IAGA: Interference Aware Genetic Algorithm based VM Allocation Policy for Cloud Systems

Tarannum Alimahmad Bloch[1]
RDIC C.U.Shah.University
Wadhwan, India

Sridaran Rajagopal[2]
FoCA, Marwadi University
Rajkot, India

Prashanth C. Ranga[3]
School of CS, University of Windsor
Windsor, Canada

***Abstract*—Diversified systems hosted on cloud infrastructure have to work increasingly on physical servers. Cloud applications running on physical machines require diverse resources. The resource requirements of cloud applications are fluctuating based on the resource intensity of the applications. The multi-tenancy of Cloud servers can be achieved based on effective resource utilization. The optimum resource utilization, maximum service level agreement, and minimization of interference are the major objectives to be achieved. Using live Virtual Machine (VM) migration techniques cloud resources can be utilized efficiently. But the migrated VMs can interfere with the ongoing applications on the targeted server which may lead to the service level agreement violation (SLAV) and performance degradation. To resolve this issue, understanding the current state of cloud hosts before the allocation of newly migrated VM is necessary. This paper presents Interference Attentive Genetic Algorithm (IAGA) based VM allocation strategy to achieve the aforementioned objectives. The proposed IAGA policy has outperformed existing policies for quantifiable performance metrics such as energy consumed by cloud systems, count of hosts shut down, average SLAV, and count of VM migrations.**

Keywords—Cloud computing; interference; VM allocation; SLA violation; resource utilization

I. INTRODUCTION

The term Cloud is buzzing technology that can be viewed as the provision of services over the Internet as per the demand of the users. The changeover of large organizations from the traditional Capital Expenditure (CapEx) model to the Operating expenses (OpEx) model supports the reality that the Cloud environment is one of the majority capable technologies in the dated digital era. The escalating number of cloud service consumers has amplified the challenges faced by the Cloud Service Providers (CSPs) to provide the requested services with high availability and reliability of the services. The virtualization technique provides the CSPs to meet these challenges. The main basis of the cloud environment is virtualization.

In virtualized scenarios, the hardware resources of every host also called Physical Machine (PM) are imitated to be independently running entities that are represented as a virtual machine (VM). In the host machine, the cloud service request from the user is managed by the VM to fulfill the computational resource demands such as the size of the memory, computing duration, CPU cycles, network bandwidth, etc. Various cloud systems may support such as the size of the memory, computing duration, CPU cycles, network bandwidth, etc. Various cloud systems may support such virtualizations, in different ways such as operating system-level virtualization, and virtualization based on type-I and type-II [1].

As and when the demand for any VMs' resource increases that particular set of VMs, in the running state, needs to be migrated and accommodated to the new host or physical machine. This entire procedure is known as live VM migration. The VM allocation is the sub-process of the full-stack live VM migration process. Once the migration decision is made by the virtual machine monitor of a particular host, the host needs to look for a new physical machine by keeping various constraints in an account. Such constraints include uninterrupted resource sharing of the existing VMs host with the migrated host after allocation.

The performance of the overall host should not be regretted because of the resource claim of newly migrated VMs and the SLA of the applications hosted on the migrated VMs should not be violated. An interference-aware technique of VM allocation is a must to achieve the aforementioned objectives. This paper discusses an Interference Attentive Genetic Algorithm-based VM allocation policy named IAGA. It is designed to achieve interference minimization by allocating the migratable VMs to the best-suited PM while maintaining high SLA and optimized resource utilization of cloud hosts.

In general, this research article aims to:

1) Discuss the state-of-art current trends in VM allocation policies.

2) The involvement of Genetic Algorithms in cloud service delivery.

3) Mathematical model, design constraints, and algorithm design of proposed technique for VM allocation using the genetic algorithm to address resource interferences.

4) Experimental result analysis of the proposed approach with the existing approaches.

The structure of the paper is as follows: Section 2 discusses the state-of-the-art and the details of genetic algorithm variants used by various researchers. Section 3 gives elaboration on the proposed system model along with the mathematical model, design constraints, and algorithm design. Section 4 talks about the experimental scenario and achieved results. Section 5 concludes the paper and the possible future extension of the proposed research.

II. State-of-the-Art

VM allocation is the major decision involved in the live VM migration procedure which raises many concerns such as proper resource utilization, increased throughput, SLA maintenance, energy consumption minimization, etc. Various optimization techniques have been applied by the researchers. According to Christina et al, VM allocation is a multi-objective constrained optimization NP-Hard problem [2].

Yuzhe et al proposed a system for optimizing VM allocation techniques in uncertain cloud environments based on the user requirements. The authors considered the optimization perspective of energy consumption for the data center for VMs with no special needs. For the remaining Virtual machines, the present throughput of the hosts and the service consumer's bandwidth requirements are considered in the allocation process of Virtual machines. They designed a VM allocation system to significantly improve multiple objectives such as proper resource utilization, minimization of PMs used, and minimization of energy consumption by taking into consideration the cost of data transmission between VMs [3].

As per Jenn-Wei et al, without considering the VM interference in the VM placement requirement, the Quality of Service (QoS) requirements of the cloud application executing in VMs may be violated. The authors have considered three factors in the proposed VM placement policy: (i) Resource demand of virtual machines (ii) The QoS of cloud applications (iii) The VM interference. As per the authors' point of view, it is difficult to accommodate all the aforementioned factors in the VM placement policy. Authors have named research problem as IAVMP – Interference Aware VM Placement problem. They have formulated an integer linear programming model to solve IAVMP as an NP-complete problem [4].

Sasmita et al considered a single parameter optimization technique to achieve the goal of cloud service consumers or cloud service providers. In realism, the user and service provider have opposing goals. This objective could be used to guide the selection of cloud hosts. They proposed an Efficient Multi-optimization Resource Allocation (eMRA) model using optimization techniques to achieve the goals of cloud service consumers and data centers in the proposed work. SGO (Social Group Optimization) technique is proposed to improve user requests by taking into account related parameters for allocation. Similarly, Particle Swarm Optimization (PSO) is being used to improve data center lists that are fitting for optimized user requests. To design the model that separates the proposed design model from other existing works, the eMRA considers distinct related parameters of cloud service consumer request, cloud host, and network. The eMRA technique is simulated using CloudAnalyst, and the authors researched ten distinct scenarios using three different CloudAnalyst broker policies [5].

S. Savitha et al. presented a perceptive priority-aware VM allocation strategy called the P-PAVA algorithm which determines an application's priority and also resource needs. Using an ML-based prediction model, the system assigns applications based on one's priority. Moreover, parallelization has been used before conveying various workloads to reduce the overhead of the allocation algorithm. To accomplish this, the algorithm uses the first fit technique as a baseline for user request allocation with a low priority norm. P-PAVA outshines the state-of-the-art algorithm for VM allocation for priority-aware applications on a variety of indicators such as average response time, execution time, and power consumption [6].

Garg R. et al. developed a virtual machine allocation policy to evaluate the behavior of commonly used heuristic allocation policies. To substantiate the logistic comparison, a policy that implements a bin-packing approach in Virtual machine allocation has been developed. The implemented research problem has been rigorously tested using a diverse range of workload data sets and experimental configurations. The performance of these algorithms is also evaluated with varying threshold and VM selection policies. Experimental results indicate that policies that take into account the server's power and computing capacity perform much better in almost all scenarios [7].

Jitendra et al. created an Intelligent SLA-aware and Energy Minimization VM allocation approach that uses the Emperor Penguin Optimization (EPO) algorithm. In a heterogeneous cloud environment, the system could indeed allocate virtual machines based on power usage. The proposed method has illustrated its appropriateness for virtual machines in the data center by making comparisons of it to the Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Binary Gravity Search Algorithm (BGSA). The outcomes of the proposed system have been evaluated by using the JAVA simulation platform. The investigation results reveal that the improved EPO-based system is very effective in limiting energy consumption, SLA Violations (SLAV), and the advancement of QoS requirements to provide desirable cloud services [8].

Rahimi Zadeh et al. [9] proposed an interference-aware and joint profit scheduling scheme (PIAS) to proficiently consolidate Virtual machines on the physical machine that hosts multi-tier application workloads in Infrastructure-as-a-Service (IaaS). For consolidating VMs, the PIAS policy took into account resource usage, costs and profits of power consumption, service level agreements (SLAs), and operational interference of VMs along with the number of transferred memory pages during live VM migration. The functioning behavior patterns of VMs are examined in this work. Furthermore, an optimization problem is presented to achieve maximum cloud service provider profit while reducing the overall cost of application workload executions.

From the various literature [11-15], it is clear that the impact of the interference is unavoidable to achieve high SLA. Recently also many researchers have developed VM migration techniques by keeping the different objectives in the center. CMIG – is a Concurrency-aware Migration System [10]. Rachael et al [11] used an intelligent approach to minimize interference and energy consumption. Anu et al [12] addressed interference minimization in the proposed system Interference Aware Live Migration (IALM). Babu et al [13] have designed an interference-aware system that has automatic scaling support to handle sudden load drift with precise prediction and minimum VM migration. Chao et al have addressed the interference awareness during the migration process to mitigate cache-based side challenge attacks in the cloud [14]. Yiling et al have proposed the live migration technique which applies to software-defined networks with awareness of interference and topology [15].

Neha et al [38], proposed an optimized VM allocation scheme called Resource Aware Provisioning (RAP). For the VM allocation criteria, the authors considered better energy efficiency and a fixed upper threshold value. RAP is working with the static upper threshold and the authors have not considered all types of required resources for different applications running inside VMs. The limitation of the RAP is that it has not taken the future resource type intensity into the consideration while choosing the PM for VM allocation. For example if one of the newly migrated VM starts grabbing more CPU cycles and if the PM is about to reach the upper threshold then any proactive steps are not taken by the authors. Only the energy consumption has been considered as a major parameter for the VM allocation which is not sufficient for the optimized VM allocation approach because interference minimization is an unavoidable parameter that has not been considered by RAP.

Genetic Algorithms have been applied significantly during the current era to achieve the optimized solution to various research problems. It has been observed from various works of literature that the involvement of genetic algorithms and variants has given improved results in the various cloud computing-related research problem solutions. The genetic algorithms have been applied to address the cloud security related issues [16-18], to achieve optimum solutions in the Internet of Things (IoT) service placement in fog computing environment cloud computing [19], and to leverage the fog computing frameworks the human activity recognition has been achieved through deep genetic algorithm [20].

Genetic algorithms are considerably being used for a better cloud service experience. The list of the Genetic algorithm and its variants applied in recent research trends by various researchers [21-32] for better cloud service is presented in Table I.

TABLE I. APPLICATION SUMMARY OF GENETIC ALGORITHM VARIANTS IN CLOUD SERVICE DELIVERY

Reference No.	**Authors**	**The variant of the Genetic Algorithm**	**Application**	**Experiment Environment**
[21]	Shilpa K., Sujata T.	Hybrid Max-Min Genetic Algorithm	Load Balancing	CloudSim Simulator
[22]	Jiawei L., Haotian Z., Wei Z., Jie Li, Gang ., Zhenbo C.	Improved Genetic Algorithm	Optimal VM Placement Strategy	Real Environment FEA - Finite Element Analysis
[23]	Zhang, B., Hao W., Xiao W.	Cluster-based Genetic Algorithm	VM Placement Strategy	Not mentioned
[24]	Mehran T., Mohammad I., Mostafa G.	Micro-genetic algorithm	VM Allocation Strategy	CloudSim Simulator
[25]	Huda I., Khaled E., Raafat O. A.,	Adaptive Genetic Algorithm	Dynamic Task Scheduling Strategy	CloudSim Simulator
[26]	Muhammad S., Muhammad T.	parallel multi-objective genetic algorithm	Task Scheduling for scientific workflow	CloudSim Simulator
[27]	Abbas A., Ahmad K., Seyed M.G.	Thermal Aware Genetic Algorithm	VM Allocation Strategy	CloudSim Simulator
[28]	Einollah J.G., Amir M. R, Nooruldeen N. Q.	Genetic Algorithm	Service Load Balancing	Matlab
[29]	Carlos G., Isaac L., Carlos J.	Non-dominated Sorting Genetic Algorithm – II	Micro Services Resource Allocation	Real Cloud Environment
[30]	Zhou Z., Fangmin L., Huaxi Z., Houliang X., Jemal H. A., Chowdhury, Morshed U	Improved Genetic Algorithm	Optimized Task Scheduling	CloudSim Simulator
[31]	Madhusudhan H S, Satish, K. T, Syed Mustapha.F. D., Punit G., and Raja P. T.	Genetic Algorithm	Resource Allocation	CloudSim Simulator
[32]	A. J. Miriam· R. Saminathan, S. Chakaravarthi	Non-dominated Sorting Genetic Algorithm – III	Resource Allocation	Hadoop Cluster

III. Proposed System

This section represents the proposed system architecture, mathematical model, design constraints, and algorithm design.

A Genetic Algorithm (GA) is a well-known meta-heuristic algorithm that is inspired by the biological evolution process [40]. In nature, GA mimics the Darwinian theory of the survival of the fittest. GA's fundamental components are chromosome representation, fitness, selection, and biologically inspired operators [40]. Chromosomes can be thought of as points in the solution space. These are processed by iteratively replacing the population with genetic operators. The fitness function is used to assign a value to each of the population's chromosomes [39].

In selection operation, chromosomes are chosen for further processing based on achieved fitness value. In the crossover operator, a random convergence point is chosen and the subsequences between chromosomes are changed to produce offspring [40].

For the proposed research problem the chromosomes are considered as the PMs to allocate the migrated VMs. The fitness function is designed by considering the total resource capacity of the server.

Crossover is performed to choose the best PMs by keeping the future resource demands of the VMs hosted on that PM. The following section discusses the proposed architecture, design constraints, mathematical model of the research problem, and ultimately the algorithm design presented thereof.

A. Proposed System Architecture

Fig. 1 depicts the system architecture of the proposed VM allocation system. In the cloud environment, all the PMs resource usage will be monitored especially at the time when that PM is the candidate to be chosen for the accommodation of a new set of migratable VMs. A genetic algorithm is applied in the proposed system to choose the best suitable PM to allocate migratable VMs.

In the proposed architecture, the major components are:

- Local Resource Monitor.
- Local Interference Monitor.
- Global Interference Monitor.
- Historical Resource Usage.
- Resource Usage-based VM Categorization System.

The local resource monitor keeps track of resource usage based on the resource intensity type and the local interference monitor observes the interference by considering SLA. From the study of existing systems, it is clear that during live VM migration, allocation of optimum resources and SLA management is the major objective. To achieve this research objective the proposed system must be clear about the available resource capacity of the chosen PM for the allocation of migratable VMs. The candidate PMs mostly has varying resource capabilities. The used resource amount of PMs depends on the hosted applications running on the VMs of that server. Based on this scenario the application need or VM resource intensity can be categorized as:

- CPU intensive VMs.
- Memory intensive VMs and.
- Network bandwidth-intensive VM.

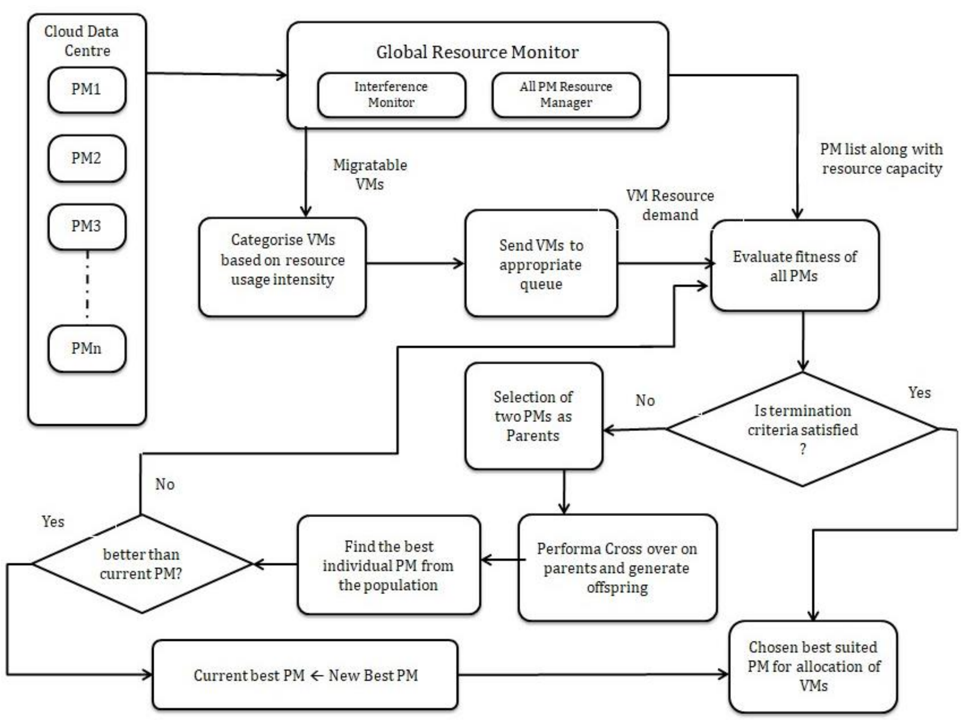

Fig. 1. Proposed System Architecture.

It is necessary to allocate the VMs based on the available resource type intensity of the PM. So in the proposed design the interference monitor categorizes VMs and aligns them to the appropriate queue before allocation using a resource usage-based VM categorization system. Categorization is performed based on the prediction carried out using historical resource usage. The global resource monitor keeps track of all the PM resource usage states and lists of migratable VMs selected by the overloaded PM in case of excessive resource usage. Ultimately the decision support system takes care of the migration process to the chosen destination PM. The list of chosen PMs and migratable VM list will be given as input to apply the genetic algorithm. The strength of the proposed solution is the fitness function evaluated for all the PMs. The fitness function is discussed in Section III C.

B. Proposed Mathematical Model

Let V be the set of n migratable VMs.

$V = \{VM_1, VM_2, VM_3, \ldots\ldots VM_n\}$

where n = number of migratable VM.

Let P be the set of PMs in the cloud environment.

$P = \{PM_1, PM_2, PM_3, \ldots\ldots PM_m\}$.

where m = number of PMs in set P.

Logically overall performance degradation due to the VM migration interferences on PM denoted by MI can be calculated as the total of co-location interference that occurred because of the resource share claim of newly migrated VMs known as co-location interference MIC and network interference is migration network time MNW.

So, for PM the total interference

$$MIC_i = CI_i + MNW_i \tag{1}$$

Based on the resource interference taxonomy provided in [33] the performance degradation due to the co-location interference is the total of CPU contention interference, memory consumption interference, and network bandwidth contention interference. Here co-location interference is denoted by CI. So, for i^{th} PM the co-location interference can be defined as:

$$CI_i = M_c + M_m + M_{nb} \tag{2}$$

Where

M_c is CPU contention time.

M_m is memory contention time, and

$M_{mb}$is network bandwidth contention time.

The CPU contention time M_c is the ratio of C_{dem} CPU cycles in demand C_{dem} and C_{avl} available CPU cycles that can be allotted. M_c can be represented as:

$$M_c = C_{dem} / C_{avl} \tag{3}$$

The severity of the performance degradation increases if the ratio increases. The demand for the CPU cycles C_{dem} can be defined as the sum of the CPU cycles being used by VMs in execution C_e and the waited CPU cycles by the VMs in the waiting queue C_{que}.

$$C_{dem} = C_e + C_{que} \tag{4}$$

Therefore, the overall performance degradation of i^{th} PM due to the network interference and co-location interference can be thought of as under:

$$I_i \approx f(CI_n, MNW_k)$$

So,

$$I_i = a \cdot CI_n + b \cdot MNW_k \tag{5}$$

where a and b are constants that are used to regulate the values.

C. Proposed System Design Constraints

At the time of VM allocation policy design, for k number of PM used from the P set of available physical machines and V set of migratable VMs there are certain design constraints (DC) that are mandatory to be considered:

1) Design Constraint 1: Each VM has to be allocated to one and only one physical machine

$$V = \bigcup_{P_k \in P} V_k$$

2) Design Constraint 2: To minimize the co-location interference, the requested resources such as CPU cycles, network bandwidth, and memory should not exceed the total capacity of the server.

$$\sum_{i=1}^{n} V_i^{CPU} \leq \sum_{k=1}^{m} P_k^{CPU}$$

$$\sum_{i=1}^{n} V_i^{mem} \leq \sum_{k=1}^{m} P_k^{mem}$$

$$\sum_{i=1}^{n} V_i^{bw} \leq \sum_{k=1}^{m} P_k^{bw}$$

Where,

V_i^{cpu}= CPU contention of i^{th} VM

P_i^{cpu}= CPU contention of i^{th}PM

V_i^{mem} = memory contention of i^{th} VM

P_i^{mem} = memory contention of i^{th} PM

V_i^{bw} = network bandwidth of i^{th} VM

P_i^{bw} = network contention of i^{th}PM

3) Design Constraint 3: The total allocation capacity of the server for i^{th}PM to proceed with the new allocation is defined by SC:

$$SC = \sum_{P_i \in P} u_i^{cpu} + \sum_{Pi \in P} u_i^{mem} + \sum_{P_i \in P} u_i^{bw} \tag{6}$$

where,

u_i^{cpu} is unused CPU cycles from the total CPU cycles of i^{th} PM consider it as x.

u_i^{mem} is unused memory from the total memory of i^{th} PM consider it as y.

u_i^{bw} is unused network bandwidth from the total network bandwidth of i^{th} PM consider it as z.

A genetic algorithm [34] is a method of an arbitrarily defined searching method with improved optimization and autonomic implied parallelism. GA can be used to automatically govern the search direction using probability, as well as to acquire and instruct the optimum searching space [35, 36]. Considering the advantages of the genetic algorithm mentioned in [34, 35, 36], the proposed algorithm employs an interference attentive genetic algorithm-based VM allocation strategy for live VM migration in a computing environment.

This method calculates resource usage in advance by taking historical data and current states into account, which will have an impact on the entire cloud system. By considering the above-described design constraints a genetic algorithm-based Interference Attentive VM allocation algorithm has been designed.

This research solution focuses on the total resource capacity of the server denoted as server capacity SC before allocating VMs to it.

As per design constraint number 3, there is a need to reduce the sum deviating from SC i.e. $| x + y + z - SC |$ should be zero. Hence the fitness can be considered as the inverse of $| x + y + z - SC |$.

So,

$$f(x) = 1 \div | x + y + z - SC | \tag{7}$$

where

$f(x) \in [0, 1]$

In the proposed solution fitness function is the driving factor for VM allocation. The fitness function is designed by considering total server capacity which considers unused CPU cycles, memory, and bandwidth. Before allocation, the migrated VM demand and available server capacity are being evaluated for choosing the optimal solution to the VM allocation problem. The interconnections of the applied solution mathematically prove that the proposed fitness function will reduce the interference raised by migratable VMs.

D. Proposed Algorithm

The section discusses the proposed algorithm design. The VM allocation to the PM should be based on the probability of fitness value of a particular PM's resource utilization described in equation (7).

The design of the IAGA – Interference Attentive Genetic Algorithm-based VM allocation policy is in the direction of achieving the aforementioned research objective while simultaneously keeping discussed design constraints in mind. There are various strategies for finding the best genes for step 6.

The fitness ratio-based selection algorithm is used in this research. The calculation of the fitness value for each PM in the present PM population has been derived first and then kept for the individual PM with the highest score in the next generation. After that for allocation, every PM probability of the accommodation based on the fitness ratio is calculated.

Algorithm: Proposed VM allocation policy

Input PM: PM list, VM: Migratable VM list
Output P: best suitable PM for allocation of migratable VMs

Procedure:

```
For all the PMs in the list initialize the random population
    Categorize VMs into different categories based on the resource intensity as well the prior history of resource utilization
    Send VM to the appropriate queue
  Evaluate fitness based on utilization of resources for each PM based on eq. (7)
  While termination condition (! = generation count) do
      Parents ← select two parent individuals from the PM list according to the fitness value.
      For each parent1, parent2 do
            Offspring1, offspring2 ← Crossover (parent1, parent2)
            Apply mutation on generated offspring
            Find the best individual in the PM population
            If the best individual in the population is better then
                  current best individual PM ← new best individual PM
            End If
  End For
End While
Return best-suited PM
```

E. Performance Metrics

Quantitative parameters such as energy consumption, amount of VM migrations, amount of host shutdowns, and average service level agreement violations are all properly considered. The following are explanations of the aforementioned performance matrices:

- Energy Consumption: The sum of energy consumed by each host during the overloading, underloading, and migration procedures.
- The number of VM migrations: The number of VMs migrated from one host to another during the allocation procedure while migrating.
- The number of host shutdowns: The number of host shutdowns performed during the migration process to minimize energy consumption as much as possible.
- Average Service Level Agreement Violation (SLAV): The average value of quantitative service level agreement violation due to migration and/or interference across all VMs.

IV. Experimental Setup and Result Discussion

The live VM migration can be considered as the whole process of host overload or underload detection, choosing the VMs for the migration and placing these VM in the target host. The term VM allocation policies are interchangeably used with PM selection policies. From the various literature discussed here, it is clear that while allocating the host to the migratable VMs different researchers have kept different research goals into consideration. Some of the common goals are energy minimization and SLA maximization. The major diverse effect of the live VM migration is the interference which has been considered by very few researchers. For the VM allocation strategy, one of the major goals should be the proactive strategy about the co-location interferences that may occur after the allocation VMs to the targeted PM along with optimum resource allocation to the ongoing VMs and newly migrated VMs. High SLA can be achieved if the resource allocation is as per the promised amount. The resource allocation could be proper if the newly migrated VMs are not exceeding their share by creating trouble for ongoing VMs. The interference awareness is the novel approach applied to the VM allocation policy to achieve multiple aforementioned research objectives.

To implement and test the proposed system the CloudSim simulator has been used. The CloudSim toolkit, developed by Rajkumar Buyya et al., was used for the experimental implementation [37], and then after the CloudSim is being used as a low-cost simulation environment for cloud-based research projects. It is an open-source simulation tool that most researchers use to simulate the cloud environment. The workload dataset compiled by PlanetLab has been used in the experimentation. The workloads used in this investigation are 20110303, 20110325, and 20110420. PlanetLab workloads are also included in the CloudSim simulator package.

Certain algorithms for VM allocation and VM selection approaches are also available in the CloudSim tool kit. Virtual machine allocation policies such as Static Threshold – THR, Median Absolute Deviation – MAD, Inter-Quartile Range – IQR, Local Regression – LR, and Local Robust Regression – LRR are given in the simulator. There are certain policies for VM selection, which include Maximum Correlation – MC, Minimum Migration Time – MMT, and Random Selection – RS. These are the benchmark algorithms used by many researchers for the comparisons of proposed systems.

LR and MAD VM allocation policies have been considered in the experiment. VM selection policies, MMT, MU, and RS have been used. The validation of the proposed system Interference Attentive Genetic Algorithm IAGA has been carried out against these policies. Workloads have been applied in various configurations of VM allocation and host overload detection policies. The evaluation has incorporated the IAGA policy and studied the imperial results of the aforementioned experiment scenarios. The proposed algorithm was also compared to Neha et al [38] RAP VM allocation policy.

The VM allocation approach applied in RAP is based on the static upper threshold whereas in the modern cloud application the workload fluctuates based on the high and the low number of user requests for a particular cloud application. Due to this, the static threshold will not lead to the appropriate resource utilization. The proposed VM allocation system is developed using a genetic algorithm.

A genetic algorithm is an adaptive heuristic search-based solution that presents the intelligent utilization of search space to solve the optimization problem. From the graphs, it is proved that for the dynamic workload-based cloud systems, the adaptive solution performs better than the static threshold-based policy for the VM allocation problem.

So the given experiments study the effect of the applied genetic algorithm for searching for the best suitable PM for the available list of candidate PMs. So the existing VM allocation policies have been modified to proceed further with the Genetic algorithm principles. The behavior of the system under study has been observed by various combinations specified in the result graphs.

Fig. 2, Fig. 3, and Fig. 4 show the number of hosts shut down for mentioned workloads. The resultant graph indicates that the count of hosts shutdowns is less in IAGA. Less number of host shutdowns will reduce the downtime for the ongoing application on the cloud host. This will lead toward the goal of SLA maximization. It can be observed in Fig. 5, Fig. 6, and Fig. 7, that the IAGA reduces energy consumption by the PMs. Fig. 8, Fig. 9, and Fig. 10 show the number of VM migrations. Average SAL violations achieved through experimentation have been presented in Fig. 11, Fig. 12, and Fig. 13. The results show that the SLA violation is lower for the proposed approach IAGA to ensure more effective service to cloud service consumers compared to the existing policies.

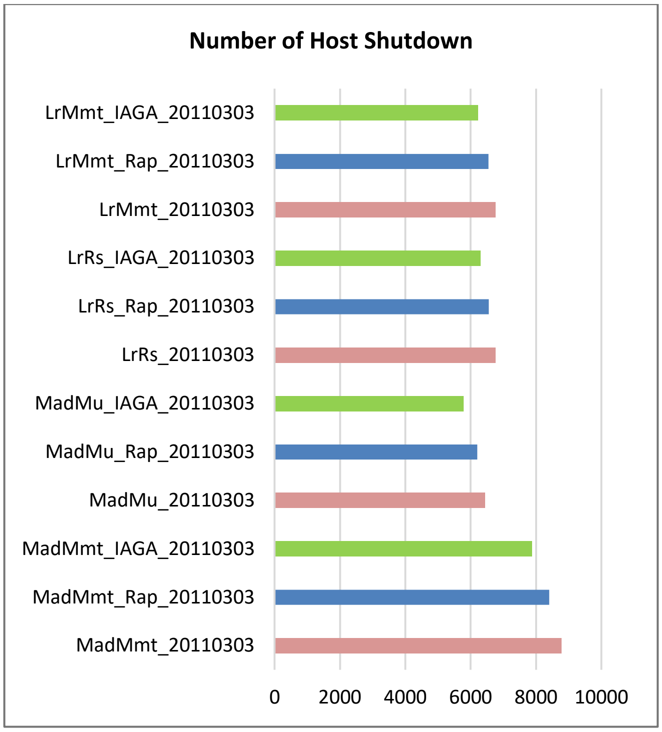

Fig. 2. Number of Host Shutdown for Workload 201100303.

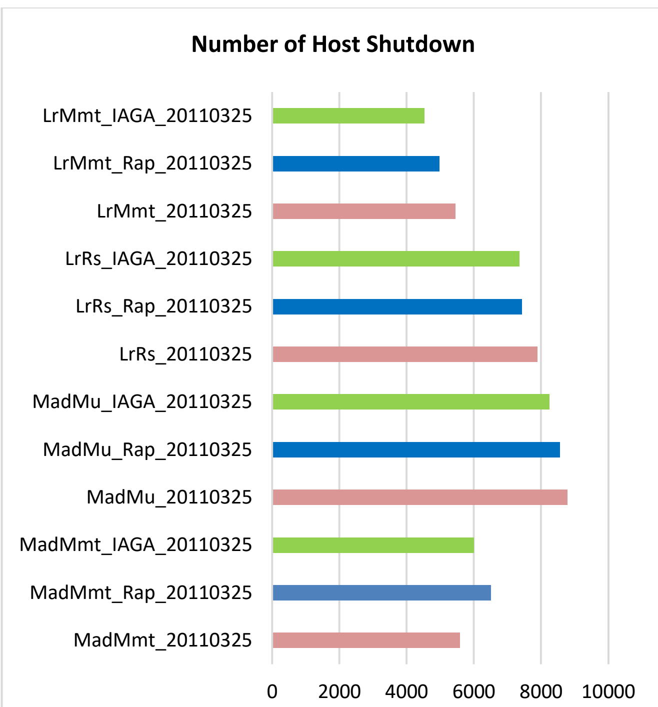

Fig. 3. Number of Host Shutdown for Workload 20110325.

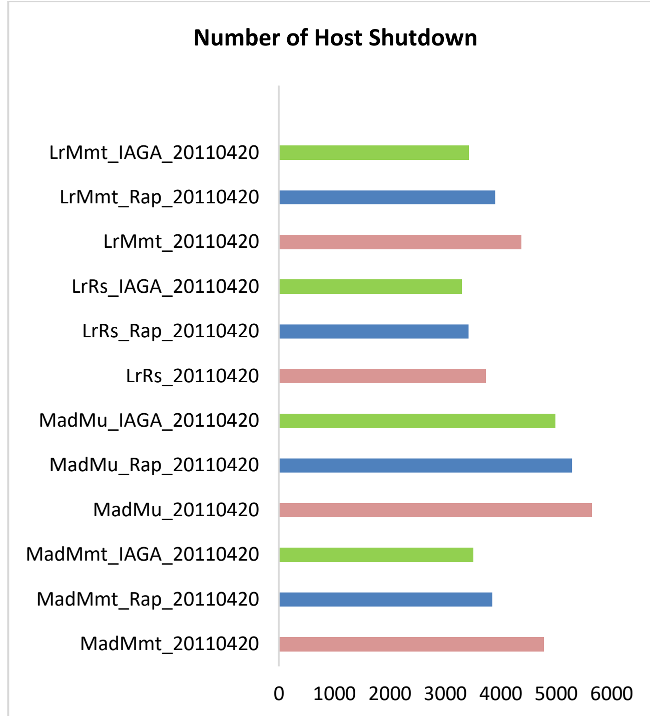

Fig. 4. Number of Host Shutdown for Workload 20110420.

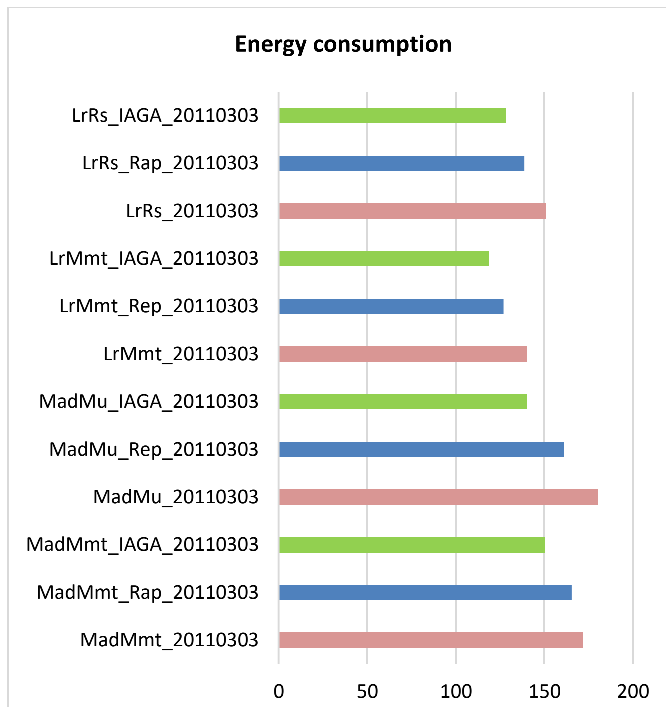

Fig. 5. Energy Consumption for Workload 20110303.

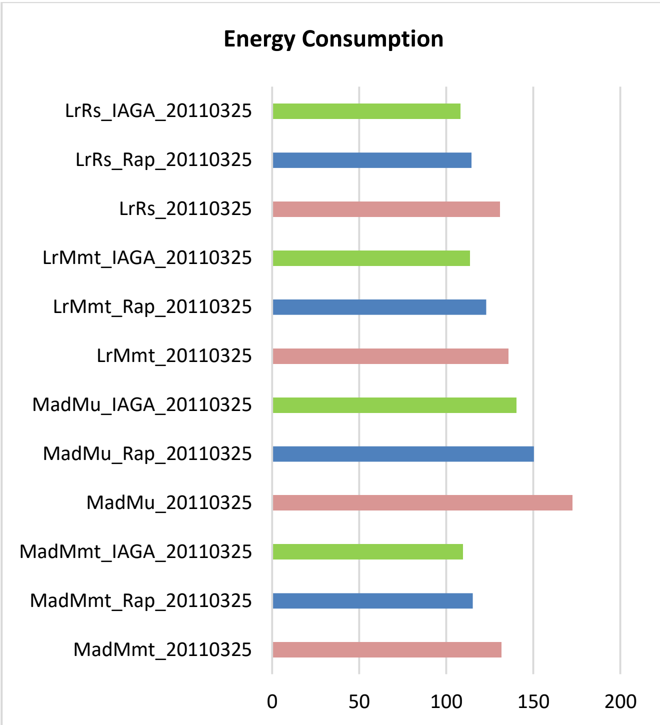

Fig. 6. Energy Consumption for Workload 20110325.

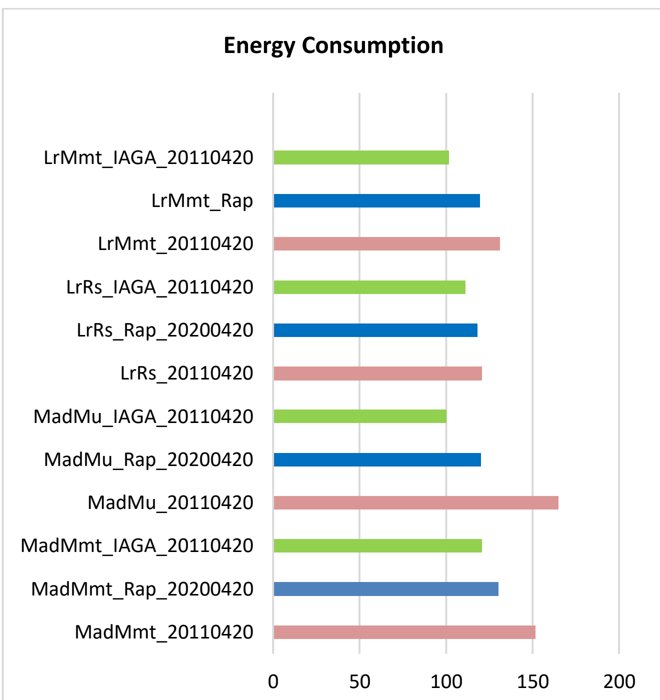

Fig. 7. Energy Consumption for Workload 20110420.

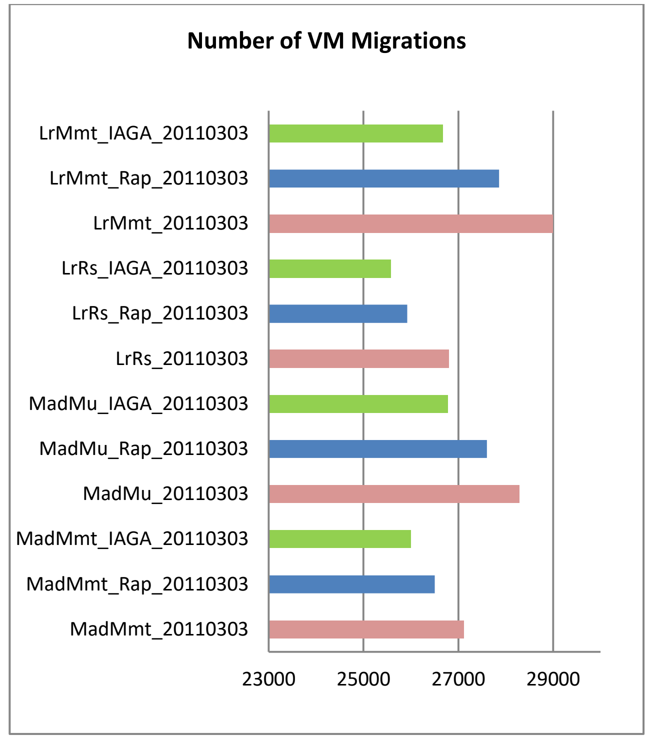

Fig. 8. Number of VM Migrations for Workload 20110303.

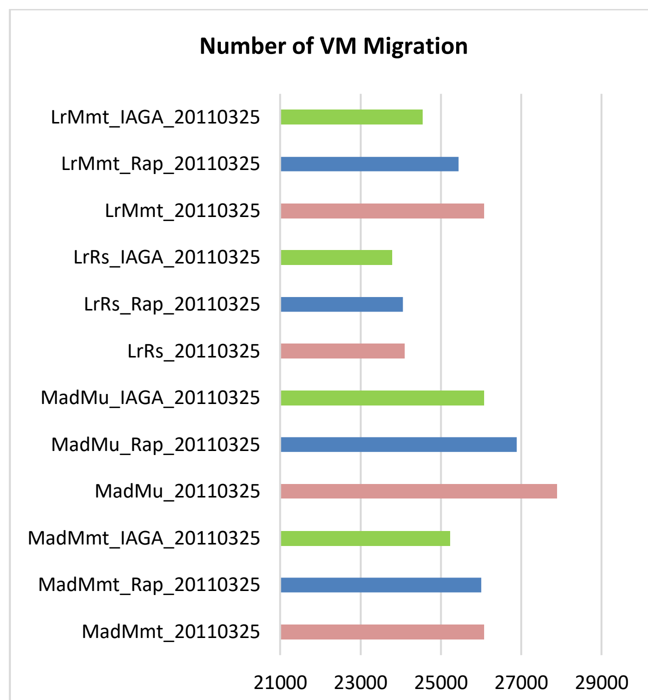

Fig. 9. Number of VM Migrations for Workload 20110325.

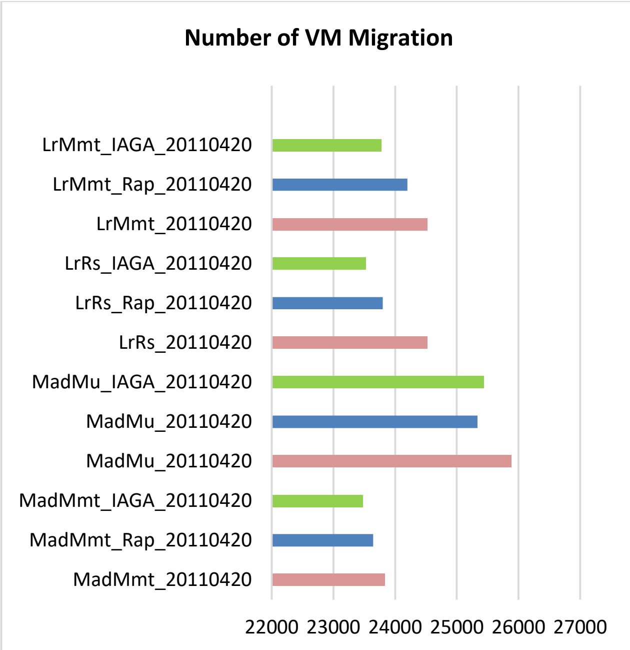

Fig. 10. Number of VM Migrations for Workload 20110420.

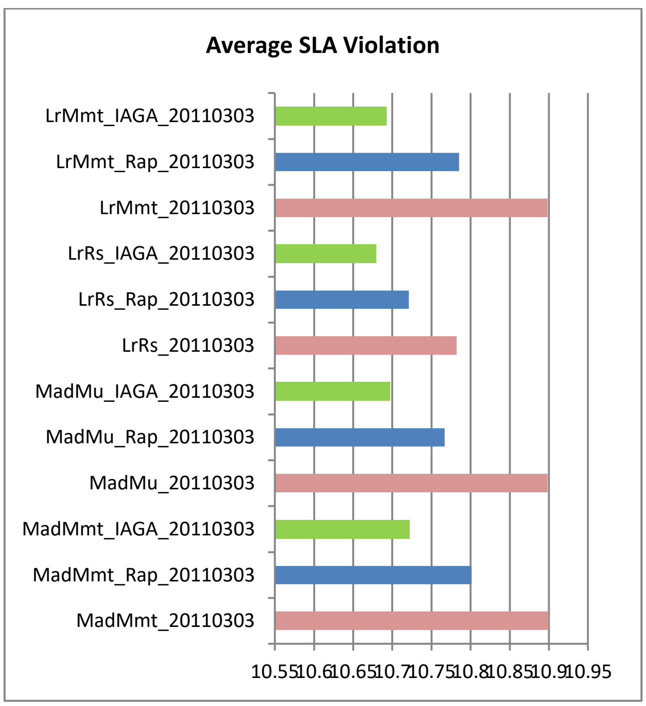

Fig. 11. Average SLA Violation for Workload 20110303.

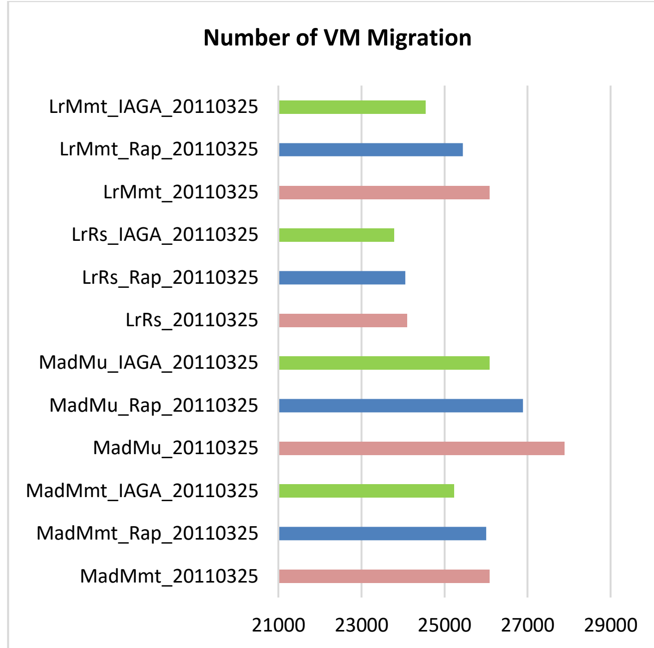

Fig. 12. Average SLA Violation for Workload 20110325.

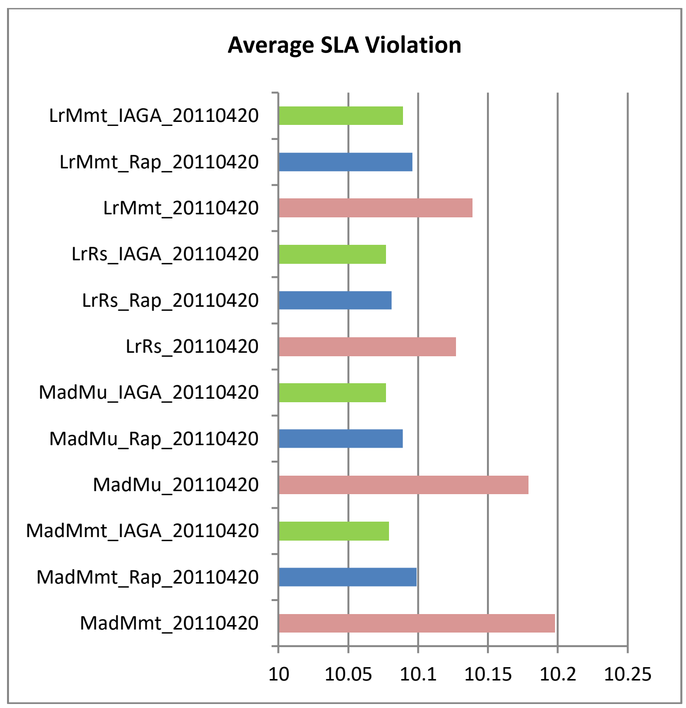

Fig. 13. Average SLA Violation for Workload 20110420.

V. Conclusion and Future Work

The movement of VMs across the system for proper resource utilization is one of the solutions that may result in outlays and service interruptions, resulting in a drop in service quality by affecting SLA. The data centers in a cloud environment require a large number of VMs to run continuously with a high SLA demand. Proposed research experimentation focuses on interference attentive VM allocation. The main challenge of the proposed research work was to strike a balance between effective server resource utilization and the preservation of cloud resources while adhering to SLA constraints with minimized interference effects. The fitness function is a critical component of the proposed method for reducing interferences in Infrastructure as a Service (IaaS) Cloud data centers as it helps the proposed algorithm to find the best optimum PM from all the candidate PMs for the allocation of migratable VMs. One of the most inescapable principles of cloud computing systems that should be pursued is the principle of service quality. As a result, in proposed work concentrated on cloud computing systems competence characteristics such as the current state of PM computing resources specifically resource usage and demand queue to ensure the expected quality of service with minimal interferences from newly moved VMs during live VM migration. To improve the efficiency of the traditional algorithms, a Genetic Algorithm has been applied for VM allocation; the paper proposes a VM allocation policy to address the reduced number of VM migrations, reduced energy consumption, and low SLA violations. The promising results obtained from the proposed approach reveal that the genetic algorithm can be efficiently applied in real data centers to achieve interference minimizations. Furthermore, as the energy consumption decreases, it can also be used in green data centers. The proposed system has achieved the research objectives with a noticeable improvement compared to the existing approaches. The proposed research solution is useful to the cloud service provider to enhance the SLA and achieve satisfactory resource allocation by using the cloud server capacity at the optimum level. As of now, the proposed approach has been tested in the Cloud simulation environment. In the future, this approach could be extended by implanting it in a real cloud environment, the behavior of which could be studied and could be validated for other workloads and/or live streaming applications.

References

[1] Tzenetopoulos, Achilleas, et al. "Interference-aware workload placement for improving latency distribution of converged HPC/Big Data cloud infrastructures."

[2] Christina Terese Joseph, K. Chandrasekaran, Robin Cyriac, “A Novel Family Genetic Approach for Virtual Machine Allocation,” Procedia Computer Science, Volume 46,2015, Pages 558-565, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.02.090.

[3] Y. Huang, H. Xu, H. Gao, X. Ma, and W. Hussain, "SSUR: An Approach to Optimizing Virtual Machine Allocation Strategy Based on User Requirements for Cloud Data Center," in IEEE Transactions on Green Communications and Networking, vol. 5, no. 2, pp. 670-681, June 2021, DOI: 10.1109/TGCN.2021.3067374.

[4] Jenn-Wei Lin and Chien-Hung Chen, "Interference-aware virtual machine placement in cloud computing systems," 2012 International Conference on Computer & Information Science (ICCIS), 2012, pp. 598-603, DOI: 10.1109/ICCISci.2012.6297100.

[5] Parida, S., Pati, B., Nayak, S.C. et al. “eMRA: an efficient multi-optimization based resource allocation technique for infrastructure cloud” J Ambient Intell Human Comput (2022). https://doi.org/10.1007/s12652-021-03598-8.

[6] S. Savitha and S. Salvi, "Perceptive VM Allocation in Cloud Data Centers for Effective Resource Management," 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1-5, DOI: 10.1109/I2CT51068.2021.9417960.

[7] Garg R., Arora I., Gupta A. (2021) “Performance Evaluation of VM Allocation Strategies on Heterogeneous Environments in Cloud Data Center,” In Panigrahi C.R., Pati B., Pattanayak B.K., Amic S., Li KC. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 1299. Springer, Singapore. https://doi.org/10.1007/978-981-33-4299-6_43.

[8] Jitendra Kumar Samriya, Subhash Chandra Patel, Manju Khurana, Pradeep Kumar Tiwari, Omar Cheikhrouhou, "Intelligent SLA-Aware VM Allocation and Energy Minimization Approach with EPO Algorithm for Cloud Computing Environment," Mathematical Problems in Engineering, vol. 2021, Article ID 9949995, 13 pages, 2021. https://doi.org/10.1155/2021/9949995.

[9] Rahimi Zadeh, K., Dehghani, “A. Design and evaluation of a joint profit and interference-aware VMs consolidation in IaaS cloud datacenter,” Cluster Comput 24, 3249–3275 (2021). https://doi.org/10.1007/s10586-021-03310-7.

[10] He T, Toosi AN, Buyya R. “CAMIG: Concurrency-Aware Live Migration Management of Multiple Virtual Machines in SDN-enabled Clouds,” IEEE Transactions on Parallel and Distributed Systems. 2021 Dec 28.

[11] Shaw, Rachael, Enda Howley, and Enda Barrett. "An intelligent ensemble learning approach for energy-efficient and interference aware dynamic virtual machine consolidation," Simulation Modelling Practice and Theory 102 (2020): 101992.

[12] Anu, V. R., and Sherly Elizabeth. "IALM: Interference Aware Live Migration Strategy for Virtual Machines in Cloud Data Centres," Data Management, Analytics, and Innovation. Springer, Singapore, 2019. 499-511.

[13] Babu, KR Ramesh, and Philip Samuel. "Interference aware prediction mechanism for auto-scaling in the cloud," Computers & Electrical Engineering 69 (2018): 351-363.

[14] Yang, Chao, et al. "Interference-based VM Migration to Mitigate Cache-based Side-channel Attacks in Cloud." 2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE, 2018.

[15] Qin, Yiling, et al. "Interference and topology-aware VM live migrations in software-defined networks," 2019 IEEE 21st International Conference on High-Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2019.

[16] Tahir, Muhammad, et al. "CryptoGA: a cryptosystem based on genetic algorithm for cloud data security," Cluster Computing 24.2 (2021): 739-752.

[17] Denis, R., Madhubala, P. “Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems,” Multimed Tools Appl 80, 21165–21202 (2021). https://doi.org/10.1007/s11042-021-10723-4.

[18] Balakrishna, S.K., Shetty, S.M., Martis, J.E., Ramasamy, B. (2022). “Genetic Algorithm-Based Pseudo-Random Number Generation for Cloud Security.,” In Misra, S., Kumar Tyagi, A., Piuri, V., Garg, L. (eds) Artificial Intelligence for Cloud and Edge Computing. Internet of Things. Springer, Cham. https://doi.org/10.1007/978-3-030-80821-1_10.

[19] B.V. Natasha, Ram Mohana Reddy Guddeti, “Adopting elitism-based Genetic Algorithm for minimizing multi-objective problems of IoT service placement in the fog computing environment,” Journal of Network and Computer Applications, Volume 178, 2021, 102972, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2020.102972.

[20] R. Raja Subramanian, V. Vasudevan, “A deep genetic algorithm for human activity recognition leveraging fog computing frameworks,” Journal of Visual Communication and Image Representation, Volume

77, 2021, 103132, ISSN 1047-3203, https://doi.org/10.1016/j.jvcir.2021.103132.

[21] Kodli, Shilpa, and Sujata Terdal. "Hybrid max-min genetic algorithm for load balancing and task scheduling in a cloud environment," Int J Intell Eng Syst. 14.1 (2021): 63-71.

[22] Lu, Jiawei, et al. "Optimal machine placement based on improved genetic algorithm in cloud computing," The Journal of Supercomputing 78.3 (2022): 3448-3476.

[23] Zhang, Binbin, Xiao Wang, and Hao Wang. "Virtual machine placement strategy using cluster-based genetic algorithm," Neurocomputing 428 (2021): 310-316.

[24] Xue, Xingsi, et al. "Optimizing ontology alignment through linkage learning on entity correspondences," Complexity 2021 (2021).

[25] Huda Ibrahim, Raafat O. Aburukba, Khaled El-Fakih, "An Integer Linear Programming model and Adaptive Genetic Algorithm approach to minimize the energy consumption of Cloud computing data centers," Computers & Electrical Engineering, Volume 67,2018, Pages 551-565, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2018.02.028.

[26] Sarfaraz M, Tahir M. "A parallel multi-objective genetic algorithm for scheduling scientific workflows in cloud computing," International Journal of Distributed Sensor Networks. August 2020. doi:10.1177/1550147720949142.

[27] Akbari, A.; Khonsari, A.; Ghoreyshi, S.M. "Thermal-Aware Virtual Machine Allocation for Heterogeneous Cloud Data Centers." Energies 2020, 13, 2880. https://doi.org/10.3390/en13112880.

[28] Jafarnejad Ghomi, Einollah, Amir Masood Rahmani, and Nooruldeen Nasih Qader. "Service load balancing, scheduling, and logistics optimization in cloud manufacturing by using genetic algorithm," Concurrency and Computation: Practice and Experience 31.20 (2019): e5329.

[29] Guerrero, C., Lera, I. & Juiz, C. "Genetic Algorithm for Multi-Objective Optimization of Container Allocation in Cloud Architecture," J Grid Computing 16, 113–135 (2018). https://doi.org/10.1007/s10723-017-9419-x.

[30] Zhou, Z., Li, F., Zhu, H. et al. "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments," Neural Comput & Applic 32, 1531–1541 (2020). https://doi.org/10.1007/s00521-019-04119-7.

[31] Madhusudhan H S, Satish Kumar T, S.M.F D Syed Mustapha, Punit Gupta, Rajan Prasad Tripathi, "Hybrid Approach for Resource Allocation in Cloud Infrastructure Using Random Forest and Genetic Algorithm", Scientific Programming, vol. 2021, Article ID 4924708, 10 pages, 2021. https://doi.org/10.1155/2021/4924708.

[32] Miriam, A.J., Saminathan, R. & Chakaravarthi, S. Non-dominated Sorting Genetic Algorithm (NSGA-III) for effective resource allocation in the cloud. Evol. Intel. 14, 759–765 (2021). https://doi.org/10.1007/s12065-020-00436-2.

[33] Bloch, T., Sridaran, R., Prashanth, C. S. R. (2018) Understanding Live Migration Techniques Intended for Resource Interference Minimization in Virtualized Cloud Environment. Big Data Analytics. 487-497.

[34] Introduction to Genetic algorithm http://www.obitko.com/tutorials/genetic-algorithms/crossover-mutation.php.

[35] Chen Ming; Li Mengkun; Cai Fuqin; "A Model of Scheduling Optimizing for Cloud Computing Resource Services Based on Buffer-pool Agent," Granular Computing (GrC), 2010IEEE International Conference, Aug. 2010.

[36] Rajkumar Buyya, "explaining Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," http://dsl.cs.uchicago.edu/TPDS_MTC/papers/TPDSSI-2010-01-0012.pdf.

[37] Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R.. "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience 2011;41(1):23–50.

[38] Neha Garg, Damanpreet Singh, Major Singh Goraya, "VM Selection and Allocation Policy to Optimize VM Migration in Cloud Environment," International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-2, July 2019.

[39] Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. Evol Comput 4(1):1–32.

[40] Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." Multimedia Tools and Applications 80.5 (2021): 8091-8126.