent era of computational technologies, the internet is needed daily. The data generated is enormous and primarily stored on dedicated servers or clouds. Data migration and transfer are significant tasks for maintaining consistency and updating data. The data is the most critical component in any cloud service. There are various methods to protect data, like secure transfer, encryption, and authentication. These techniques are used as per need and transmission of the data. As data grows on a server or cloud, it must be migrated securely. Here, the exhaustive survey is provided for building a framework for migrating and transmitting cloud data. The framework should be sustainable and adaptable for load-balancing recovery and secure transmission. Various security load balancing parameters must be considered to obtain these state-of-the-art functionalities in the framework. The existing similar frameworks are studied, and findings are proposed in the paper to develop the framework.

Corresponding Author:

Naresh Vurukonda
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation
Vaddeswaram, Guntur, Andhra Pradesh, India
Email: nareshvurukonda789@gmail.com

1. INTRODUCTION

The emergence of cloud technologies comes with advantages and disadvantages. Cloud technology is beneficial in various factors over dedicated servers like security, load balancing (LB), and optimization, while some are challenges, such as the number of customers using the cloud. IT industries are developing and rigorously utilizing the cloud for application development and data storage, which come up with different questions related to the same factors as security and all. Real-time applications need fast computation and data access; overburdening secure layers may lose purpose [1], [2].

The recent decade's development in cloud computing has created various research in these areas, which needs different architectures and frameworks in cloud technology, which may enable modeling, simulation, and infrastructure to be secure, sustainable, and optimized for various applications and their needs. Most cloud users need fast, secure, easy customization, and configuration without considering technological backend features such as migration, LB, and virtual machine (VM) management. Some recent development uses intelligence in different layers to provide needed features like heuristic in migration to nodes. Other uses swarm intelligence for it. These intelligent algorithms help in providing fast LB in VMs. Cloud services (infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS)) have other challenges and need different solutions. Cloud service providers (CSP) must consider all these issues and challenges [3], [4].

Hence, we are motivated to provide a generic framework to provide state-of-the-art solutions for different layers of problems. The exhaustive survey indicates different parameters to provide various types of solution integration in the framework. This integration with the optimal utility to maintain fast responses is a key feature of the proposed framework. The application-specific customization is another advantage of the framework, so need-based customization and optimization is a standout feature of the proposed framework.

2. PROPOSED METHOD

A proposed framework for optimal end-to-end secure access shown in Figure 1. From the literature survey, as per need, the optimal best algorithm selection module is provided in the framework. Existing end-to-end transmission control protocol/internet protocol (TCP-IP) protocol-based security is identified. Optimal secure algorithms are incorporated to provide layer-based end-to-end security. The secure protocol suite with two-way data storage encryption will improve security on different brute force and middleman attacks. The LB algorithm selection provides an optimal algorithm selection per resource utilization and VM's ideal time.

Similarly, recovery algorithm selection is based on deployment, user, and cloud type. This will improve access and security to data and applications. The significant improvement with this framework is algorithm selection for fast response and maintaining end-to-end security.

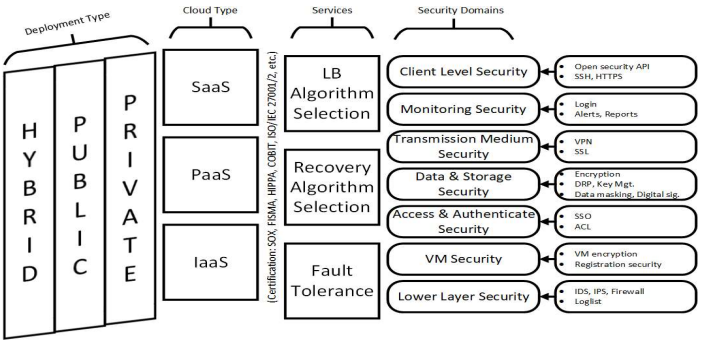

Figure 1. End-to-end optimal cloud security framework

3. OPEN CLOUD

As discussed in the introduction, several cloud services are classified into IaaS, PaaS, and SaaS-based on their utilities and the cloud services shown in Figure 2. Here, the literature survey mainly focuses on fault tolerance and recovery, LB, and security. Some open-source cloud simulators are being used for experimental purposes. Their features are described in Table 1.

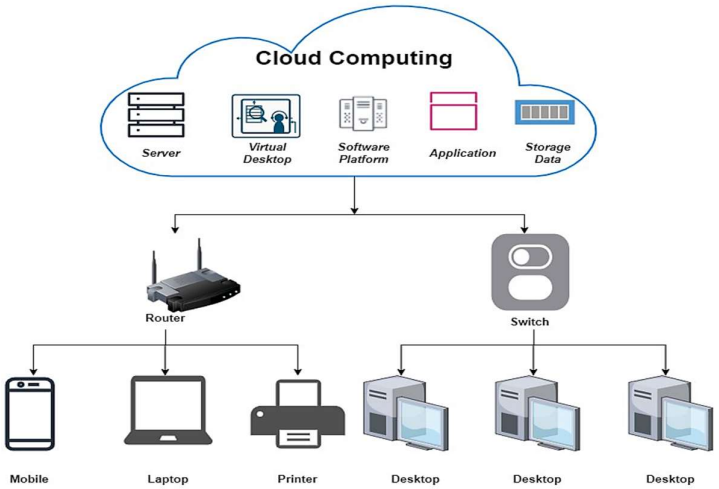

Figure 2. Cloud services

Based on their utilities, different services in the ever-changing cloud computing landscape are divided into three categories: SaaS, PaaS, and IaaS. These divisions are essential in defining the features and services provided by cloud services, as Figure 2 shows. This survey's investigation of fault tolerance and recovery techniques in cloud systems is one of its main goals. Given the inherent complexity of distributed systems, it is critical to comprehend how cloud services manage errors and bounce back from disturbances. The assessment assesses best practices and current methodology in this field, illuminating novel strategies to improve system resilience. Understanding and developing cloud services require experimentation. In light of this, the survey investigates the use of free and open-source cloud simulators for testing. The features of these simulators are listed in Table 1, which also provide an overview of their capabilities and how they support the creation and testing of cloud-based solutions.

Table 1. Features of cloud simulators

Simulator	Features			
	Language	Type	GUI support	Stimulation time
CloudSim [4], [5]	Java	Open source	Limited	Second
Green cloud [6]	C++	Open source	Limited (support via nam)	Minute
iCanCloud [7]	C++	Open source	Not limited	Second

4. LOAD BALANCING

It is one of the key tasks in cloud computing, sometimes called migration. Applications and data are being migrated from nodes or VMs based on policies mentioned in Figure 3. Location, information, selection, and transfer-based policies represent different load-balancing needs. Location policy indicates geographical data migration, like country data can be migrated from one to another for fast access. Similarly, other policies are designed for the migration of data [8], [9].

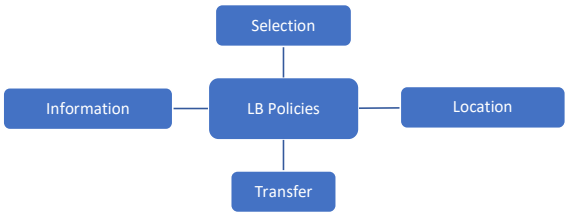

Figure 3. LB policies

There are LB techniques shown in Figure 4. Based on system state and initiation, there are sub-types discussed. Sender, receiver, and symmetric initiation typically represent LB initiation started. The same algorithmic strategies used for static and dynamic LB are shown under system state-based LB [10]–[12]. Tables 2 and 3 shows LB algorithm comparison.

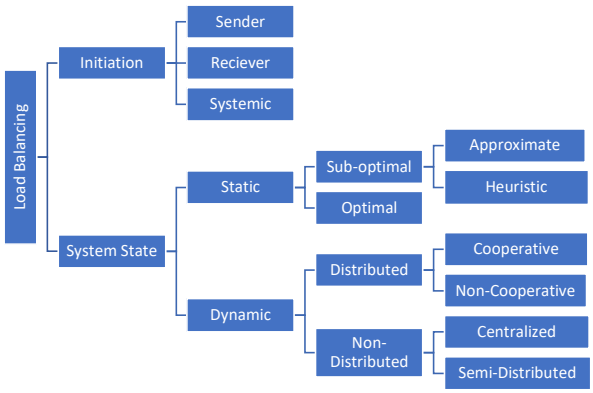

Figure 4. LB techniques

Table 2. LB procedures comparison

Scheduling algorithms	Merits	Demerits
Throttled LB	Worthy performance; and the list manages tasks	Tasks want to have waited
Carton	Worthy performance; and the list manages tasks	Tasks want to have waited
Ant-colony	It required low communication; equal distribution of responses; good performance; and fairness	It depends upon lower costs
Honey bee	Faster data collection; minimizes makespan; independent tasks; and computationally intensive	The search takes a long time; an unknown number of ants; and the network needs to be fixed
Dynamic LB	Response time reduced; and upturns throughput	With a VM machine, high-importance tasks can be made
Max–min	Allocate work at runtime; and current state fault tolerance.	More difficult; and want nodes constant check
Min-min	Necessities are previously known	The average waiting time is longer
Round robin	The completion time value needs to be improved; and it displays the best result in the presence of more minor tasks	Tasks and machine differences can't be expected; and starvation
Static LB	Also used priority; fairness performs better for short CPU bursts; easy to understand; and fixed time quantum	More context switch; and larger tasks take a long time
	Fewer complexes; divides the traffic equally; and compile time lb	No changes at runtime; and it is restricted to load variations

Table 3. LB algorithms comparison continue

LB algorithms	Fairness	Response time	Throughput	Overhead	Fault tolerance	Performance	Resource utilization	Speed	Complexity
Static [9]–[13]	✓	High speed	Large	N/A	×	High speed	Large	High speed	Less
Round Robin [14]	✓	High speed	Large	Large	×	High speed	Large	N/A	Less
Min-Min [15], [16]	×	High speed	Large	Large	×	High speed	Large	Fast	Less
Max-Min [17]	×	High speed	Large	Large	×	High speed	Large	High speed	Less
Dynamic [18], [19]	×	Slow	Large	Large	✓	Slow	Large	High speed	High speed
Honey bee [20]	×	Slow	Large	Minimum	×	Slow	Large	High speed	Less
Ant colony [21]	×	Slow	Large	Large	N/A	Slow	Large	High speed	×
Carton [22]	✓	High speed	Large	N/A	N/A	High speed	Large	High speed	High speed
Throttle [23]	×	High speed	Less	Minimum	✓	High speed	Large	High speed	Less
OLB+LBMM [24]	×	Slow	Large	Minimum	×	High speed	Large	Slow	High speed

5. RECOVERY AND FAULT TOLERANCE

Different CSPs use different recovery mechanisms in case of data loss. A comparative analysis of such mechanisms is given in Table 4 based on their properties. Similarly, different architectures have different policies for fault tolerance compared to Table 5 for disaster recovery (DR).

Table 4. Cloud recovery mechanism and properties

Cloud-based DR systems	User premises backup	Dual-role operation	Multi-tier	Multiple back-ups	Shared data storage	Security techniques	Quorum host	Live VM migration	Knowledge-based DR service	Pipeline replication
SecondSite [25]	✓	✓	×	✓	×	×	✓	×	×	×
Remus [26]	×	×	×	×	×	×	×	✓	×	×
Romulus [27]	×	×	×	×	×	×	×	✓	×	×
DT-enabled cloud architecture [28]	×	×	×	×	×	×	×	✓	×	✓
Kemari [29]	×	×	×	×	×	×	×	✓	×	×
RUBiS [30]	×	×	✓	×	×	×	×	×	×	×
Taiji [31]	×	×	×	×	✓	×	×	✓	×	×
HS-DRT system [32]	×	×	✓	✓	×	✓	×	×	×	×
PipeCloud [30]	×	×	×	×	×	×	×	✓	×	✓
Disaster-CDM [33]	×	×	×	×	×	×	×	×	✓	×
Distributed cloud system architecture [34]	×	✓	×	×	✓	×	×	×	×	×

CSPs use several recovery techniques to lessen the effects of data loss. A thorough comparison of these systems based on their salient characteristics is shown in Table 4. It takes into account things like scalability, consistency of data, and recovery time. It's critical for firms to comprehend these attributes in order to select a CSP that best suits their unique recovery needs. Fault tolerance policies play a crucial role in guaranteeing the robustness of cloud infrastructures, even beyond data recovery. Table 5 lists the various fault tolerance strategies that various architectures have chosen. This covers methods to deal with system failures and DR tactics. Organizations that want to continue operating smoothly even in the face of unforeseen difficulties must evaluate these policies. The ability of a company to recover from data loss and survive system outages is greatly impacted by the cloud service provider that it chooses. Organizations can make decisions that are relevant to their needs by comparing the recovery techniques and fault tolerance rules shown in Tables 4 and 5.

Table 5. Cloud fault tolerance comparison

Fault tolerance architectures for cloud computing	Fault recovery			Techniques and policies					Fault detection			Main features/usage
				Proactive			Reactive					
	Fault mask	Node recovery	System recovery	Self-healing	Preemptive migration	Checkpoint/restart	Replication	Job migration	Self-detection	Other detection	Group detection	
Map–Reduce [35]	×	×	✓	✓	✓	×	×	×	×	✓	×	Big data processing
Haproxy [36]	×	×	✓	×	×	×	✓	✓	×	✓	×	Uninterrupted
BFT-Cloud [37]	×	×	✓	×	×	×	✓	×	×	×	✓	Arbitrary fault detection
Gossip [38]	×	×	✓	×	×	×	✓	×	✓	×	✓	Optimize than byzantine
MPI [39]	×	×	✓	×	×	✓	×	✓	×	✓	×	For parallel programming
FTM [40]	✓	×	✓	×	×	✓	✓	✓	✓	✓	×	Full policy
PLR [41]	×	×	✓	×	×	✓	✓	✓	✓	×	×	Real-time HPC
FTM-2 [42]	×	×	✓	×	×	✓	✓	×	×	✓	×	Detached fault detector
LLFT [43]	×	✓	×	×	×	×	✓	×	×	×	✓	Low latency
AFTRC [44]	✓	×	✓	×	×	✓	✓	✓	×	✓	×	Real-time HPC
FT-Cloud [45]	×	×	✓	✓	×	×	×	×	×	✓	×	Component ranking based
FTWS [46]	×	×	✓	×	×	✓	✓	×	×	✓	×	Workflow
Vega Warden [47]	×	✓	×	×	×	×	✓	✓	×	✓	×	Virtualization
Magi Cube [48]	×	×	✓	×	×	×	✓	✓	×	✓	×	Low redundancy
Candy [49]	×	×	✓	×	×	×	✓	×	×	✓	×	Component-based

6. SECURITY

Security is the most crucial issue in dedicated server vs cloud comparison. Different aspects need to be considered before comparison. There are some characteristics [50] of cloud storage given in Table 6. Based on these storage characteristics, data is stored in the cloud. The stored data needed to be protected; hence, there are some security groups and problems [51] described in Table 7. Table 8 represents the cloud security framework comparison.

Table 6. Cloud storage characteristics

Characteristic	Description
Control	Capacity to control a system for different parameters like configuration, costing, and performance
Manageability	Achieve a system with minimal resources
Scalability	Capability to scale to get higher demands
Access technique	Protocol over which cloud storage is unprotected
Data availability	System's uptime measure
Multi-tenancy	Support for multiple users
Storage efficiency	They are measured on storage parameters such as speed, redundancy, and data errors.
Cost	Measure based on space needed and used

There is a growing controversy in the field of data management regarding dedicated servers vs cloud storage. Security is a key component of this conversation since businesses want to safeguard their important data. Prior to comparing cloud computing to dedicated servers, it is critical to take into account a number of factors that influence the security paradigm. The fundamental features of cloud storage are listed in Table 6, which forms the basis for cloud data management. These attributes—which range from accessibility to scalability—have a significant impact on the storage market. Still, security continues to be the main focus, highlighting the necessity of strong defenses. Cloud data storage necessitates a closer attention to security measures. Table 7 explores security groups and possible issues pertaining to cloud storage. Every component, including access controls and encryption techniques, helps to protect the confidentiality and integrity of data that is stored.

Table 7. Cloud security categories 3

Category	Description	Label	Issues
Security standards	Describes the standards to prevent attacks as per government laws	I1	Security standard's absence
		I2	Risks of compliance
		I3	Auditing absence
		I4	Legal aspects absence (service level agreement)
		I5	Trust
Network	Network attacks include connection, denial of service (DoS), DDoS, and availability	I6	Network firewall's appropriate installation
		I7	Configurations of security of network
		I8	Vulnerabilities of internet protocol
		I9	Dependence of internet
Access	Attacks related to authentication and access control	I10	Service hijacking and account
		I11	Malevolent insiders
		I12	Authentication appliance
		I13	Restricted user admittance
		I14	Security of browser
Cloud infrastructure	Attacks on cloud infrastructure like tampered privileged and binaries insiders.	I15	API's insecure interface
		I16	Service quality
		I17	Technical fault's sharing
		I18	Supplier's dependability
		I19	Misconfiguration of security
		I20	Multi-tenancy
		I21	Backup and server location
Data	Security issues like data migration, integrity, and confidentiality	I22	Redundancy of data
		I23	Information leakage and loss
		I24	Location of data
		I25	Recovery of data
		I26	Privacy of data
		I27	Protection of data
		I28	Availability of data

Table 8. Cloud security framework comparison

Item/framework	Layer	Function	Security goal	Infrastructure	Approach	Technology	Application	Architecture	Collaboration
Wang *et al.* [52]	Yes	Yes	Yes	Yes	NA	Yes	Yes	NA	Yes
Talib *et al.* [53]	Yes	Yes	Yes	Yes	Yes	Yes	NA	Yes	Yes
Takabi *et al.* [54]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	NA	Yes
Yu *et al.* [55]	NA	NA	Yes	Yes	Yes	Yes	Yes	NA	Yes
Du *et al.* [56]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Venkatesan and Vaish [57]	NA	Yes	Yes	Yes	Yes	Yes	NA	Yes	Yes

7. CONCLUSION

The proposed framework provides layer-based end-to-end protection for data and applications in the cloud. A selection of load-balancing algorithms at the runtime provides optimal migration. These dynamic algorithm selection modules for LB, fault tolerance framework and recovery help optimally execution for migration, security, and data transmission. The various certification as per domain security ensures data privacy. A significant improvement will be observed in algorithm selection as per the need of resources and system, VM ideal time. This will improve the throughput of the system without compromising security. The proposed framework is an open security structure that includes two-way protection at each layer of end-to-end security. The performance major is still a significant concern and will be calculated based on different parameters for domains like security, migration, and throughput.

REFERENCES

[1] H. Yeganeh, A. Salahi, and M. A. Pourmina, "A Novel Cost Optimization Method for Mobile Cloud Computing by Capacity Planning of Green Data Center with Dynamic Pricing," *Canadian Journal of Electrical and Computer Engineering*, vol. 42, no. 1, pp. 41–51, 2019, doi: 10.1109/CJECE.2019.2890833.

[2] T. Saxena and V. Chourey, "A survey paper on cloud security issues and challenges," in *Proceedings of the 2014 Conference on IT in Business, Industry and Government: An International Conference by CSI on Big Data, CSIBIG 2014*, IEEE, Mar. 2014, pp. 1–5, doi: 10.1109/CSIBIG.2014.7056957.

[3] C. Thiam and F. Thiam, "An Energy-Efficient VM migrations optimization in Cloud Data Centers," in *IEEE AFRICON Conference*, IEEE, Sep. 2019, pp. 1–5, doi: 10.1109/AFRICON46755.2019.9133776.

[4] B. Priya and T. Gnanasekaran, "Optimization of Cloud Data Center using CloudSim-A methodology," in *2019 Proceedings of the 3rd International Conference on Computing and Communications Technologies, ICCCT 2019*, 2019, pp. 307–310, doi: 10.1109/ICCCT2.2019.8824950.

[5] R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," *arXiv preprint arXiv:0903.2525*, 2009, doi: 10.48550/arXiv.0903.2525.

[6] L. Liu *et al.*, "Greencloud: A new architecture for green data center," in *Proceedings of the 6th International Conference Industry Session on Autonomic Computing and Communications Industry Session, ICAC-INDST'09*, New York, NY, USA: ACM, Jun. 2009, pp. 29–38, doi: 10.1145/1555312.1555319.

[7] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, "ICanCloud: A Flexible and Scalable Cloud Infrastructure Simulator," *Journal of Grid Computing*, vol. 10, no. 1, pp. 185–209, Mar. 2012, doi: 10.1007/s10723-012-9208-5.

[8] V. N. Volkova, L. V. Chemenkaya, E. N. Desyatirikova, M. Hajali, A. Khodar, and A. Osama, "Load balancing in cloud computing," in *Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2018*, IEEE, Jan. 2018, pp. 387–390, doi: 10.1109/EIConRus.2018.8317113.

[9] R. Lee and B. Jeng, "Load-balancing tactics in cloud," in *Proceedings - 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2011*, IEEE, Oct. 2011, pp. 447–454, doi: 10.1109/CyberC.2011.79.

[10] R. Rathore, B. Gupta, V. Sharma, and K. K. Gola, "A New Approach For Load Balancing In Cloud Computing," *International Journal of Computers & Technology*, vol. 13, no. 12, pp. 5193–5198, 2014, doi: 10.24297/ijct.v13i12.5277.

[11] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 50–71, Jun. 2017, doi: 10.1016/j.jnca.2017.04.007.

[12] S. Khare, U. Chourasia, and A. J. Deen, "Load Balancing in Cloud Computing," in *Cognitive Science and Technology*, 2022, pp. 601–608, doi: 10.1007/978-981-19-2350-0_58.

[13] R. Tong and X. Zhu, "A load balancing strategy based on the combination of static and dynamic," in *2010 2nd International Workshop on Database Technology and Applications, DBTA2010 - Proceedings*, IEEE, Nov. 2010, pp. 1–4, doi: 10.1109/DBTA.2010.5658951.

[14] U. Dubey and L. S. Songare, "Analysis of load balancing in cloud computing," *International Journal of Scientific and Technology Research*, vol. 8, no. 12, pp. 3912–3914, Jan. 2019.

[15] S. S. Chauhan and R. C. Joshi, "A weighted mean time min-min max-min selective scheduling strategy for independent tasks on grid," in *2010 IEEE 2nd International Advance Computing Conference, IACC 2010*, IEEE, Feb. 2010, pp. 4–9, doi: 10.1109/IADCC.2010.5423047.

[16] H. Chen, F. Wang, N. Helian, and G. Akanmu, "User-priority guided min-min scheduling algorithm for load balancing in cloud computing," in *2013 National Conference on Parallel Computing Technologies, PARCOMPTECH 2013*, IEEE, Feb. 2013, pp. 1–8, doi: 10.1109/ParCompTech.2013.6621389.

[17] U. Bhoi and P. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing," *International Journal of Application or Innovation in Engineering & Management*, vol. 2, no. 4, pp. 259–264, 2013.

[18] N. Haryani and D. Jagli, "Dynamic Method for Load Balancing in Cloud Computing," *IOSR Journal of Computer Engineering*, vol. 16, no. 4, pp. 23–28, 2014, doi: 10.9790/0661-16442328.

[19] N. Raghava and D. Singh, "Comparative Study on Load Balancing Techniques in Cloud Computing," *Open Journal of Mobile Computing and Cloud Computing*, vol. 1, no. 1, pp. 18–25, 2014.

[20] L. D. D. Babu and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2292–2303, May 2013, doi: 10.1016/j.asoc.2013.01.025.

[21] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in *Proceedings - 2011 6th Annual ChinaGrid Conference, ChinaGrid 2011*, IEEE, Aug. 2011, pp. 3–9, doi: 10.1109/ChinaGrid.2011.17.

[22] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *Proceedings - 3rd International Symposium on Parallel Architectures, Algorithms and Programming, PAAP 2010*, IEEE, Dec. 2010, pp. 89–96, doi: 10.1109/PAAP.2010.65.

[23] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in *24th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2010*, IEEE, 2010, pp. 551–556, doi: 10.1109/WAINA.2010.85.

[24] K. K. Sowjanya and S. K. Mouleeswaran, "Load Balancing Algorithms in Cloud Computing," *Cognitive Science and Technology*, vol. 3, no. 2, pp. 483–493, 2023, doi: 10.1007/978-981-19-2358-6_45.

[25] S. Rajagopalan, B. Cully, R. O'Connor, and A. Warfield, "SecondSite: Disaster tolerance as a service," *ACM SIGPLAN Notices*, vol. 47, no. 7, pp. 97–107, 2012, doi: 10.1145/2365864.2151039.

[26] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield, "Remus: High availability via asynchronous virtual machine replication," *5th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2008*, pp. 161–174, 2008.

[27] M. C. Caraman, S. A. Moraru, S. Dan, and D. M. Kristaly, "Romulus: Disaster tolerant system based on kernel virtual machines," *Annals of DAAAM and Proceedings of the International DAAAM Symposium*, pp. 1671–1672, 2009.

[28] M. C. Caraman, S. A. Moraru, S. Dan, and C. Grama, "Continuous Disaster Tolerance in the IaaS clouds," in *Proceedings of the International Conference on Optimisation of Electrical and Electronic Equipment, OPTIM*, IEEE, May 2012, pp. 1226–1232, doi: 10.1109/OPTIM.2012.6231987.

[29] H. Meng, J. Li, W. Liu, and C. Zhang, "MMSD: A metadata-aware multi-tiered source deduplication cloud backup system in the personal computing environment," *International Review on Computers and Software*, vol. 8, no. 2, pp. 542–550, 2013.

[30] T. Wood, E. Cecchet, K. K. Ramakrishnan, P. Shenoy, J. van der Merwe, and A. Venkataramani, "Disaster recovery as a cloud service: Economic benefits & Deployment challenges," *2nd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud 2010*, vol. 10, pp. 8–15, 2010.

[31] J. Zhu, Z. Jiang, Z. Xiao, and X. Li, "Optimizing the performance of virtual machine synchronization for fault tolerance," *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1718–1729, Dec. 2011, doi: 10.1109/TC.2010.224.

[32] Y. Ueno, N. Miyaho, S. Suzuki, and K. Ichihara, "Performance evaluation of a disaster recovery system and practical network system applications," in *Proceedings - 5th International Conference on Systems and Networks Communications, ICSNC 2010*, IEEE, Aug. 2010, pp. 195–200, doi: 10.1109/ICSNC.2010.37.

[33] K. Grolinger, M. A. M. Capretz, E. Mezghani, and E. Exposito, "Knowledge as a service framework for disaster data management," in *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, IEEE, Jun. 2013, pp. 313–318, doi: 10.1109/WETICE.2013.48.

[34] B. Silva, P. Maciel, E. Tavares, and A. Zimmermann, "Dependability models for designing disaster tolerant cloud computing systems," in *Proceedings of the International Conference on Dependable Systems and Networks*, IEEE, Jun. 2013, pp. 1–6, doi: 10.1109/DSN.2013.6575323.

[35] Q. Zheng, "Improving MapReduce fault tolerance in the cloud," in *Proceedings of the 2010 IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum, IPDPSW 2010*, IEEE, Apr. 2010, pp. 1–6, doi: 10.1109/IPDPSW.2010.5470865.

[36] V. Kaushal and M. Bala, "Autonomic fault tolerance using haproxy in cloud environment," *International Journal Of Advanced Engineering Sciences And Technologies*, vol. 72, no. 7, pp. 222–227, 2010.

[37] Y. Zhang, Z. Zheng, and M. R. Lyu, "BFTCloud: A Byzantine Fault Tolerance framework for voluntary-resource cloud computing," in *Proceedings - 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011*, IEEE, Jul. 2011, pp. 444–451, doi: 10.1109/CLOUD.2011.16.

[38] J. Lim, J. Lee, S. Chin, and H. Yu, "Group-based gossip multicast protocol for efficient and fault tolerant message dissemination in clouds," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, pp. 13–22, doi: 10.1007/978-3-642-20754-9_3.

[39] J. L. Reyes-Ortiz, L. Oneto, and D. Anguita, "Big data analytics in the cloud: Spark on Hadoop vs MPI/OpenMP on Beowulf," *Procedia Computer Science*, vol. 53, no. 1, pp. 121–130, 2015, doi: 10.1016/j.procs.2015.07.286.

[40] R. Jhawar, V. Piuri, and M. Santambrogio, "A comprehensive conceptual system-level approach to fault tolerance in Cloud computing," *SysCon 2012 - 2012 IEEE International Systems Conference, Proceedings*, pp. 601–605, 2012, doi: 10.1109/SysCon.2012.6189503.

[41] P. KumarPatra, H. Singh, and G. Singh, "Fault Tolerance Techniques and Comparative Implementation in Cloud Computing," *International Journal of Computer Applications*, vol. 64, no. 14, pp. 37–41, 2013, doi: 10.5120/10705-5643.

[42] W. Zhao, P. M. Melliar-Smith, and L. E. Moser, "Fault tolerance middleware for cloud computing," in *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, IEEE, Jul. 2010, pp. 67–74, doi: 10.1109/CLOUD.2010.26.

[43] T. Nadu, "Fault tolerant workflow scheduling based on replication and resubmission of tasks in Cloud Computing," *International Journal on COmputer Science and Engineering*, vol. 4, no. 06, pp. 996–1006, 2012.

[44] F. Machida, E. Andrade, D. S. Kim, and K. S. Trivedi, "Candy: Component-based availability modeling framework for cloud service management using SysML," in *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, IEEE, Oct. 2011, pp. 209–218, doi: 10.1109/SRDS.2011.33.

[45] B. Mohammed, M. Kiran, K. M. Maiyama, M. M. Kamala, and I. U. Awan, "Failover strategy for fault tolerance in cloud computing environment," *Software - Practice and Experience*, vol. 47, no. 9, pp. 1243–1274, 2017, doi: 10.1002/spe.2491.

[46] S. M. Hosseini and M. G. Arani, "Fault-Tolerance Techniques in Cloud Storage: A Survey," *International Journal of Database Theory and Application*, vol. 8, no. 4, pp. 183–190, 2015, doi: 10.14257/ijdta.2015.8.4.19.

[47] S. Prathiba and S. Sowvarnica, "Survey of failures and fault tolerance in cloud," in *Proceedings of the 2017 2nd International Conference on Computing and Communications Technologies, ICCCT 2017*, IEEE, Feb. 2017, pp. 169–172, doi: 10.1109/ICCCT2.2017.7972271.

[48] S. Malik and F. Huet, "Adaptive fault tolerance in real time cloud computing," in *Proceedings - 2011 IEEE World Congress on Services, SERVICES 2011*, IEEE, Jul. 2011, pp. 280–287, doi: 10.1109/SERVICES.2011.108.

[49] M. Hussain and S. Farid, "Fault Tolerance Techniques in Cloud and Distributed Computing- A Review," *Precision Engineering*, vol. 22, no. 4, pp. 56–67, 2017.

[50] J. Velmurugan and P. A. Kumar, "Survey on data storage security in cloud computing," *International Journal of Applied Engineering Research*, vol. 9, no. 22, pp. 13299–13316, 2014.

[51] I. M. Khalil, A. Khreishah, and M. Azeem, "Cloud computing security: A survey," *Computers*, vol. 3, no. 1, pp. 1–35, Feb. 2014, doi: 10.3390/computers3010001.

[52] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," *IEEE International Workshop on Quality of Service, IWQoS*, vol. 186, no. 978, pp. 1–9, 2009, doi: 10.1109/IWQoS.2009.5201385.

[53] A. M. Talib, R. Atan, R. Abdullah, and M. A. A. Murad, "Formulating a Security Layer of Cloud Data Storage Framework Based on Multi Agent System Architecture," *Gstf International Journal on Computing*, vol. 1, no. 1, pp. 120–124, 2010, doi: 10.5176/2010-2283_1.1.20.

[54] H. Takabi, J. B. D. Joshi, and G. J. Ahn, "SecureCloud: Towards a comprehensive security framework for cloud computing environments," in *Proceedings - International Computer Software and Applications Conference*, IEEE, Jul. 2010, pp. 393–398, doi: 10.1109/COMPSACW.2010.74.

[55] H. Yu, N. Powell, D. Stembridge, and X. Yuan, "Cloud computing and security challenges," in *Proceedings of the Annual Southeast Conference*, New York, NY, USA: ACM, Mar. 2012, pp. 298–302, doi: 10.1145/2184512.2184581.

[56] J. Du, W. Wei, X. Gu, and T. Yu, "RunTest: Assuring integrity of dataflow processing in cloud computing infrastructures," in *Proceedings of the 5th International Symposium on Information, Computer and Communications Security, ASIACCS 2010*, New York, NY, USA: ACM, Apr. 2010, pp. 293–304, doi: 10.1145/1755688.1755724.

[57] S. Venkatesan and A. Vaish, "Multi-agent based dynamic data integrity protection in cloud computing", *in Communications in Computer and Information Science (CCIS)*, Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 142, 2011, doi: 10.1007/978-3-642-19542-6_13.

BIOGRAPHIES OF AUTHORS

Ravinder Bathini is an assistant professor in Department of Information Technology, Vardhaman College of Engineering, Kacharam, Shamshabad–501218 Hyderabad, Telangana, India and research scholar in KL University, Green Fields, Vaddeswaram, Andhra Pradesh, India. His research area includes green cloud computing, cloud analytics, cloud deployment models, and cloud security. He can be contacted at email: bravinder552@gmail.com.

Naresh Vurukonda is associate professor Department of Computer Science and Engineering at KL University, Green Fields, Vaddeswaram, Andhra Pradesh, India. His main research subject is cloud computing and big data. He can be contacted at email: nareshvurukonda789@gmail.com.