International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I6P201

Volume 70 Issue 6, 1-12, June 2022

Original Article

Provisioning of Defensing Mechanism Against Threats During VM Migration in Cloud Environment

Nelli Chandrakala[1], Vamsidhar Enireddy[2]

[1,2]*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation Vaddeswaram, AP, India.*

Received: 29 March 2022 Revised: 26 May 2022 Accepted: 05 June 2022 Published: 27 June 2022

kala5136@gmail.com

Abstract *- Defensing cloud computing towards security threats is a challenging task. Cloud Computing (CC) acts as an open platform that lays a platform for attackers with malicious activities from unauthorized users targeting the host. VM migration is a target defense mechanism that mitigates the attackers and offers superior VM position management. Moreover, there is an apparent demand to concentrate on security benefits to the VM migration considering cloud system architecture. This research intends to fill the gap using a random VM model to provide the security gap by state action. The security metrics include measuring the attack rate over the random VM field. The simulation is done in the MATLAB 2020a environment with the unavailability measure of VM migration. The major research contributions are the probability of analyzing the attack rate over the configured system and the selection of VM based on attack tolerance level. The outcomes are validated against simulation outcomes to confirm the level of prediction.*

Keywords *- Cloud computing, VM migration, Threat, Attack probability, Random field model.*

1. Introduction

Security and reliability in cloud technology are critical problems for academics and industry. According to existing research, cloud computing technology is on customers' minds [1]. In the public Cloud, ensuring high reliability is also an important research direction. There is a particular need to build a detailed model for evaluating cloud reliability [2]. The earlier research has identified signs of computer aging [3]. Software aging is a gradual result of crashes or other problems [4], whereas programme rejuvenating is a technique for preventing software aging [5]. Virtual machine migration (VMM) scheduling is a technique for reducing VMM product rejuvenating interruption [6]. However, the security implications of using routing as a complement for VMM rejuvenate are yet unknown. Aside from that, given the unpredictability associated with security incidents, protection software regeneration programming is difficult to implement [8]. The significance of investigating the security implications of software rejuvenation strategies is explained. Several researchers [9] have looked at the possibility of a Routing scheduler supporting VMM software regeneration. These primarily aim to determine the best rejuvenating timetable to maximise accessibility, but neither addresses security problems. From a security point of view, there are several efforts on VM migration protection, such as those given in [10]. However, none of them address the program aging and regeneration issue. The research objective is to assess the security implications of the VM migration schedule as a complement for VMM software rejuvenation. The following research question will be explained in this section: RQmain - What are the privacy implications of using various Routing rules for VMM software reinvention?

• RQs1 - Define the VM approach to immigration that minimises the vulnerabilities to the mechanism?

• RQs2 - When employing VM migration management as support for VM regeneration, what is the exchange regarding protection and freedom?

The three fundamental parts of the network infrastructure are VMs (virtual machines) that execute the desired action, Main Nodes (physical machines that host the VMs), and Emergency Nodes. The chosen design addresses the most important aspects of network virtualization. Complex virtualized infrastructures in Cloud Computing do include main components that are comparable [11]. We derive a safety feature called RiskScore from the suggested availability model. Instead of expecting aggressor behaviour (which is impossible to predict), this statistic is based on the amount of time the program spent in potentially dangerous states. The hypothesis is that the efficiency of an attack is proportional to the amount of spending time in a dangerous condition. As a result, the risk score metric records the time spent in a situation (or state) that elevates or facilitates an effective security attack. As a result, because various security attacks have distinct prerequisites, the RiskScore computation is based on the security concern under consideration (e.g., DoS and MITM).

The security danger of a MITM, DoS attack, and the mixture of both concerns are presented in this study. The various Routing rescheduling options aid in examining the unavailability with the security risk tradeoff examined. Reliability is both reliability and a reliability feature [12]. Both dependable incidents such as failures, breakdowns, or hangs and suspicious activities such as despicable behaviour or vulnerability attacks can impact the system available. The examination of unavailability is done from only the standpoint of reliability [13]. The proposed work presents the following themes :

• The simple and flexible range of security evaluation that considers various threats while maintaining the same service concept.

• The findings enable researchers to investigate a VM migration approach that maximizes security or dependability levels across the board;

This study looks into virtual servers' dependability and vulnerability assessment tradeoff using VM migration. Without changing the accessibility framework, the modification vulnerability assessment method to emerging threat models.

2. Related work

The proposed work examines the literature relevant to three primary research axes relevant to VM migration in cloud computing, contextual security evaluation, and VM migration concerning protection and performance needs. We've also spoken about policy coherence affects the functioning for finding and correcting inconsistencies and conflicts inside a defense policy setup. The following are the most substantial achievements in the three scientific fields. The EDAMP approach for VM migration between different virtualization implementations was presented [13]. To our information, it is the only study that looks at the topic of VM migration in a homogeneous cloud infrastructure. We offer a quick introduction to the EDAMP approach since we analyze the routing challenge in heterogeneity hypervisor architecture. The following are the stages of Virtualization in EDAMP. To distinguish a group of files in each VM, EDAMP creates a local list. The document path, data format, and date and time associated with the VM are all included in this list for each file. Following this step, the approach creates a shared list with various properties of documents utilised by multiple VMs [14]. To create this common list, the hypervisor must combine the virtual servers. The sending and receiving virtual machines are defined in the following phase. The virtualization also provides the needed funds to the resource during this step, and the local VM list is updated accordingly. Following that, EDAMP creates migrations statistics by identifying a difference between the network storage list and recipient VMs (initiation phase). LS denotes the source VM's local file list when allocating resources. The company is to be able to be downloaded using the migration file under the management of the destinations virtualization during the implementation stage. It enables the OS to start in the new location. The inventors of this technique focused solely on VM state movement. Cybersecurity, on the other hand, was not discussed.

The formal technique provided [15] for VMM planning is to discover a series of conversion phases that satisfy all protection, dependence, and parameters. The above conditions may be broken if VMs are migrated in a fractional derivative. The author discusses the VMM-Planner, a paradigm for determining a migration path in which the proposed routing conforms to the goal placement while maintaining dependence, risk, and constraints. VMM-planner converts the Routing scheduling issue into a demand fulfillment problem, encoding all needs, the cloud infrastructure state, and the desired allocation as requirements. After that, a satisfiability modulo theory (SMT) solver is used to resolve the simulated restrictions. If a service operation solution to the particular iteration exists, the solver finds it. The proposed fix is a VM migration sequence that can be used to accomplish a live VM migration effectively. According to our findings, i,) security context is not considered in VMM architecture. (ii) migratory sequence is not time-tagged. (iii) serial migratory step sequenced is mentioned; however, simultaneous migratory step sequences may be explored for improved quality, and (iv) Functional flaws and privacy breaches have not been studied in each phase of the VMM strategy.

Stateful inspection and other network features help enforce public cloud protection. The encryption implemented on the VM must be transferred to the target computer throughout virtualization. A firewall-secured VM environment is provided [16], a methodology for security scenario shifting. The security context transfer technique, as well as its implementations, has been described in their works. The hypervisor's Security Context (SC) module takes the SC from the originating VM and transfers it to the target server along with the VM state. The SC Gartner defines a TCP session here between sender and recipient. The SC module implements the serious effect at the endpoint [17]. The researchers used three possible sample cases to evaluate the work. VM relocation was completed even in the first test scenario. VM was imported using constant SC knowledge in the subsequent test scenario. In the successive scenario, VM containing static and dynamic knowledge was moved. Test cases are insufficient to ensure that the security perspective transfer is valid and consistent.

Furthermore, this technique assumes that the IP address of the VM does not modify when it is moved to a new host. Investigators devised a method for an assailant to take managerial control of multiple cloud applications during route discovery by modifying the identification code via a man-in-the-middle approach. The necessity of protecting the VM migration process was established by [18]. The authors divided the dangers into network control, user plane, and movement component. The developers created a programme called Xensploit to undertake MITM attacks on virtual machines in real-time.

During a deployment, the tool manipulates the memory of a VM as it transits the connection. The XEN platform illustrates simple memory modification and privilege escalation threats. Identity Modification abused the authentication process in VMware Virtual Network [19].

The three crucial elements of the software architecture are the Main Node, VM, and Standby Access point. First is the physical system that executes VM, e VM is the target software, and the Standby Network is utilized as a Routing endpoint. This arrangement is typical in virtualized situations, as previously stated. The computational machines that operate as the Access points of our design process are included in cloud applications such as Openstack1. Our analysis looks at two significant security attacks: i) MITM and ii) DoS attacks. As a result, we opted to focus on the software aging effects in the VMM portion of our initial study. We suggest a rejuvenation technique on VMM to address system aging issues. The objective behind the regeneration approach is to relocate the virtual machine (VM) from a physical computer that is suffering software aging difficulties to another physical machine that is not experiencing technology aging challenges (i.e., Standby Node). To allow VM live migration, we propose a remote storage volume [16]. Such a remote volume preserves the virtual disc of the VM.

Furthermore, the VM live movement technique requires the system to move memory and microprocessor state rather than the entire VM component. We also investigate using a pre-copy approach in the VM migration procedure [16]. The pre-copy procedure may be divided into two phases: 1) copy-phase, which involves assigning assets on the migration goal and transporting physical memory, and 2) timeout, which involves transmitting the processor's information and recognizing the migration. The VM still functions and provides that service during replication [20]. We further emphasize that the costs of Virtual machines are based on the quality of the resources consumed in the VM, notably the number of dirty system resources and the rate of dirty memory locations.

After a VM fails, the network remains unusable. After a VM repair, the server is backing up and running. Because the VM's activities depend on the Main Station, malfunctions on the Central Node impact the VM's availability. The platform recovers from Main Hardware failures by undertaking two phases: repairing Main Terminal and restarting VM. Screw-ups do not cause malfunctions on the Standby Connection point. Nonetheless, failures of the Contingency Nodes hinder the Virtualization technology. The suggested approach also considers problems that aren't caused by software. The following errors are not taken into account by the model: i) network problems; (ii) computer glitches during Virtual machines; (iii) VM Live migration failures, and (iv) problems in VM Live relocation. Incorporating these behaviours into the conceptual scheme might result in a state explosion, causing the modeling rewards calculation to take too long. As a result, we plan to address these in future employment using strategies such as interacting models. [17]

This work presents the Risk Score metric, which measures the system chance in a state that allows (increased risk) specific operation. Using the same unavailability framework, we can compute the RiskScore indicator for various consequences with our technique. Consider the following scenario to understand the recommended technique better. Consider using a Virtual Machine that can migrate [22]. The VM operates Based on qualitative, down when it is in the DW province, and migration when it is in the MG state. We ignore the possibility of VM failures throughout migrations in this case. Let us now imagine that such a problem is due to a data theft attack throughout the translation. As a result, the only stage that creates a risk to the public in this scenario is MG. As a result, the Risk Score is a metric based on the likelihood of the systems entering the MG condition [23]. The centralized controller may then alter the quantity (which is connected to migration regularity) to attain the appropriate Risk Score values. As a result, our sensitivity assessment technique quantifies protection levels based on the network state; ii) section reviews with unmodified availability - security assessment technique does not necessitate the availability variations. The analysis was based on the steady-state likelihood over a state, as we acquire security levels from the system's state (s). As a result, we may do vulnerability scans using the appropriate reward factors from the model. We want to emphasize that the Risk Score measurement is designed to assess the vulnerabilities relating to time invested in high-risk situations.

2.1. Reviews on Man-in-the-middle (MITM)

The perpetrator has access to this information relation between two interaction endpoints in MITM attacks [23]. The attacker uses it to modify data traffic or listen in on conversations. An attacker may be able to identify moving VMs on the networking with even adequate VM migration and high bandwidth confidentiality. An attacker with the relevant expertise to hijack the Routing path and conduct a harmful activity is considered. Since we use VM migration scheduling for VMM rejuvenation, numerous transitions may raise concerns about this attack. The preceding equation calculates the risk score of this study focus: Migration Probability = Risk Score. Migration Probability is the likelihood of a Virtual machine for the organization. The probability of token occurrence in areas of LM or DW migration is used to calculate Migration Probability [24].

2.2. Reviews on DoS

The primary threats to Cloud Technology dependability are DoS and DDoS attacks [25]. DoS attacks against the public Cloud often try to saturate internet bandwidth or degrade the performance of applications operating on virtual machines. Because of the flexibility of the Cloud, DoS attacks are considerably more deadly. As a result, when the Community cloud receives a heavy workload, it begins allocating extra funds (i.e., servers and

VMs) to accommodate the customer orders. As a result, the DoS attack may impact the total Cloud Technology capability by overwhelming just one of the systems. The technological implications of 'Cloud computing has been proposed and empirically evaluated.[26].Suppose the file is vital to the user. In that case, the file is partitioned into smaller sections and stored in separate virtual machines (VMs) and find a solution that would keep cloud service providers from accessing users' private data. It is thus possible to improve data dependability while respecting the privacy of user information[27].

3. Motivation

Most VM survivability works concentrate on restoring and protecting VM migration with reduced resource consumption. However, it eliminates the SLA constraints like an interruption of service and VM availability. For instance, various authors attain superior resource efficiency and a better acceptance ratio, leading to service interruption and VM migrations. Existing works consider the periodical VM synchronization part of the protection strategy. Therefore, the failed VM is recovered from the prior security state over the available server. The investigator considers the VM replication-based acknowledgment in various servers to fulfill the VM state loss after the traces of malicious activities.

Moreover, provisioning VM backup at the servers and constant replication of VM memory consumes massive transmission bandwidth and node resources and leads to high costs. The massive traces of malicious activities lead to immense damage over the wider area and huge VM service disruption and resource consumption. Thus, fulfilling constant service during malicious activities is a challenging task.

During the earlier warning time, certain reactive functionalities are adapted to survive the threatening activities from the disruption. When VM migration facilitates constant service provisioning during migration, it is executed to survive in an alive state. Therefore, VM migration is possible with the anticipated model before the termination of warning time and offers optimal outcomes. This work concentrated on modeling a reactive VM survivability model and anticipated the last warning time-based VM evaluation to combat the malicious activities. The anticipated model reconfigured the anticipated model before the threat occurrences, and the model adopts post-copy to migrate the active VM towards the virtualized environment. To enhance the efficacy of the evacuation model, the reconfiguration strategy is achieved along with resource optimization. This work performs a basic migration process and updates the bandwidth before the downfall. During migration, the target is to take advantage of earlier warnings to perform parallel migration and improve resource utilization. Therefore, specific metrics are computed to achieve a better completion time to enhance the completion ratio (evacuation) before the deadline.

4. Problem statement

This work focuses on modeling an efficient approach for handling the VM migration problem during threat cases like DoS attacks, MITM, and both attacks. The attacks are predicted with the large-scale environment, and an earlier warning is triggered to alert the VM migration process without any interruptions. These threats may also damage the entire cloud infrastructure and disrupt VM services. The proposed model intends to reconfigure the threatened VM and migrate the VM during the earlier warning time to prolong the service. Generally, some inactive VM migrations are suspended, and the active nodes are considered for live migration and reconfiguration. This work aims to perform as many migrations and reconfigurations within the given time, reducing the potential service disruptions.

5. Network model

Consider a set of PM and VM machines specified as $G^*\ (N^s, L^s)$ where N^s specifies the substrate nodes and L^s Specifies the link nodes. For all the substrate nodes $n^s(\in N^s)$, r_{n^s} specify available resources like memory, CPU, and storage resources. For all substrate link $l^s\ (\in L^s)$, b_{l^s} specifies the bandwidth capacity. The vulnerability G^s is specified as G_d^s Which considers substrate nodes and adjacent substrate links. Therefore, $\bar{G}_d^s(\bar{N}_d, \bar{L}_d^s) = G^s(N^s, L^s) - G_d^s(N_d^s, L_d^s)$. Here, VM is specified as $G^v(N^v, L^v)$, where N^v represents virtual nodes, and L^v specifies virtual links. The virtual node $N^v(\epsilon N^v)$ is offered with VM and demands node resources capacity r_{n^v} while virtual link $l^v\ (\in L^v)$ demands bandwidth capacity b_{l^v}. The vulnerability G^v specifies G_d^v includes virtual nodes and links. Therefore, $\bar{G}_d^v(\bar{N}_d^v, \bar{L}_d^v) = G^s(N^v, L^v) - G_d^v(N_d^v, L_d^v)$.

6. Methodology

To survive during the threatened environment (DoS, MITM, and both) and sustain during the service, we anticipate a novel approach to evacuate VM within the earlier warning time. Generally, VM is an NP-hard issue that needs to be resolved. Therefore, the anticipated model includes two diverse phases: VM reconfiguration and live migration.

6.1. VM reconfiguration

The anticipated model will re-map the vulnerability part connected to G_d^s into the nodes and link $\bar{G}_d^s$. Consider a threatened environment $VM\ G^v$. The model attains a suitable substrate node $n^s\ (\in \bar{N}_d^s)$ and re-maps the vulnerable nodes. Some constraints from Eq. (1) → (4) must be fulfilled. From Eq. (1), the model needs to guarantee the candidate node n^s has virtual node n^v With appropriate node resources. Eq. (2) to Eq. (4) needs to ensure the substrate node n^s Can host with virtual node from same VM and virtual node NV is mapped over the substrate node. It is expressed as in Eq. (1) to Eq. (5):

$$r_{n^v} \leq r_{n^s} \tag{1}$$

$$X_{n^v,n^v} = \begin{cases} 1 & if\ n^v\ is\ re-mapped\ with\ n^s(\in \bar{N}_d^s) \\ 0 & else \end{cases} \quad (2)$$

$$\sum_{n^v \in N^v} X_{n^v,n^s} \leq 1; \quad \forall n^s(\in \bar{N}_d^s) \quad (3)$$

$$\sum_{n^s \in \bar{G}_d \wedge s} X_{n^v,n^s} = 1 \quad (4)$$

$$b_{l^v} \leq b_{l^s} \quad (5)$$

Then, the model initiates re-mapping with all the virtual links connected to the vulnerable nodes. Every virtual link $l^v\ (\in L_d^v)$ needs to be re-mapped over the newly substrate path p_{l^v}, composed of successive substrate links $\{l^s\} \subset \bar{L}_d^s$. Here, p_{l^v} is used to compute the shortest path algorithm with bandwidth constraints and fulfills the substrate link l^s with path p_{l^v} and offer appropriate bandwidth for l^v. For resource optimization with bandwidth usage during VM reconfiguration, this work selects substrate nodes $\{n^s\}$ from $\bar{N}_d^s$ As destination candidates. It is utilized to offer solution space, h_{n^s} specifies the distance among n^s and related nodes $\bar{N}_d^v$ Where H specifies the constraint distance hopping. The following equations are adopted to evaluate the path length p_{l^v} is specified as $h_{p_{l^v}}$. For every candidate substrate node, the sum of virtual nodes' length is attacked, and the destination node predicts the minimal total length. Therefore, the VM reconfiguration is provided as in Eq. (6) to Eq. (9):

$$h_{n^s} \leq H \quad (6)$$

$$Y_{l^v,l^s} = \begin{cases} 1 & if\ l^v\ traverses\ l^s(\in \bar{L}_d^s) \\ 0 & else \end{cases} \quad (7)$$

$$h_{p_{l^v}} = \sum_{l^s \in \bar{G}_d^s} Y_{l^v,l^s} \quad (8)$$

$$\min \sum_{p_{l^v}} h_{p_{l^v}} \quad (9)$$

6.2. Live migration phase

It is known as the process of migrating VM from the anticipated model to reconfigured virtual node devoid of service disruption with three phases: 1) establish start and migration path; 2) upgrade migration bandwidth during VM downtime is fulfilled, and 3) release bandwidth and a migration path when migration is performed. The pre-copy approach is exploited to eliminate dirty page generation when the VM live migration transfers massive data. Therefore, we need to reduce the data to be optimized and transfer migration bandwidth evaluation. Consider. VM reconfiguration G^v and the anticipated model establishes the migration path for the vulnerable (DoS, MITM, and both) VM among the source and destination servers. To fulfill the life G^v migration is performed before the provided deadline, migration bandwidth specified as $b_{G^v}^o$ is measured with Eq. (10). Here, T specifies the last warning time, t_c specifies the current time, τ_g specifies maximal downtime, and D_{G^v} specifies the total storage and memory data to be transferred. Therefore, migration bandwidth is allocated for the path p_{G^v} and related to constraints over Eq. (11). In the end, time migration $t_{G^v}^e$ is predicted with Eq. (12), where $D_{G^v}^c$ specifies the data migrated in the current state and $D_{G^v} - D_{G^v}^c$ specifies the amount of data that needs to be migrated.

$$b_{G^v}^0 = \frac{D_{G^v}}{T - t_c - \tau_g} \quad (10)$$

$$b_{G^v}^c \geq b_{G^v}^0 \quad (11)$$

$$t_{G^v}^e = t_c + \frac{D_{G^v} \_ D_{G^v}^c}{b_{G^v}^c} \quad (12)$$

$$C_{l^s} = e^{-1/n} + \varepsilon \quad (13)$$

To enhance the migration completion ratio over the provided time, the anticipated model intends to conduct various VM migrations with $b_{G^v}^c$ at time. Assume CPU state with bit information and τ_g the migration influences downtime, $b_{G^v}^c$ needs to be initialized after VM downtime and improves the upper bound of the migration path and bandwidth. Thus, a cost function is designed for shortest path evaluation where c_{l^s} specifies the substrate link $l^s\ (\in L^s)n$ specifies the total migration path for traversing $l^s\ (\in L^s)$, $e^{-\frac{1}{n}}$ specifies the constraint over substrate link, ε specifies the decimal to zero. It is expressed as in Eq. (13):

$$S \rightarrow system\ state, S = 00\ is\ default\ status,$$
$$S = 01\ specifies\ G^v\ successfully\ reconfigured, S$$
$$= 10\ specifies\ the\ complete\ downtime, and\ S$$
$$= 1\ \ specifies\ successful\ migration.$$

When the migration is completed, the bandwidth is allocated to migration path pGv, which is utilized for newer VM reconfiguration and migrated bandwidth. Then, $t_{G^v}^d$ specifies the downtime moment, Q specifies the threatened VM, M specifies the VM under migration process, and C specifies the candidates for VM reconfiguration.

6.3. Heuristic model for VM migration

The anticipated model comprises two diverse phases: 1) traces the system status and reconfigures the attack-based threatened VM. In the initial phase, the nodes from the threatened VM predict the shortest path for the related virtual links. The nodes are connected with minimal virtual link length chosen, and the VM and the corresponding links are reconfigured to other available PMs. After the reconfiguration process, the successive step needs to be executed. The migration path for every VM is provided based on the sharing principle and migration bandwidth. Then, VMM is initiated. When the threatened

VM completes the downtime process, the bandwidth is upgraded with available bandwidth with the migration path.

At last, the VM is migrated completely, and allocated bandwidth is utilized and released to start the successive migrations. Based on these procedures, the anticipated model reduces the computational complexity. Based on $O(|Q|.|C|.D.nlog\ n)$. D specifies the maximal degree of a specific virtual node and $n\log n$ specifies the algorithm complexity.

Algorithm 1

1. Initialize the present state $S \rightarrow 00$ while identifying the traces of the VM $\{G^v\}$;
2. If $t_c < T$ move to step 3;
3. else
4. stop the process;
5. If $S \rightarrow 00$ move to step 4;
6. else
7. move to step 23;
8. if $M \neq \emptyset$, verify $\{G^v\}$ in M and move to step 12;
9. else
10. move to step
11. if $t_c = \min\{t_{G^v}^d\}$
12. specify $S \rightarrow 10$ move to 21;
13. else
14. Move to step
15. if $t_c = \min\{t_{G^v}{}^e\}$
16. specify $S \rightarrow 11$ and move to step 21;
17. else
18. move to step
19. If $Q \neq \emptyset$ for G^v in Q execute the successive steps;
20. Move to step 2;
21. For G^v and execute the next process
22. else
23. move to step 2.

//successive process

1. Choose the candidate node $\{C_i^s\}$ from $\bar{N}_d^s$ based on Eq. (1) → Eq. (4) by initializing $i = 0$;
2. if $i < |C|,\ i = i + 1$ and evaluate the shortest path for all $l^v\ (\in L_d^v)$ with certain constraints of Eq. (3)
3. Else
4. Move to step 9;
5. If the process is successful, move to step 2
6. else
7. Eliminate n_i^s from C;
8. Move to step 2;
9. if $C \neq \emptyset$ based on $Eq.(7) \rightarrow Eq.(9)$;
10. Choose the nodes with the minimal virtual link from C and performs mapping with the virtual node;
11. Demand the node resources;
12. Move to step 15;
13. Else
14. End queue with $S \rightarrow 00$ and move to step 15;
15. Perform mapping with $l^v\ (\in L_d^v)$ and related path with the required bandwidth b_{l^v};
16. Set $S = 01$;
17. Move to step 18;
18. Return S.

//Sub-phase process

1. If $S \rightarrow 01$; move to step 2;
2. else
3. Move to step

4. Compute migration path with essential migration bandwidth and move to step 5;
5. If the process is a success, $b_{G^v}^{c} = b_{G^v}^{0}$ move from Queue to Migration;
6. Initiate VM migration (live) and set $S \rightarrow 00$;
7. Move to step
8. If $S \rightarrow 10$ move to step 13;
9. else
10. move to step 15;
11. Upgrade the available bandwidth and mark $S \rightarrow 00$;
12. move to step 15;
13. Release the allocated bandwidth and remove prior migration, and set $S \rightarrow 00$;
14. move to step 15;
15. Return S;

7. Results and Discussions

Here, the outcomes attained with the VM evacuation model due to threat traces like DoS, MITM, and both attacks. The simulation outcomes are analyzed and evaluated to project the performance of the anticipated model. To project the significance of the anticipated model, two diverse approaches, baseline models and best-effort model, is adopted for comparison. The reconfiguration approach is applied to all these methods and may vary during the post-copy migration process. The provided baseline model used traditional transmission approaches during the initial migration process and measured the shortest path among the source and destination servers. Then, allocate available bandwidth and perform the migration.

Moreover, predicting the shortest path for migration and allocating bandwidth for migration intend to upgrade the model's performance. Consider a testing environment with 22 virtual links and 14 connected nodes and another with 43 virtual links and 24 nodes. Assume that every node with massive storage and computational ability. With the former testing environment, the bandwidth allocated for every link is 80 Gbps and for the successive environment is 150 Gbps. Consider substantial threat traces over the environment, and the probability of occurrence is 1. Before predicting threats, 100 and 450 VM requests are injected and provided to the testing environment (See Fig 1 and 2). For all VM, the number of available virtual nodes is produced randomly among the PMs, and every VM's data is distributed. For all virtual nodes, the link is produced with 0.5 probability and requires specific bandwidth that ranges from 1 to 3 Gbps. The VM migration downtime is produced randomly among the 1 sec and 2 sec. When there are some attack traces, the hypervisor needs to generate an alert with prior warning time, and the simulation is set for 10 to 50 seconds, respectively. There is some set of virtual requests among the provided testing environment with all these considerations. The outcomes are averaged, and the threat occurrence state is measured.

There are two diverse performance metrics in the anticipated model, i.e., relative VM completion ratio evacuation and average evacuation time. The former metric is the difference among the evacuated VM, and the latter model is the average evacuation time. Fig 3 and Fig 4 depict the last warning time generated over the provided testing environment. Based on the increased warning time, the average evacuation time is increased as the model is evacuated completely. The evacuation is ultimately a vast data to be migrated and possess narrow bandwidth migration and initiates the migration. The average evacuation time of the baseline approach is lesser than the proposed model. However, it is increased faster than the other methods with the rise in the earlier alert. It shows that the migration bandwidth allocated for the baseline models is lesser than the other two approaches, and it causes prolonged evacuation time, i.e., specifically for huge VM. The evacuation time of the anticipated model is higher than the other approaches. Because the proposed model targets executing VM evacuation before the deadline, it evaluates the preliminary migration bandwidth based on the parallel migration and earlier warning time for the provided time. Thus, the model limits the completion time to the deadline. Moreover, the evacuation of the existing model needs to be faster. Therefore, it seizes the existing bandwidth with the migration path, reduces the completion time of every VM, and reduces evacuation time.

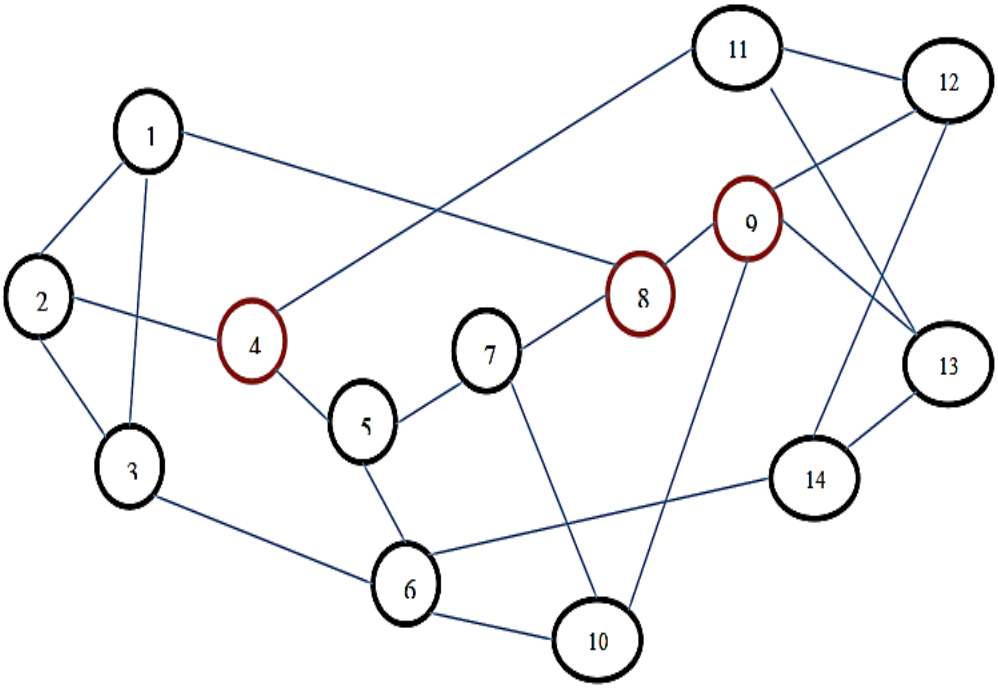

Fig. 1 Testing environment 1

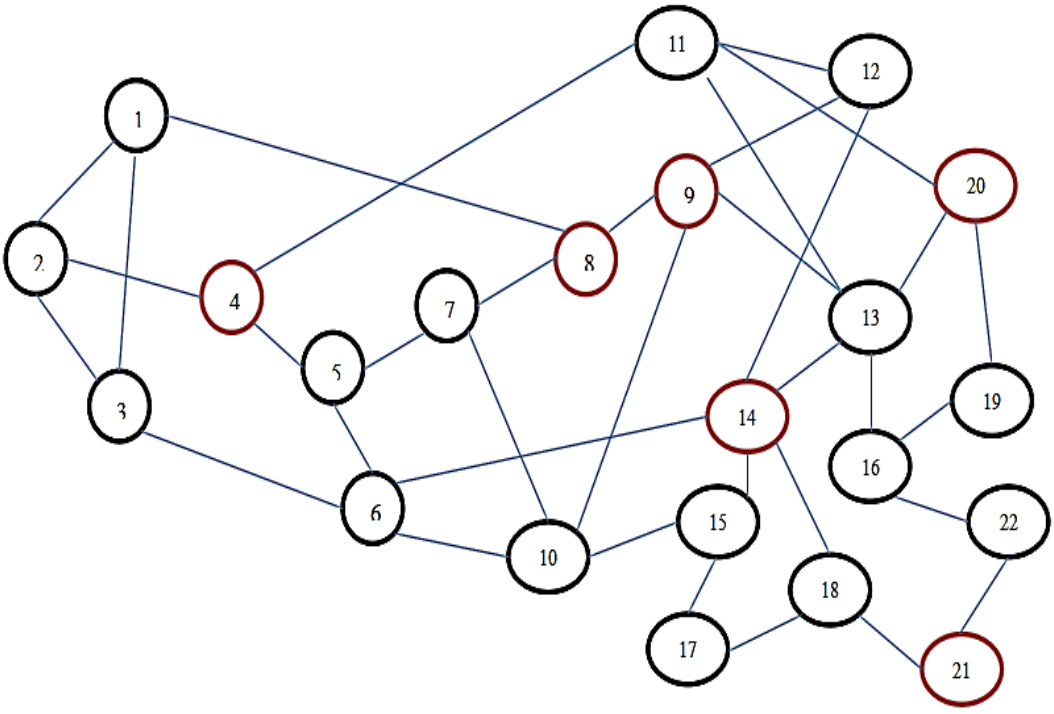

Fig. 2 Testing environment 2

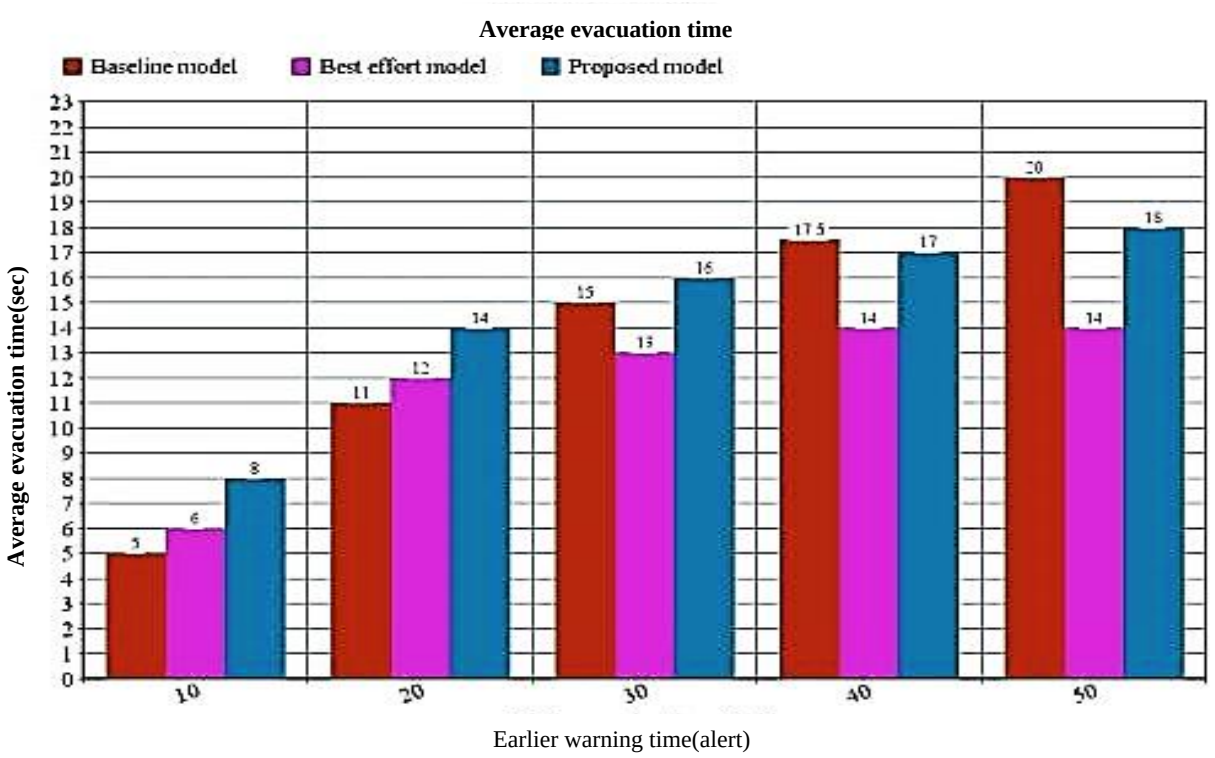

Fig. 3 Average evacuation time (testing environment 1)

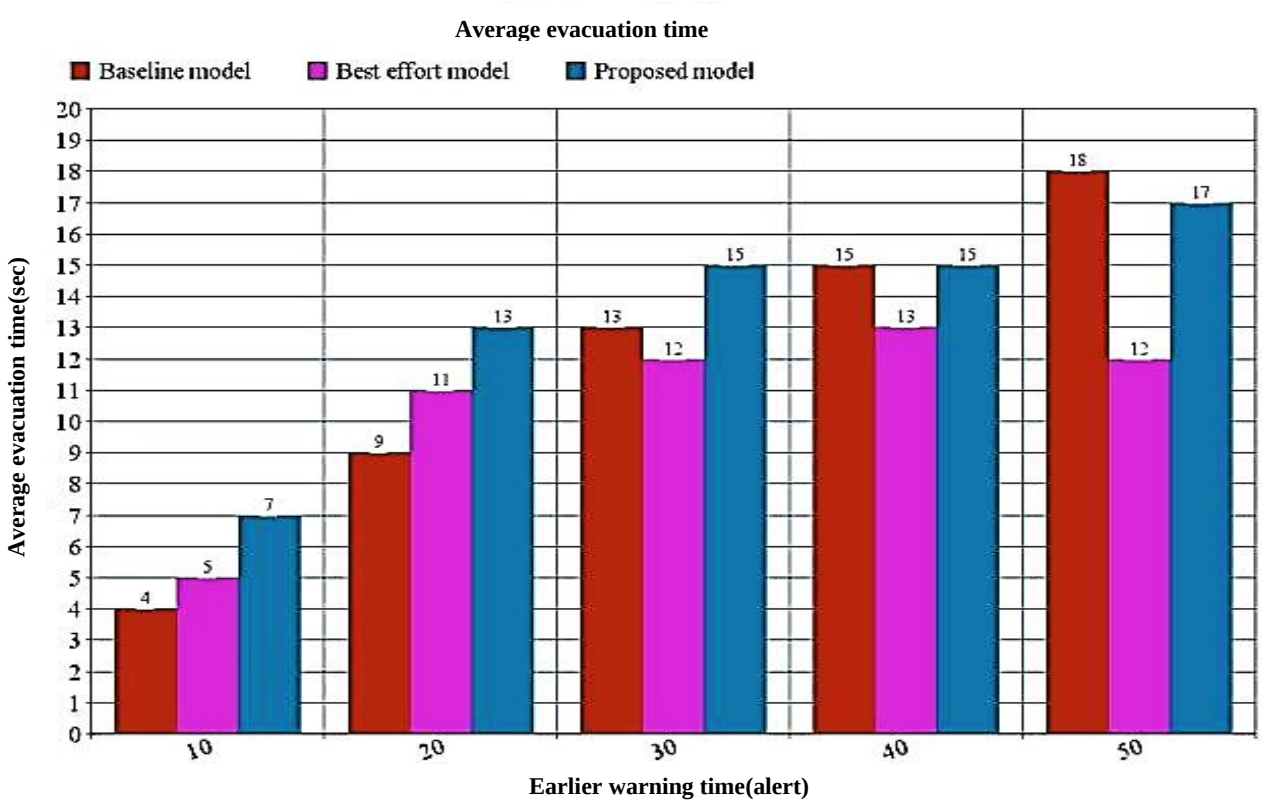

Fig. 4 Average evacuation time (testing environment 2)

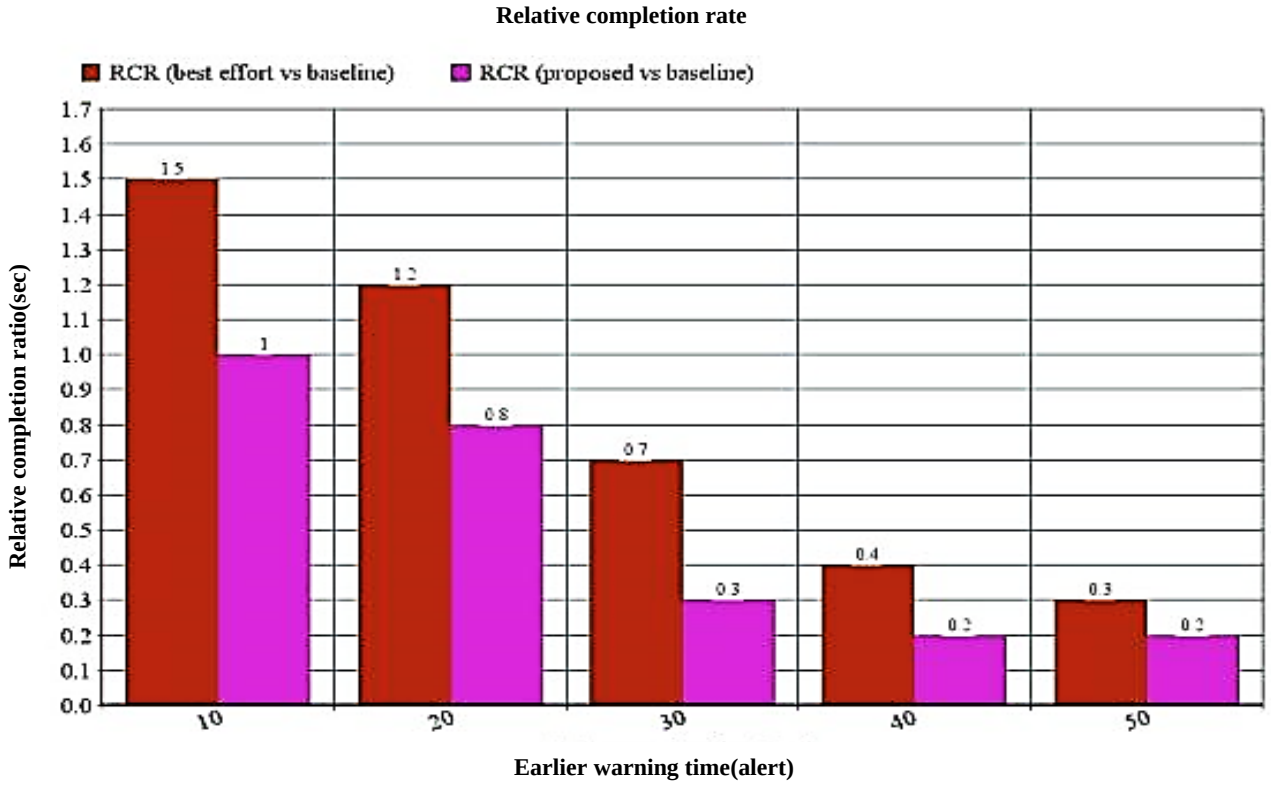

Fig. 5 Relative completion time (testing environment 1)

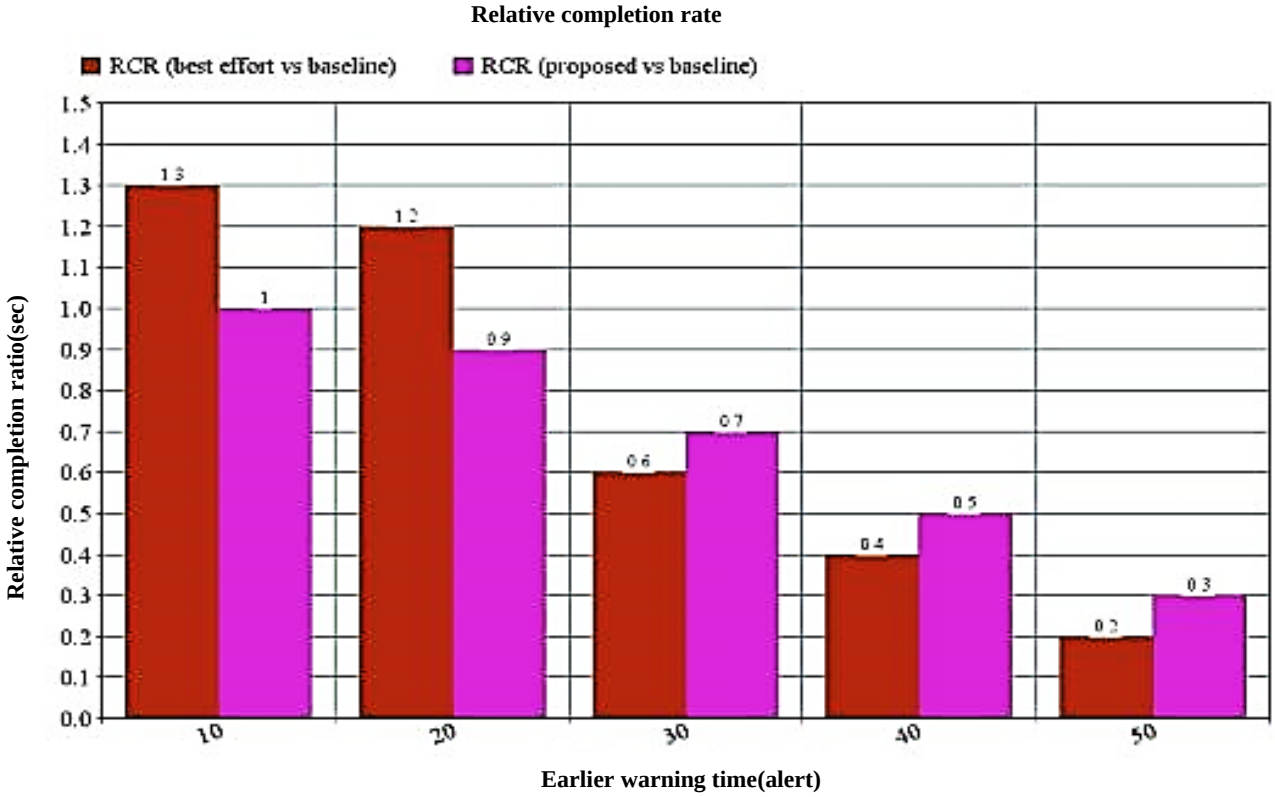

Fig. 6 Relative completion time (testing environment 2).

Fig 5 and Fig 6 depict the relative completion ratio of the anticipated model under the provided testing environment, which specifies that both models attain a higher evacuation completion ratio than the baseline approach. It includes two aspects. The model uses migration bandwidth to promote the parallel evacuation process and assists in taking the benefit of provided time. The other model intends to eliminate huge migration paths traversing the links, upgrade the migration bandwidth after the downtime, enhance the bandwidth utilization, reduce the evacuation completion time, and lead to deadline handling. Compared to other approaches, the baseline model intends to compress and seek the completion time for all VM.

Moreover, it does not determine the migration path and warning time. Compared to other approaches, earlier warning time should be lesser than 10 to 30 seconds. Sometimes, the migration opportunity is reduced, leading to VM evacuation before the deadline.

8. Conclusion

The dependability and trustworthiness of virtualized systems with VM evacuation enabled by VM migration management were evaluated in-depth in this article. Our findings give a tradeoff interpretation in three studies: 1) MITM, 2) DoS attacks and 3) a combination of both. Each real-world example includes a series of settings that include the ideal rejuvenating schedule and intermediate weight arrangements for security and protection, giving the decision-maker more knowledge. Our vulnerability assessment method is adaptable and can be used in various threat models and modelling contexts. The ultimate findings reveal that lowering the risk of a MITM attack is contradictory to reducing the level of DoS attacks since strategies that lower one tends to raise one another. Many study avenues might be pursued in the coming. We want to look at various attacks, such as infiltration attacks and particular DoS domains (server overload). The scope of the proposed approach is to include more significant system designs (multiple VMs). Aside from that, we want to continuously expand the scope of our framework by including new types of dependable scenarios. In the future development, specialized business models may be included in the modelling framework.

References

[1] Wei, H. Gu, K. Wang, X. Yu, and X. Liu, Improving Cloud-Based IoT Services Through Virtual Network Embedding in Elastic Optical Inter-DC Networks, IEEE Internet of Things Journal. 6(1) (2019) 986–996.

[2] Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, Virtual Network Embedding: A Survey, IEEE Communications Surveys & Tutorials. 15(4) (2013) 1888–1906.

[3] Habib, M. Tornatore, F. Dikbiyik, and B. Mukherjee, Disaster Survivability in Optical Communication Networks, Computer Communications. 36(6) (2013) 630–644.

[4] Guo, C. Qiao, J. Wang, H. Yu, Y. Zuo, J. Li, Z. Chen, and Y. He, Survivable Virtual Network Design and Embedding to Survive a Facility Node Failure, Journal of Lightwave Technology. 32(3) (2014) 483– 493.

[5] Jiang, L. Gong, and Z. W. Zuqing, Efficient Joint Approaches for Location-Constrained Survivable Virtual Network Embedding, IEEE Global Communications Conference, Austin, TX, USA. 8(12) (2014) 1810–1815.

[6] Gu, K. Shaban, N. Ghani, S. Khan, M. R. Naeini, M. M. Hayat, and C. Assi, Survivable Cloud Network Mapping for Disaster Recovery Support, IEEE Transactions on Computers. 64(8) (2015) 2353–2366.

[7] Couto, S. Secci, M. E. M. Campista, and L. H. M.K. Costa, Server Placement with Shared Backups for Disaster-Resilient Clouds, Computer Networks. 93(3) (2015) 423–434.

[8] Shahriar, R. Ahmed, S. R. Chowdhury, A. Khan, R. Boutaba, and J. Mitra, Generalized Recovery From Node Failure in Virtual Network Embedding, IEEE Transactions on Network and Service Management. 14(2) (2017) 261–274.

[9] Tsakalozos, V. Verroios, M. Roussopoulos, and A. Delis, Live VM Migration Under Time-Constraints in Share-Nothing IaaS-Clouds, IEEE Transactions on Parallel and Distributed Systems. 28(8) (2017) 2285–2298.

[10] Khan, N. Shahriar, R. Ahmed, and R. Boutaba, Multi-Path Link Embedding for Survivability in Virtual Networks, IEEE Transactions on Network and Service Management. 13(2) (2016) 253-266.

[11] Beloglazov A, Abawajy J, Buyya R, Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centres for Cloud Computing, Future Gener, Comput. Syst. 28(5) (2012) 755–768.

[12] Stergiou C, Psannis K.E, Kim B.G, Gupta B, Secure integration of IoT and Cloud Computing, Future Gener, Comput Syst. 78 (2018) 964–975.

[13] Gupta B.B, Gupta S, Chaudhary P, Enhancing the Browser-Side Context-Aware Sanitization of Suspicious HTML5 Code for Halting the DOM-based XSS Vulnerabilities in the Cloud Int. J. Cloud Appl. Comput. IJCAC. 7(1) (2017) 1–31.

[14] Xue W, Li W, Qi H, Li K, Tao X, Ji X, Communicationaware Virtual Machine Migration in Cloud Data Centres, Int. J. High Perform. Comput. Netw. 10(4–5) (2017) 372–380.

[15] Hu B, Lei Z, Lei Y, Xu D, Li J, A Time-Series Based Pre-Copy Approach for Live Migration of Virtual Machines, In: IEEE 17th International Conference on Parallel and Distributed Systems, ICPADS, IEEE. (2011) 947–952.

[16] Bauchi A, Toshimi Midorikawa E, Netto M, Improving Virtual Machine Live Migration via Application-Level Workload Analysis, In: 2014 10th International Conference on Network and Service Management CNSM, IEEE. (2014) 163–168.

[17] Zhang X, Shae Z.Y, Zheng S, Jamjoom H, Virtual Machine Migration in an Over-Committed Cloud, In: Network Operations and Management Symposium NOMS, IEEE. (2012) 196–203.

[18] Ghavipour M, Meybodi M.R, An Adaptive Fuzzy Recommender System Based on Learning Automata, Electron. Commer. Res. Appl. 20 (2016) 105–115.

[19] Beloglazov A, Buyya R, Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centres, Concurr. Computat. Pract. Exp. 24(13) (2012) 1397–1420.

[20] Beloglazov A, Buyya R, Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centres, Concurr Comput. Prac Exper. 13 (2012) 1397–1420.
[21] Ding W, Luo F, Han L, Gu C, Lu H, Fuentes J, Adaptive Virtual Machine Consolidation Framework Based on the Performance-To-Power Ratio in Cloud Data Centres,Future Gener Computer Syst. 111 (2020) 254–270
[22] Zahedi Fard SY, Ahmadi MR, Adabi S, A Dynamic VM Consolidation Technique for Qos and Energy Consumption in the Cloud Environment, J Supercomput .73 (2017) 4347–4368.
[23] Mohiuddin I, Almogren A, Workload Aware VM Consolidation Method in Edge/Cloud Computing for Iot Applications, J Parallel Distributed Comput. 123 (2019) 204–214.
[24] Wan J, Chen B, Wang S, Xia M, Li D, Liu C, Fog Computing for Energy-Aware Load Balancing and Scheduling in Intelligent Factory, IEEE Trans Indus Inf. 14(10) (2018) 4548–4556.
[25] Ilager S, Kotagiri R, Rajkumar B, Thermal Prediction for Efficient Energy Management of Clouds Using Machine Learning, IEEE Trans Parallel Distrib Syst TPDS. 32 (2020) 1044–1056
[26] R. Santhana Lakshmi1, Factors Influencing SMEs towards Execution of Technology Adoption Model in Cloud Computing.International Journal of Engineering Trends and Technology. 69(3) (2022) 189-194.
[27] V. Gokula Krishnan , J. Deepa , S. Venkata Lakshmi,Securing Mass Distributed Big Data Storage using Intelligent Elliptic Curve Integrated Encryption Scheme in Multi-Cloud Computing, International Journal of Engineering Trends and Technology. 70(1) (2022) 35-42.