AN OPTIMIZABLE CONDITIONAL RANDOM FIELD-BASED MIGRATION STRATEGY FOR VIRTUAL MACHINES TO IMPROVE THE SECURITY AND PRIVACY ISSUES IN CLOUD COMPUTING

Nelli Chandrakala[1]
[1]Research Scholar, Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India.
[1]Email:: klala5136@gmail.com

Vamsidhar Enireddy[2]
[2]Professor,Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India.
Email:: vamsidhar@kluniversity.in

Abstract

Virtualization creates virtual OS, platform, network devices, storage, software, and hardware devices in cloud computing. Moreover, Virtual Machine (VM) technology has essential building blocks like cluster systems and data centre. The advancement is due to migrating, consolidating and isolating workloads. The VM migration seeks to enhance the security, performance, manageability and fault tolerance systems. In a virtual CC environment, some sets of tasks from various users are scheduled over the VMs, and load balancing turns out to be a crucial issue in achieving security and energy efficiency. Therefore, a novel optimization algorithm is initiated to resolve these issues and attain superior balancing with the influence of external resources. The Conditional Random Field-based Moth Algorithm (CRF-MA) considers the multi-objective functions by handling metrics like security, energy consumption, CPU utilization, makespan, migration, and resource cost. The performance of the CRF-MA is examined by determining the energy consumption, SLA violation, solution size and migration number. The simulation is done in CloudSim, and the proposed CRF-MA gives a better trade-off than other approaches.

Keywords- VM migration,security, cloud, optimization, multi-objective constraints, migration number

1. Introduction

Cloud computing and artificial intelligence enable direct centre facilities to be somehow accessible utilizing virtual servers, and it is used to solve a variety of issues [2]. Virtual machines (VMs) in the virtualized setting employ virtual network routing to migrate from one host to another in the little done to resolve, improving host speed, management, and high availability [1]. Live migrants have used computation dynamically to re-allocate VMs depending on unique massive resource consumption and move unoccupied hosting to reduce the power consumption to reduce the number of devices in the network and power consumption [3]. Other important considerations to improve translation include avoiding prospective Risks as much as possible and lowering movement duration. Multiple studies have overcome various VM placement issues in cloud servers [4]. In general, there seem to be two techniques to reduce implementation complexity: lowering the overall number of migrants and minimizing implementation complexity. Comprehensive migration number minimization approaches decrease the number of migrations required, lowering power consumption [5]. Technologies for reducing

implementation complexity, on the other hand, aim to enhance the productivity of virtualization to reduce the overall virtual machine period [6]. While numerous researches have concentrated on lowering implementation complexity, only a little has improved the overall number of migrations. Furthermore, most previous studies have already ignored the effects of deployment and substitute minimization on the likelihood of VM instances or, although addressing the problem, have suggested no effective solutions [7].

To tackle the difficulties above, avoid needless migration and lower Routing possibilities, the author presents an adaptive strategy, named CLANFIC, based on neuro-fuzzy modelling and improved CLA-EC to minimize Live migration possibility in cloud computing environment [8]. The Takagi– Sugeno model predictive control makes the neuro-fuzzy projection. CLANFIC is analyzed utilizing the CPU usage data set obtained in the PlanetLab project; likewise, the SPEC power benchmark 2 (the first industry-standard benchmark that assesses the power and performance aspects of a dedicated processor and parallel computing servers) consumption model has been applied [9]. PlanetLab is a free software environment for building, delivering and using planetary-scale solutions. A result attained is a worldwide research community that develops innovative communication networks [10]. Over 1000 investigators have utilized PlanetLab from leading significant universities and commercial research facilities to create new technologies for distributed storage, network mapping, peer-to-peer networking, decentralized codes, and information retrieval since its inception in 2003. PlanetLab now has 1353 nodes spread over 717 locations. The response complexity and migration frequency components compared of the suggested algorithm with First Fit Decreasing (FFD) and the research framework by Buyya et al. [11] revealed the successful optimization of the routing quantity as well as area and power usage [12] – [13]. But, there is a lack in measuring the migration time average mean of energy consumption. Thus, these issues can be resolved using mathematical analysis and optimization. Furthermore, the suggested strategy needs to have the potential to lower the rate of SLA violations. The following are the significant research contributions:

(1) To formulate the issues of VM placement relies on the Conditional Random Field (CRF) and the improved Moth optimization approach;

(2) The work is to diminish the number of VM migrations along with SLA violation rate and energy utilization;

(3) The advantage is to produce the future resource demands and eliminate the unavoidable migration using the prediction of the further demand.

The work is organized as: Section 2 elaborates various existing approaches; Section 3 offers an extensive analysis of the anticipated methodology. Section 4 determines the numerical outcomes of the anticipated model. Section 5 provides the summary of the research.

2. Related Work

The VM migration problem can be defined as an initial translation of VMs to guests and aims to discover a final optimum matching with the least VM mobility depending on the power usage and network infrastructure SLA violators. Different studies have been presented in the literature to reduce the frequency of VM migrations. Author et al. [13] used Remote Direct Memory Access to continue improving computation offloading performance and help the objective of minimizing relocation time and increasing channel capacity, preventing TCP/IP stack higher level compared by enabling I/O devices to securely connect recent memories, and decrease the impact on the desired locations throughout transmitting data. Using the step by step approach, Salfner et al. [14] more precisely detected high filthy pages in the previous and planning point of view and transmitted them in last year's iterate, reducing needlessly recurring distribution of polluted webpages. Nosy et al. [15] used parameter estimation to construct a design methodology that predicted implementation complexity and guided resource planning processes. They demonstrate that the degree of option available for the virtual machine significantly influences cell adhesion and proliferation.

Cerroni al. [16] presented the Magical optimized migrations paradigm, which uses stacked duplicate and recollection decompression methods to decrease migrations wastage and increase migrating efficiency by enhancing authentic translation time and space complexities. Jo et al. [17] introduced the reply approach for

References	Approach	Resources	Target
Aldossary et al.,	LP-relaxation model	Memory and CPU	Reducing server cost with constraint server capacity
Rybina et al.,	Constraint programming model	Storage, RAM and CPU utilization	Energy efficiency optimization
Nosy et al.,	Constraint programming	Bandwidth, RAM and CPU	Cost function optimization for VM provisioning and reducing the number of PM during packing
Chen et al.,	Heuristic approach	Bandwidth, memory and CPU	They reduce SLA violations and energy consumed during the under loaded and overloaded hosts to select VMs for migration and best-fit decreasing heuristics.
Cerroni et al.,	Heuristic approach	Bandwidth, memory and CPU	Predicting overloaded host with gradient-descent for correlation and regression and bandwidth-based VM selection
Cerroni et al.,	Heuristic approach	Bandwidth and CPU	Reducing power consumption with the first-fit approach
Cerroni et al.,	Heuristic with Bernoulli	CPU	Reduces the number of active nodes and energy costs with the prediction of CPU utilization
Jo et al.,	Heuristics for the stochastic bin packing process	RAM and CPU	Consolidate the migrations and assignments based on success probabilities
Huang et al.,	Heuristic algorithm	CPU	Reduces the number of active hosts and energy cost by analyzing CPU utilization
Bezerra et al.,	Heuristic algorithm with hierarchical clustering process	Network interface, storage, memory and CPU	Reduces migration, link utilization and energy using clustering model of minimum-cut
Melhem et al.,	Heuristic approach	Bandwidth, storage, RAM and CPU	Reduces energy consumption with a heuristic model
Bashar et al.,	Heuristic approach	RAM and CPU	Reduces power consumption and improves resource utilization with multi-dimensional model utilization
Shi et al.,	Heuristic approach	RAM and CPU	Reduces energy consumption with multi-dimensional partitioning

Table 1. Comparison of VM migration strategies

virtual machines to shorten the overall implementation complexity. A continuous personality technique is implemented to stop the flow of available memory pages. This solution minimizes the redundant sending rate of pre-copy by using passive shoving. It allows for its use of comment with adaptability to reduce the infrastructure

article. To mitigate overall virtual machine time, Huang et al. [18] created anan application-aware virtual machine framework to decide the optimal time to undertake a migration based on studying data received in combination with standard dynamic operating.

Cerroni et al. [19] adopted the well before-based VM migration methodology to integrate the advantages, including both strategies, combining the Time Series-based methodology with the three different methods for live migration. Although the suggested method reduces implementation complexity, it may result in a bit of increase in availability when contrasted to the usual pre-copy strategy. Bezerra et al. [20] presented a novel content movement approach based on multi-content duplication and fractional compressed to lower overall implementation complexity, migrations efficiency, total sample communication, and VM results of operations. Melhem et al. [21] introduced the WS Clock unit due to the placement in combination with a type before a strategy to decrease the amount of the transmitted recollection area and overall migrations duration. They also looked at the factors affecting the success of route discovery.Melhem et al. [22] cut the overall relocation rate in half by transmitting just a critical fraction of memory locations across the networking that wasn't on disc and downloading the deleted webpages in the meantime. At the same time, the VM is restarted on the ideal habitat. Based on indicators such as an outage, total migratory moment, and workflow achievement overhead cost, Bashar et al. [23] investigated the effect of various channel access processes on the achievement of route discovery in both reference and focus machines and analyzed the utilization of parallel migration project and amount of homework relocation strategic planning. According to Bezerra et al. [24], the bandwidth use behaviour inside the virtualized environment has the most significant impact on total cell adhesion, proliferation, and disruption.

Shi et al. [25] suggested dynamical aggregation using a migrating control strategy traditional regression computer framework and heuristics to regulate deployment, prioritizing virtualization with stable performance to decrease the number of migrations with the minor importance of body infrastructure penalties. The threshold-based dynamically consolidating strategy, introduced by Katzburg et al. [26], is used for migratory particularly likely to occur to reduce the number of frequent migrations. Alouri et al. [27] proposed a power, strength congestion control technique based on adaptive virtualization that considers RAM and speed and CPU use to improve the performance and congestion control. They implemented a limited migrating time regulation to limit the number of transfers and power consumption. Ivanov et al. [28] developed the dispersed online computation offloading method, which reduces the danger of congestion, reduces the number of required movements, and slightly affects VM routing overhead.Varaee et al. [29] presented the network interface, which uses the computational problems to forecast future network demand and move virtual instances from being over to the under servers. The number of migrations is reduced with effective load balance. Sikandar et al. [30] proposed a load balancing strategy for virtual machines that employ bottom and top thresholds to reduce migratory and electricity demand while increasing bandwidth utilization. The author used Mixed Integer Linear Programming to design the computation offloading issue, redistributing the VMs using the SVM Classifier technique to reduce the VM placement probabilities. The author introduced a heuristic-based proposed routing approach that incorporates the hardware server's consumption of resources by running the VMs that will maintain the movement rate and accuracy of conversions to the lowest.

3. Methodology

This section is partitioned into three diverse sections: 1) preliminaries of VM migration strategies; 2) CRF for VM migration; and 3) Moth algorithm for migration analysis. These sections provide a detailed analysis of the anticipated model to examine the migration of VMs concurrently.

3.1. Preliminaries

The process initiates with the definition of notations as follows: Consider $n_{iteration}$ specifies the total amount of successive iteration, $T_{i,j}$ determines the total time consumed by the jth iteration during migration process from i^{th} VM memory. The VM's memory to be transmitted is specified as V_{mem}. Let V^{th} determine the memory threshold to stop the copy when the threshold is not up to the level. Assume $r = \frac{(pg*d)}{R}$ specifies the dirty rate of memory that needs to be transferred. Here, d specifies the dirty page rate (pages/sec), pg specifies the memory page size, and R sets the networks transmission rate. The migration (pre-copy) is performed for successive rounds, and the total migration time ($i^{th}\ VM\ migration$) is expressed as in Eq. (1).

$$T_{i,migration} = \sum_{j=1}^{n_{iteration}+1} T_{i,j} \tag{1}$$

The total numbers of iterations $n_{iterations}$ are evaluated using Eq. (2).

$$n_{iterations} = \min([\log_r(vth/V_{memory\ size})]\ ;\ n_{maximal;}\ \ 0 < r < 1 \tag{2}$$

Here, $n_{maximal}$ specifies the maximal number of rounds/copies. The migration time for serial migration, which is shown in Fig 1(a). is expressed as in Eq. (3).

$$T_{migration}^{serial} = \sum_{i=1}^{m} T_i, migration \tag{3}$$

The download time for serial migration is evaluated using Eq. (4).

$$T_{download}^{serial} = \frac{V_{memory\ size}}{R} r^{no.of\ pre-copy\ rounds} + (m-1)\ \frac{V_{memory\ size}}{R} + T_{resume} \tag{4}$$

The migration time for parallel migration, is as shown in Fig.1(b) is expressed in Eq. (5).

$$T_{migration}^{parallel} = \sum_{i=1}^{m} T_i, migration \tag{5}$$

The download time for serial migration is evaluated using Eq. (6).

$$T_{download}^{parallel} = \frac{mV_{memory\ size}}{R} r^{no.of\ precopy\ rounds} + T_{resume} \tag{6}$$

Here, T_{resume} specifies the time taken by the VM for resuming towards the destination. The major issues in VM migration (serial and parallel) arises with the lack of compatibility among the servers and hardware's where the unnecessary presence of dedicated hardware, inappropriate resources and network access leads to disrupt in migration process. It can be resolved using the proposed conditional random field based VM migration process.

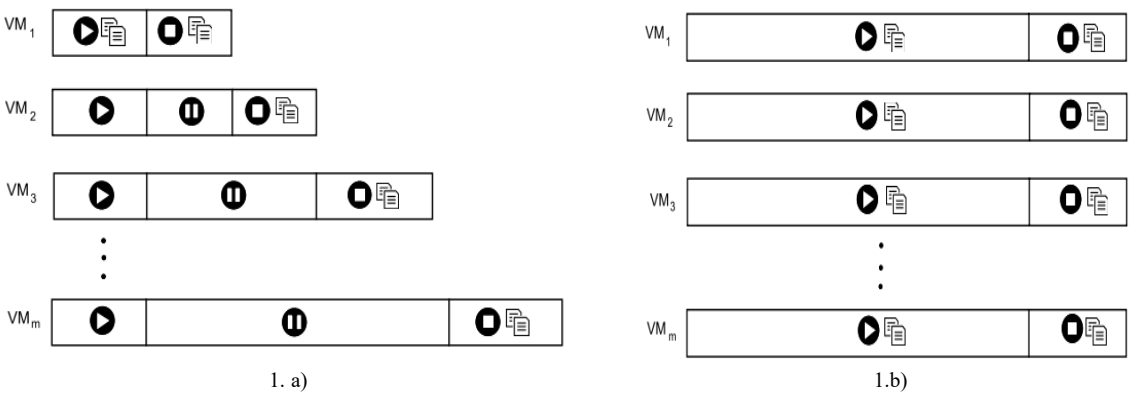

Fig.1.a) Serial migration b) Parallel migration

3.2. Conditional Random Field for VM migration

Conditional random fields (CRFs) are a sort of undirected Bayesian network separated into two distinct sets, the observations O and states S, instead of DBNs' directed knowledge graph. Because O does not need to be represented in the same task as S, DBN is a model of the likelihood function $P[O, S]$. Still, a CRF is a simulation of the conditional probability distribution. As a result, only S is considered to be categorized by the endpoints of a digraph $G = (V, E)$ in a CRF is conditional on observations O universally. The findings' rich and comprehensive characteristics are utilized, and interconnections between the inferences would not need to be overtly expressed. The directed graphical model of DBNs is identified as a vector representation where the observations O cannot be a parent of a phase and be regarded as probability created by the states S. On the contrary, the undirected graphical model, known as discrimination model $P[O]$, is not needed.

A CRF is defined as $P[s|o]$ if the probability $P[s|o]$ factorizes as shown in a material directed graph of S for any given o. The contingent allocation $P[s|o]$ has the form if $S's$ factor graph $G = (V, E)$ is a network or a branch. It is expressed as in Eq. (7).

$$P[s|o] = \frac{1}{Z(o)} \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, s_e, o) + \sum_{v \in V,k'} \mu_{k'} g_{k'}(v, s_v, o) \right) \tag{7}$$

Here, s_e and s o are the element sets of s corresponding with the endpoints of edge e and vertices v, accordingly; f_k and $g_{k'}$ are provided and preprocessing step functions, and $Z(o)$ is a specific example normalization kernel function using Eq. (8).

$$Z(o) = \sum_{s} \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, s_e, o) + \sum_{v \in V,k'} \mu_{k'} g_{k'}(v, s_v, o) \right) \tag{8}$$

A CRF with specified highlight variables can represent an HMM. CRFs expand the HMM by incorporating an unlimited number of image parameters to examine the whole observing chain. A statistical understanding is not required for the feature functions f_k and $g_{k'}$. When graph G is a chain or a tree, forward-backwards techniques and the Viterbi approach for HMMs can be employed to produce precise inference in a CRF. The accurate interpretation in a CRF is challenging for a generic network; however, methods like loopy believe dissemination can be employed to get accepted. This minimization is continuous for the CRFs where all points have logarithmic family dispersion and monitor all devices in the network connectivity. To overcome the issue, conjugate gradient procedures and Quasi-Newton techniques are utilized. If some conditions are not detected, they are extrapolated. It leads to the rise of an optimization approach that targets to provide an optimal solution in terms of resource utilization and measure the migration of VM over the host in an intelligent manner. This research concentrates on modelling and improved moth optimization to attain the optimal solution with VM migration.

4. Moth Algorithm

The moth–flame optimization (MFO) is a popular SI method inspired by moths' spiral migration for dusk illumination. This behaviour is based on moths' guidance system, which allows them to fly considerable distances in a single direction while keeping a constant inclination to the moon. If the light source is somewhat close to the moths, this fundamental navigation mechanism transforms into a fatal spiral route towards the beam of light. Based on the review, the MFO algorithm is made up of insects and flames. Moths are regarded searching agents, arranged in matrix M (t), that explore the D −dimensional subspace, as illustrated in Eq. (9), where N is the number of insects.

$$M(t) = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,D} \\ \vdots & \vdots & \vdots \\ m_{N,1} & m_{N,2} & m_{N,D} \end{bmatrix} \tag{9}$$

In particular, as illustrated following, the suitability of the respective caterpillar is kept in an array $OM\ (t)$. It is expressed as in Eq. (10):

$$OM(t) = \begin{bmatrix} OM_1(t) \\ OM_2(t) \\ \vdots \\ OM_N(t) \end{bmatrix} \tag{10}$$

Subsequently, flames are the superior positions identified by moths and stored in similar matrix $F(t)$, along with their two parameters in an array $OF(t)$. As shown in Eq. (10), the creatures turn all over their commensurate blaze, where $M_i(t)$ is the location of the ith moth in the current iterations, Dis_i ultimately decides the separation with both M_i and its corresponding j^{th} flame (F_j) developed, b indicates the structure of the fractal dimension curve. K is an actual random value among both periods $[1, 1]$. It is expressed as in Eq. (11).

$$M_i(t) = Dis_i\ (t) * e^{bk} * \cos\ (2\pi k) + F_j(t) \tag{11}$$
$$Dis_i\ (t) = |F_j(t) - M_i(t)|$$

The number of flames reduces in the number of evolution to convergence the system and give more exploiting, as shown in Eq. (11), where t denotes the actual time complexity. At the same time, N and Max_{It} reflect the total count of flaming and maximal amount of iterations, accordingly. It is expressed as in Eq. (12).

$$Flame_{num}\ (t) = round\ \left(N - t * \frac{N-1}{MaxIt}\right) \tag{12}$$

5. Moth-Conditional Random Field

The MFO is a well-known population-based methodology that has found success in various fields. According to the analysis and similar works, meanwhile, the MFO method suffered inadequate discovery and quick loss of genetic variation. The technique delivers more localized searching throughout the rounds as the hot spots coincide. As a result, due to the application's limited essential generate of moths surrounding their matching fires, it is vulnerable to sinking into the optimal solutions. As a result, this research offers a migration-based moth–flame optimization method, a mix of the MFO algorithm and the GA's overlap operator is shown in Fig 2. In addition, the model employs a guiding archive to preserve population variety, and a process model that uses the method to improve explores capabilities. By utilizing a modified GA's crossovers, the migration approach incorporates two variables, RM and GM. During initial tests, the RM operator is added to give adequate detection performance, while the GM operator corresponds the community towards attractive locations. Furthermore, as mentioned in Definition 1, a guide record is added to keep lucky moths that have progressed utilizing the migratory technique to sustain colony variety.

Definition (guiding archive): The guidance repository maintains the location of lucky moths, which is enhanced by the migration strategy to preserve species variety and prevent early colony merging. Whereas both RM and GM contribute enhanced moths to the guidance record, only the GM operator uses it. D and N's directing archiving potential ($MaxArc$) are dimensional and crowd size is shown in Eq.(13).

$$MaxArc = D * [lnN] \tag{13}$$

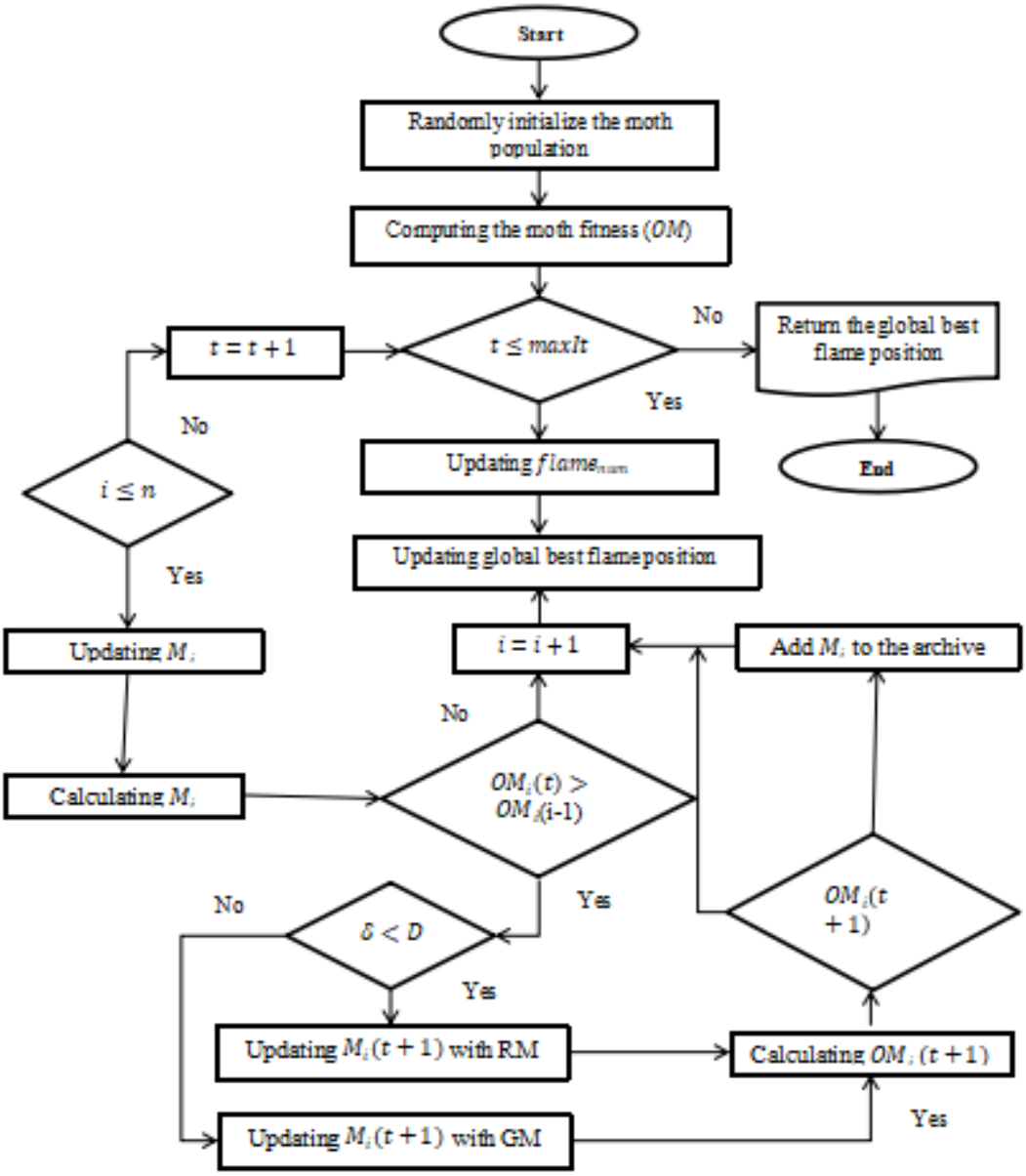

Fig 2 Flow diagram of Moth algorithm

The present dimension of the advising database (PlanetLab) must be more significant than the size of the issue elements to guarantee that it is developed enough to guide additional unfortunate moths (D). This constraint allows a dimensional aware switch to select the appropriate operative in the migration activity. The quantity of reaches $MaxArc$, the next moth, is swapped with a membership of the guidance archive at the chance. RM and GM operators are included in the migration activity to guarantee sufficient exploratory capabilities and convergent near potential zones. The RM implementation decisions for further exploring by pseudo-random shifting M_i location. Simultaneously, the GM operative is added to focus the demographic into potential zones by utilizing better moths from the leading repository. Furthermore, as shown in Eq. (14), the migratory method advantages from a dimensional space aware transition between this controller, where are the exact value of the guided collection. Accordingly, the M-pseudo-code MFO's and diagrams are shown in Algorithm 1 and Fig. 2.

$$M_i(t+1) = \begin{cases} RM\ operator & \delta < D \\ GM\ operator & \delta \geq D \end{cases} \tag{14}$$

Random migration (RM) operator: Let unlucky moths $(t) = \{M_1, M_2, \ldots, M_i, \ldots\}$, which is a finite set of unlucky moths, such that $OMi\ (t)\ > OMi\ (t\ -\ 1)$. As a result, the positioning of M_i in this function change when a freely

produced moth (M_r) is considered, and M_i symbolizes the parental in the interphase stated in Eq. (15) and Eq. (16), where α is a sequence of numbers in the range $[0, 1]$.

The crossover generates two offspring, and the fitter of the two is chosen and matched to the other children to determine which is the fittest. The position of $OM_i\ (t + 1)$ is introduced to the leading repository if it predominates the $OM_i(t)$. In the kth iteration, the RM operation serves the demand for exploring by pseudo-random shifting the unfortunate moths to explore attractive regions.

$$offspring_1 = \propto * M_i + (1-\propto) * M_r \tag{15}$$
$$offspring_2 = \propto * M_r + (1-\propto) * M_i \tag{16}$$

Guided migration (GM): The operator can vary the unlucky moth position, M_i when the GM size reaches the variable size. The GM changes the M_i position using the cross over-determined using Eq. (14), where M_r is a random moth (lucky) from the guiding archives. Alike of RM operator, when the new offspring attains the superior position compared to $M_i(t)$, position ($M_i\ (t + 1)$ is revised and provided to the guiding archives. It is expressed as in Eq. (17) and Eq. (18).

$$offspring_1 = \propto * M_i + (1-\propto) * M_r \tag{17}$$
$$offspring_2 = \propto * LM_r + (1-\propto) * M_i \tag{18}$$

The convergence qualities of the moth optimization are investigated in this experimentation collection, and the findings are analyzed to candidate methods for sizes 30, 50, and 100. Each approach achieves the divergence contours of the maximum two parameters on bimodal distribution, heterogeneous, hybrid, and composed validation data. The convergence behaviour of strategies is shown in the first row. In the early iterations, the optimization found the globally optimal for all variables, demonstrating moth-flame capacity exploitation. Other algorithms' development tendencies, on the other hand, were limited by local minima or showed a slow speed of convergence. Due to its exploration capabilities resulting from migrating method, the model gave the best results among rivals for bidirectional functionality in the early versions. The merging of the hybridization processes is seen in the third and fourth rows. By achieving a balance between work and personal, the model avoids the local minima and continues its slow march toward near-optimal solutions until about the ultimate repetitions. In the initialization step, the model acquired the right plan among rivals, as shown by the composition feature divergence curves in the last row. To summarize, the graphs show that the model outperforms its other algorithm capable of utilizing, discovering, and balancing such two inclinations. It's also worth noting that the model produced more similar results by up the responsibilities of the tissue elements.

Algorithm 1: Moth optimization

Input: Maximal iterations, no. of moths, dimension and maximal size of guiding archive.
Output: Finest optimal position and fitness value.
Begin
Random distribution of search space and moth;
Compute fitness (OM);
Set$t$ and $\delta \rightarrow 1/\delta$ specifies the archive members;
$OF(t) \rightarrow sort\ OM\ (t)$;
$F(t) \rightarrow sort\ M\ (t)$;
While $t \leq MaxIt$
Update the finest moth from the current position;
Update $Flame_{num}$;
For $i \rightarrow 1{:}N$
Update position and compute OM;
If $OM_i(t) > OM_i(t-1)$
$\tau \rightarrow generate\ random\ number\ among\ the\ intervals\ [1, D]$;
For $j = 1{:}\tau$
If $\delta \leq D$ //guiding archives
Generate successive positions $M_i(t + 1)$ using RM operator;
Else
Generate successive position $M_i(t + 1)$ using GM operator;
End if

End for
If $OM_i(t+1) < OM_i(t)$
Update position $M_i(t)$ and $M_i(t+1)$ to guide archive;
End if
End if
End for
Update fitness value and position of the global flame;
End while

6. Results and Discussions

The suggested VM migration frequency reduction is tested using the PlanetLab dataset's monitoring infrastructure. The existing approach [25] utilized workload traces obtained over ten days chosen randomly from March and April 2011 to test the structural equation modelling for minimizing migration factors. The archives of the historical folder contain CPU statistics. It gathers the archive every 5 minutes by over 1000 servers in over 500 locations across the globe.

The author also chose ten other days at random to train the prediction model, referred to as the migration dataset. Also, every sequence in this database contains six integers, from the first four characteristics representing CPU utilization before migration. The fifth superior customer services CPU utilization when in out sensors detect the need for movement of people, and the sixth value represents CPU utilization when the U2 limit is exceeded. All rows' first five entries are utilized as the testing phase, and each frame's six variable is used to name it. The line is labeled zero if the sixth value will be less than or identical to the U1 barrier, indicating that no migration is required. The row is also labeled to one, which means that the system state is not enhanced in stipulations of CPU utilization and migration is needed.

6.1. Planet Lab description

Here, a real dataset known as PlanetLab is compared with massive servers located worldwide. The workload comprises 5 days of data with various resource demand profiles attained from the monitoring project. The data traces are now available and operative with CloudSim, where the workloads are used for simulation purposes. The significant features of every five sets: the number of VMs and the mean and SD of the CPU utilization, as depicted in Table 2. The dataset consists of the CPU utilization time based on available VM over a monitoring interval of 300 sec. It is considered for every independent workload executed for the data centre initial size of VM. The traces utilization from the real-time system makes the simulation applicable for a real-time environment.

Date	VMs	CPU utilization (%)	SD of CPU utilization
2011.3.3	1052	12.31	17.09
2011.3.6	898	11.44	16.83
2011.3.9	1061	10.70	15.57
2011.4.12	1054	11.54	15.15
2011.4.20	1033	10.43	15.21

Table. 2 Planet Lab workload features

The proposed CRF-MA model's migration parameters are $L = 45\%, U1 = 75\%\ and\ U2 = 85\%$. In this experimentation, the random and the guided migration are considered based on the moth position, equal to three. Also, the state of the moth is set as four and 10^{-4}, respectively. During the simulation process, the hosting cost with the active model is determined equally during the migration of 20 VM. Here, the maximal value is 250W, i.e. standard for computing servers. In the proposed model, the training process is done for 500 iterations to measure the CPU utilization and other related factors. For 500 iterations, the method is tested on a Core I3 processor with 2.30 GHz PC with 4 GB RAM using the CloudSim simulator. The cost of maintaining hosting in the positive cycle is comparable to the cost of migrating 20 virtual servers in our experiments. The evaluation is performed and compared with another computational model for comparative purposes.

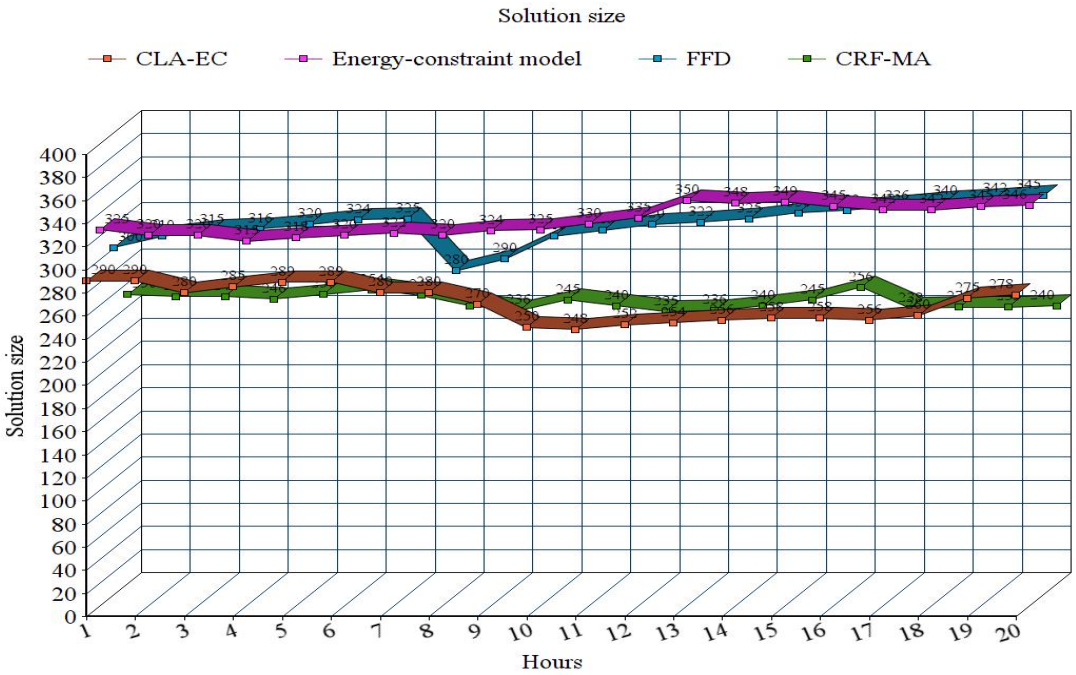

Fig. 3. Solution size comparison

This section explains the experimental outcomes of the anticipated CRF-MA algorithm, where the computation is done based on the measurement with CPU utilization data set gathered from the Project Lab. The solution is acquired with the evaluation of solution size metrics which is expressed as in Eq. (17).

$$Solution\ size = PM * VM_{MC} + MN \tag{17}$$

Here, MN and PN specify the number of physical machines and migrations analyzed with the proposed CRF-MA model, respectively. This work also compared the presented VM migration methodology to two alternatives directly contributing, namely FFD, energy resource allocation model and CLA-EC approaches. Fig. 3 shows the comparison of solution size for these three techniques over a specific time. The suggested strategy CRF-MA outperforms the others in CPU utilization, according to the findings. The proposed technique chooses a process that requires fewer computational nodes and migrations. Fig.4 shows the average migrant needs three techniques to establish a new position through every renewal phase. The suggested method has a lower score than the others, per the data.

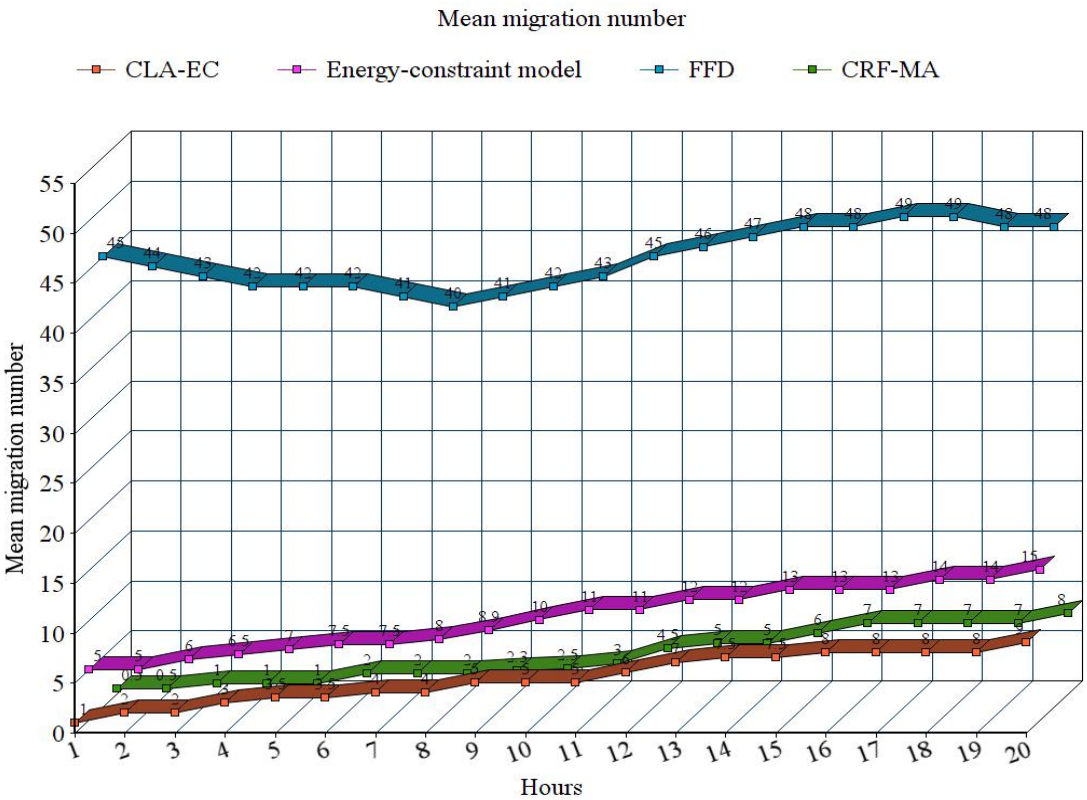

Fig .4 .Mean migration comparison

This method minimizes the likelihood of transfer in two stages. After the criterion passes U2, the system tries to find scenarios that will improve related to future CPU utilization predictions. Second, utilizing the proposed CRF-MA technique, the method seeks to minimize the migratory amount for installation, as shown in Fig. 5. The proposed modelling approach can cut the likelihood of migration by 22%. Regarding migration frequency, the prediction outperforms the prognosis of the existing system. Two methods reduce migration frequency by 22% and 13%, respectively. The average energy consumption of our suggested method compared to the current energy resource allocation model is shown in Fig. 6. The existing energy resource model employs three VM selection initiatives: In 10 days, the Minimum Migration Time Policy (MMT), Random Choice Policy (RC), and Maximum Correlation Policy (MC) all had the very same usage threshold. According to these findings, the suggested CRF-MA technique reduces overall energy usage by 8.5% compared to the other approaches, which obtain the most significant results among VM selection policies. The proposed technique and VM placement strategies are evaluated and compared in the defined SLA Violation (SLAV) measure.

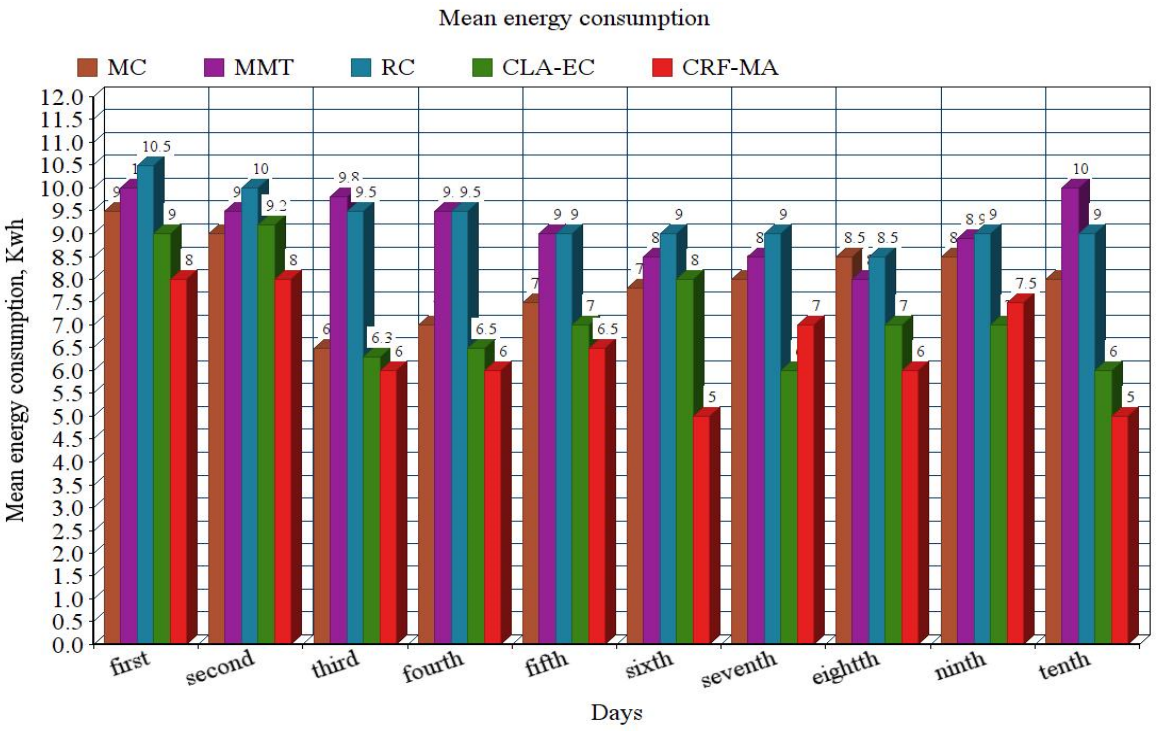

Fig. 5. Mean energy consumption analysis

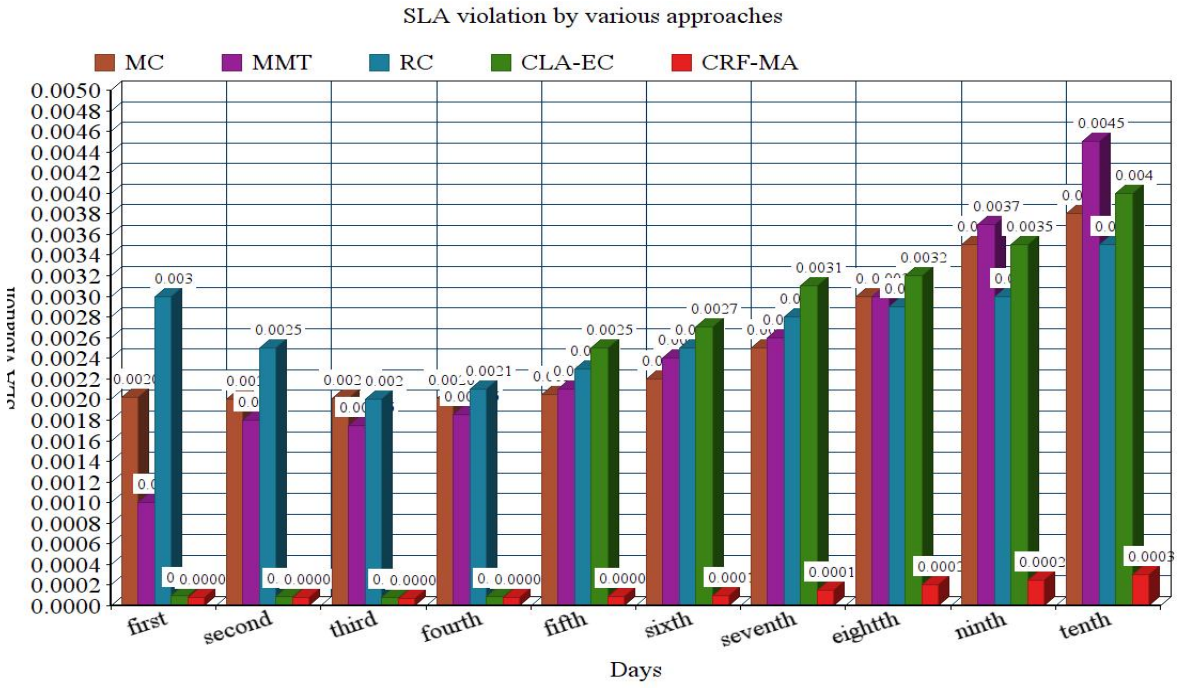

Fig. 6 .SLA violation analysis

The mean value of SLA violation in the existing method is 0.00325, whereas the mean value of SLA violation in the suggested technique is 0.00095, as shown in Fig. 6. The proposed approach may reduce the mean value of SLA violation by 0.00095. Based on these analyses, it is proven that the model works effectually compared to other approaches.

7. Conclusion

This work recommends a novel CRF-MA technique for virtual machine migration based on CRF-MA prediction to improve the security issues. Conditional random field-based VM selection criteria determine the Live migration period. During the migration process, the suggested repair technique is used to locate the best virtual machine location in two basic categories: disparity and VM energy usage. The suggested proposed method is compared to various other approaches w.r.t. CPU usage information attained with the PlanetLab data set. According to the results of the experiments, the presented algorithm outperforms two previous methods in terms of CPU consumption and mean needed transfer quantity, as well as preventing unnecessary migrants. Because virtual machine deployments might result in SLA violations, minimizing the volume of migrants can result in fewer possible SLA contraventions and lower data centre power consumption. As per the findings, the proposed approach minimizes the mean relocation number during every partial substitute, total energy consumption, and SLAV by 59.05%, 8.53%, and 70.76% compared to the other technologies, which attains improved outcomes among some other VM selection policies initiatives.

8. Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Gou, Z., Yamaguchi, S., Gupta, B.B.: Analysis of various security issues and challenges in cloud computing environment: a survey. Identity Theft: Breakthroughs in Research and Practice, pp. 221–247. IGI Global (2017)
[2] Gupta, B.B. (ed.): Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives. CRC Press, Boca Raton (2018)
[3] 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), 372–380 (2017)
[4] Dow, E.M., Fitzsimmons, T.D., Yu, J.: U.S. Patent No. 9,753,757. Washington, DC: US Patent and Trademark Office (2017).
[5] Ghavipour, M., Meybodi, M.R. Irregular cellular learning automata-based algorithm for sampling social networks. Eng. Appl. Artif. Intell. 59, 244–259 (2017)
[6] Fernando, D., Terner, J., Gopalan, K., Yang, P.: Live migration ate my VM: Recovering a virtual machine after the failure of post-copy live migration. In: IEEE INFOCOM 2019—IEEE Conference on Computer Communications, pp. 343–351 (2019).
[7] Gupta, T., Ganatra, J., Samdani, K.: A survey of emerging network virtualization frameworks and cloud computing. In: 2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 14–15 (2018).
[8] Elsaid, M.E., Sameh, M., Abbas, H.M., Meinel, C.: Live migration timing optimization integration with VMware environments. In: 10th International Conference, Cloud Computing and Services Science CLOSER 2020, Prague, Czech Republic, May 7–9. Revised Selected Papers 1399, (2021), pp. 133–152 (2020).
[9] Elsaid, M.E., Meinel, C.: Live migration impact on virtual datacenter performance: Vmware vmotion based study. In: 2014 International Conference on Future Internet of Things and Cloud, pp. 216–221 (2014)
[10] . Strunk, A.: Costs of virtual machine live migration: a survey. In: Proceedings of the 2012 IEEE Eighth World Congress on Services, SERVICES '12, IEEE Computer Society, Washington, DC, USA, pp. 323–329 (2012)
[11] Aldossary, M., Djemame, K.: Performance and energy-based cost prediction of virtual machines live migration in clouds. In: Proceedings of the 8th International Conference on Cloud Computing and Services Science, CLOSER 2018, Funchal, Madeira, Portugal, March 19–21, pp. 384–391 (2018)
[12] Rybina, K., Schill, A.: Estimating energy consumption during live migration of virtual machines. In: IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 1–5 (2016)
[13] Salfner, P.T.F., Polze, A.: Downtime analysis of virtual machine live migration. In: The Fourth International Conference on Dependability (DEPEND 2011), ISBN: 978-1-61208-149-6, pp. 100– 105, French Riviera, France
[14] Nosy, M., Ibrahim, A., Ali, H.A.: Optimization of live virtual machine migration in cloud computing: a survey and future directions. J. Netw. Comput. Appl. 110, 1–10 (2018).
[15] Chen, Y., Liu, I., Chou, C., Li, J., Liu, C.: Multiple virtual machines live migration scheduling method study on VMware vmotion. In: 2018 3rd International Conference on Computer and Communication Systems (ICCCS), pp. 113–116 (2018)
[16] Cerroni, W.: Multiple virtual machine live migration in federated cloud systems. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 25–30 (2014)
[17] Jo, C., Cho, Y., Egger, B.: A machine learning approach to live migration modelling. In: Proceedings of the 2017 Symposium on Cloud Computing, SoCC '17, ACM, New York, NY, USA, pp. 351–364 (2017)
[18] Huang, Q., Shuang, K., Xu, P., Liu, X., Su, S.: Prediction-based dynamic resource scheduling for virtualized cloud systems. J. Netw. 9, 375–383 (2014)
[19] Cerroni, W.: Network performance of multiple virtual machines live migration in cloud federations. J. Internet Serv. Appl. 6(1), 6:1-6:20 (2015).
[20] Bezerra, P., Martins, G., Gomes, R., Cavalcante, F., Costa, A.: Evaluating live virtual machine migration overhead on client's application perspective. In: International Conference on Information Networking (ICOIN), pp. 503–508 (2017).
[21] Melhem, S.B., Agarwal, A., Goel, N., Zaman, M.: A Markov-based prediction model for host load detection in live VM migration. In: 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 32–38 (2017).
[22] Melhem, S.B., Agarwal, A., Goel, N., Zaman, M.: Markov prediction model for host load detection and VM placement in live migration. IEEE Access 6, 7190–7205 (2018).
[23] Bashar, A., Mohammad, N., Muhammed, S.: Modeling and evaluation of pre-copy live VM migration using probabilistic model checking. In: 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1–7 (2018)
[24] Bezerra, P., Martins, G., Gomes, R., Cavalcante, F., Costa, A.: Evaluating live virtual machine migration overhead on client's application perspective. In: International Conference on Information Networking (ICOIN), pp. 503–508 (2017)
[25] Shi, B., Shen, H.: Memory/disk operation aware lightweight VM live migration across datacenters with low-performance impact. In: IEEE INFOCOM 2019—IEEE Conference on Computer Communications, pp. 334–342 (2019)
[26] Katzenberg, N., Golander, A., Weiss, S.: Nvdimm-n persistent memory and its impact on two relational databases. In: IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE), pp. 1–5 (2018

[27] Aloui, M.; Hamidi, F.; Jerbi, H.; Omri, M.; Popescu, D.; Abbassi, R. A Chaotic Krill Herd Optimization Algorithm for Global Numerical Estimation of the Attraction Domain for Nonlinear Systems. Mathematics 2021, 9, 1743.
[28] Ivanov, O.; Neagu, B.-C.; Grigoras, G.; Scarlatache, F.; Gavrilas, M. A Metaheuristic Algorithm for Flexible Energy Storage Management in Residential Electricity Distribution Grids. Mathematics 2021, 9, 2375.
[29] Varaee, H.; Ghasemi, M.R. Engineering optimization based on ideal gas molecular movement algorithm. Eng. Comput. 2017, 33, 71–93.
[30] Sikandar, S.; Baloch, N.K.; Hussain, F.; Amin, W.; Zikria, Y.B.; Yu, H. An Optimized Nature-Inspired Metaheuristic Algorithm for Application Mapping in 2D-NoC. Sensors 2021, 21, 5102.

Authors Profile

Nelli Chandrakala completed M.Tech from JNT University,Anantapur in 2008 .The research Interests Includes Cloud computing,computer networks and Information Security.Presently she is a research scholar in department of Computer Science and Engineering,Koneru Lakshmaiah Education Foundation'Guntur,AP,India

Dr.Vamsidhar Enireddy completed M.Tech from Andhra University in 2008 and Ph.D from JNTU Kakinada in 2017.The research Interests Includes Image preprocessing,Machine Learning,,Data Mining models and Neural Networks.Presently he is working as a Professor in department of Computer Science and Engineering,Koneru Lakshmaiah Education Foundation'Guntur,AP,India