A Synergistic Game Theory-Genetic Approach in VM Migration for Cloud Environment: Enhancing Scalability and Migration Decisions

MSc Research Project
Cloud Computing

Sheffy Batra
Student ID: x22121790

School of Computing
National College of Ireland

Supervisor: Dr. Punit Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sheffy Batra
Student ID:	x22121790
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Punit Gupta
Submission Due Date:	14/12/2023
Project Title:	A Synergistic Game Theory-Genetic Approach in VM Migration for Cloud Environment: Enhancing Scalability and Migration Decisions
Word Count:	6088
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	28th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Synergistic Game Theory-Genetic Approach in VM Migration for Cloud Environment: Enhancing Scalability and Migration Decisions

Sheffy Batra
x22121790

Abstract

With the growing cloud-oriented era of technology, virtual machine (VM) migration has become extremely common. There are n-number of VM migration strategies that are studied for optimal VM allocation, however, these strategies still suffer in the cloud computing model owing to inappropriate energy utilization, increased number of VM Migration, and execution time. The game-genetic Algorithm is proposed to find the optimal solution. It combines the concepts of Genetic algorithm with the Game theory algorithm for VM Migration. The proposed solution aims to optimize the allocation of VM which reduces the number of VM migrations, which in turn reduces the execution time and energy consumption. The efficiency of the proposed hybrid solution compared to the existing algorithm is 23.5% in terms of energy consumption, 16% in terms of the number of VM Migration, and 1% improvement in terms of execution time. With the same number of Resources, the proposed algorithm provided 24.80% efficiency in energy, a 31.77% reduction in the number of migrations compared to the genetic, IQMC, and MADMC algorithms.

1 Introduction

Cloud computing has changed the Information Technology world in recent years by making it easier to scale, react, and use resources in ways that were not possible before. With today's cloud-oriented environment, large cloud provider giants like Google, Microsoft, Amazon, IBM, etc. have built their data centers and distributed them geographically for wider bandwidth and expand their services worldwide aiming to improve performance and maintenance cost Greenberg et al. (2009). These big cloud providers target to provide various Cloud services to millions of users of various organizations that use cloud services for supporting their business needs either as public or private cloud Zhang et al. (2013). Cloud environment basically consists of an enormous number of data centers(collection of storage, CPU, and memory) which are being utilized by the users/organization for their needs with pay-per-usage policy or on the basis of agreement between two parties. Virtual machines play an important role through which virtualization of resources is done and services are provided to users Morabito and Beijar (2016). An important part of the cloud environment is managing and assigning Virtual Machines (VMs) on the fly across huge data centers. Load balancing, fault tolerance, scalability, and overall system performance all face substantial hurdles due to dynamic VM allocation, which is still critical for resource optimization.

The Figure 1 shows the example of a cloud computing environment.

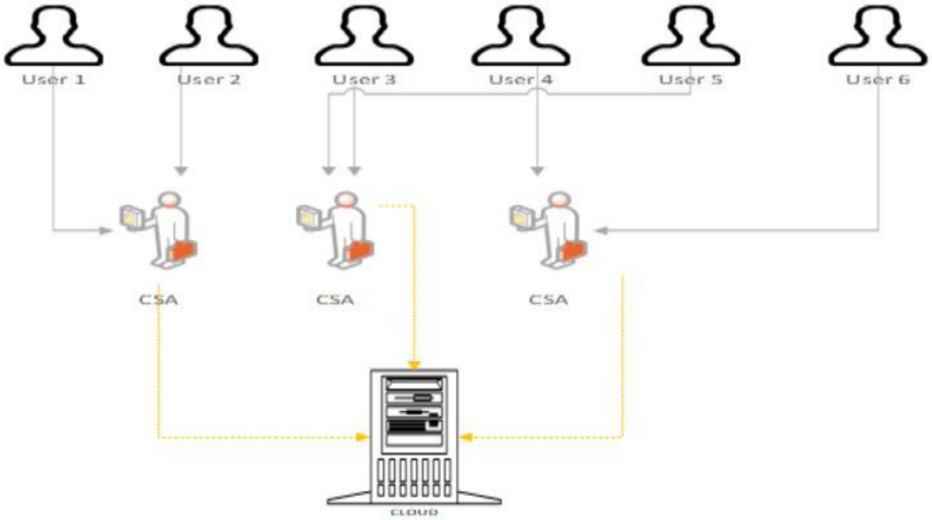

Figure 1: Cloud Computing (*Source:Kaur et al. (2023))*

Load in the cloud is balanced with the help of Virtual Machines. A request can be handled by one or more VMs. In a cloud computing environment, Dynamic workloads are the area that still needs to be figured out. The performance of Cloud services is majorly affected by the allocation of Virtual Machines to the Host. This is because of improper distribution of VMs of hosts since some of the hosts are overutilized and some are underutilized. This is the scenario when Virtual Machine Migration comes into the picture. The necessitate of virtual machine migration grows as more and more companies implement large-scale virtualized systems and depend on cloud-based services. VM Migration is moving VMs from one host to another to balance the load on all the hosts for efficiency and performance of systems using various techniques(Live or Non Live) Morabito and Beijar (2016).

TheFigure 2 represents the overview of VM Migration.

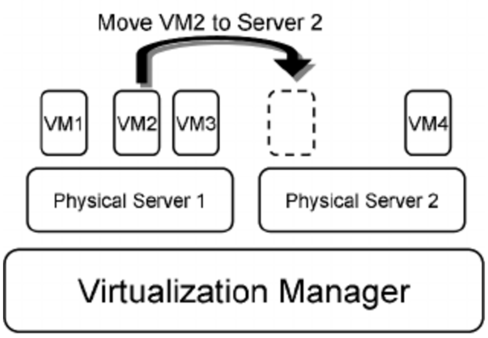

Figure 2: Overview of VM Migration

There are a number of challenges and parameters taken into consideration when going for VM migration. Determining the selection criteria for moving Virtual Machines (VMs) is one aspect of the difficulty. It is clear that in a cloud computing platform or system, a virtual machine's (VM) communication time with another VM on a different physical node takes a lot longer than it does inside the same physical node because the data transfer via random memory is quicker than over the network. Thus, communication costs need to be taken into account when determining whether to move a virtual machine (VM) from a Host that is overutilized/overloaded. Another factor taken into consideration is to choose underutilized Hosts to migrate VMs. When several VMs need to be migrated and various underutilized Hosts are feasible for migration, the primary aim becomes to reduce both migrations as well as scalability of VM Migration.

Various researchers have performed their own investigations to improve the efficiency, cost, power, and network to improve the VM Migration. Numerous Meta-heuristic algorithms such as Ant Colony Optimization(ACO), Genetic Algorithm, Live Migration, and Machine learning algorithms have been explored for Virtual Machine Migration. This research paper aims to explore the concepts of game theory to improve the Migration and scalability in VM Migration.

1.1 Research Problem

Various huge companies and techs have moved to cloud infrastructure and want to utilize the cloud services to benefit their needs of dynamic loads which requires an increase in VM Migration to balance their dynamic workloads. To meet their extensive dynamic workloads the migrations have caused various expenditures in terms of cost, energy, SLA, and performance. Huge Cloud Providers such as AWS, and Google have various resource services but could not provide better performances even with increased resources. There are increase in many VM Migrations, and execution time which increases energy consumption and leaves no room to scale the VM migration in case of an increase in demand. In the era of Green computing, reduction in energy consumption is always taken as a parameter in addition to other parameters to find a solution. As a result, it is imperative to create a solution that will take less execution time, less number of VM Migration, and eventually less energy consumption and have the capability to handle more number of VM migrations without compromising the stability and performance of the System.

1.2 Research Question

From the Research Problem, the main goal of this research is to answer below Question:

1. *How can hybrid solution of Game Theory-genetic algorithm be efficiently leveraged to enhance scalability by minimizing the number of VM migrations?*

2. *How can the principles and methodologies of Game theory and Genetic algorithms be employed to optimize energy consumption and migration decisions for reduction migration time within a cloud environment?*

The main goal of this research is to decrease the number of VM Migration and execution time which in turn decreases the energy consumption. The research aims to provide a solution that will work efficiently even if the VM migrations are scaled up with less degradation in performance and SLA and proves to be a scalable solution for VM Migration. Simulators are used to carry out testing for the solution before implementing the solution in a real-time environment. The objective can be achieved by efficiently making the decision to move a VM from one host to host to find the best solution for that VM. This can be achieved with proper load balancing and VM allocation policies that will make the best decision. For addressing this Hybrid approach for Game theory and Genetic algorithm is explored in this research paper.

The research paper is organized as follows: Section 2 provides a review of related work and contributions in this domain. Section 3 introduces the proposed model, focusing on the proposed Game-Genetic algorithm for VM migration. Section 4 defines the Design architecture of the system. Section 5 consists of the Implementation and various configurations of the project. The evaluation of our algorithm is outlined in Section 6. Section 7 discusses the implications of our work. Finally, Section 7 concludes and discusses about the future work.

2 Related Work

As Technology is becoming more cloud-oriented, using cloud services, Virtual machine (VM) migration has become crucial for cloud providers and customers to support their business needs of scalability, performance, availability, and optimal resource use inside modern data centers. Various studies and surveys by Choudhary et al. (2017) have been done to discover various issues and techniques for Virtual Migration. In spite of this, The survey by Zhang et al. (2018) that depicts that there is still a notable area of customer perspective and needs-specific service delivery in particular with regard to the impact of customer migration overall and rising demand and workloads that eventually impair productivity and performance by creating a shortage of resources.

2.1 VM Migration in Cloud Computing

The study by Li and Cao (2023) proposes a novel strategy for increasing the efficiency and reliability of online migration for Virtual Machines (VMs) running memory write-intensive workloads. To address concerns related to pre-copy migration, the proposed technique employs a parallel transmission system with two channels and receiver threads. Furthermore, it provides an adaptive hybrid migration strategy for smooth migration completion, improving the switching time based on an iterative convergence factor. Further research is required to include Wide Area Network (WAN) situations.

The author Arif et al. (2016) proposed a live migration technique with the aim to increase the impact of VM migration over WAN's because over WAN due to network traffic it is very difficult to move the large workloads, named machine learning and prediction-based (MLDO) adaptive live migration technique. Although the study was able to achieve overall Migration time and downtime objectives using 2-tier decision tree but lacks scalability in this method.

2.2 VM Migration using Optimization Algorithms

Singh and Singh (2023) proposed a novel approach to VM Performance-Aware VMM method (PAVMM) based on the Ant colony optimization Algorithm which aims to optimize the performance and cost of Migration. This proposed model played major role in boosting efficiency and optimizing resource allocation inside data centers. The proposed solution worked out with good results but it does not consider the other metrics like service level agreements (SLAs), and quality of service (QoS) and Energy consumption. On the other hand, Zhao et al. (2023) made a proposal for the gap above by proposing an energy-aware methodology based on a bio-inspired optimization algorithm named R&DWOA algorithm which aims to minimize the energy consumption in VM Migrations. It was able to achieve a harmonic balance between energy use, migration costs, and resource availability. It shows 24.80% efficiency in energy, 31.77% reduction in migration cost over traditional chic algorithm. However, this approach falls back in case when it comes to SLA's which created a major impact for this algorithm where SLA is an important factor.

Virtual Machine (VM) migration is important for optimal resource management in cloud computing. Research by Khan and Santhosh (2022) proposed a hybrid optimization algorithm that combines Cuckoo search and particle swarm optimization,which aims to provide a solution for minimizing energy consumption, computation time, and migration costs. The solution works well as compared to the traditional algorithms but the hybrid model has not been tested across different cloud environments and dynamic workload conditions. Kansal and Chana (2016) had reported in their findings of the hybrid optimization, that they had integrated a lion optimization algorithm with a whale optimization algorithm. This was done to identify a solution for the migration issues in VM's. In this approach the fitness function was calculated for the hybrid model and was used to regulate the resource utilization, energy consumption and migration costs. . The average of 44.39 % energy consumption is by reduced as compared to other algorithms and also a reduction of 72.34 % of migrations and saving 34.36 % of hosts.

Research done by Najm and Tamarapalli (2022) focuses on improving cost-aware Virtual Machine (VM) migration in federated clouds, with the goal of reducing migration time and cost. The method, which incorporates VM characteristics and network factors, produces significant reductions in total cost of operation, migration time, downtime, and overall migrations, demonstrating its efficacy and practical usefulness. They proposed cost-aware VM Migration Algorithm (CAVMA) with reduced the TCO (total cost)by 48%.However this approach does not take into consideration the communication cost between the VM's and scalability which leaves the room for further research

2.3 Scalabilty in VM Migration

The research on resource utilization in Virtual Machine Migration was deeply studied by Shahapure and Jayarekha (2018b) . The study focuses on how to improve resource utilization. The approach they followed focused on achieving green computing with low energy usage and traffic within the virtual machines. They were able to achieve the desired results but their study does not take the latency into account between clients and cloud servers which leads to huge latency in service times. In order to overcome this gap, another research by Shahapure and Jayarekha (2018a) proposed a Distance and Traffic (VMMADT) Algorithm for Virtual machine Migration which focuses on latency from the client's perspective. In this approach, the Algorithm runs iteratively to check on the network and distance with the client request and data center, and the nearest one to a client is selected for the request. But this approach could not stand with the efficient use of resource utilization and energy consumption

Scalability is an important aspect of VM migration in cloud systems. A large number of research as been done to improve the scalability of VM migration. One of the studies by Canali and Lancellotti (2014) uses the principle component analysis (PCA) to deal with scalability in VM. This proposed methodology sought to significantly decrease computing costs and resource requirements. This method provided a realistic alternative for dealing with scalability concerns in cloud computing systems. Similarly, Hummaida et al. (2022) used reinforcement learning (RL) Methodology for scalability in decentralized cloud infrastructure. The primary goal of this research was to improve QoS and resource allocation in large data centers. Although the preceding study focuses on improving scalability, there are some literature gaps that must be filled. The RL-based technique has shown to be effective in terms of centralized management and VM migration optimization, but there are still other hurdles to overcome, such as deployment, training on real systems, and dealing with variable workloads.

2.4 Game Theory in Cloud Computing

The potential of cooperative game theory to describe and analyze interactions and decision-making processes in complex and large-scale systems has piqued the attention of cloud computing experts. Approaches based on game theory offer a foundation for enhancing resource allocation, SLA's, energy utilization, system performance, and overall system efficiency.

The study by Ghosh and De (2019) used game theory concepts to increase the efficiency and fairness of Virtual Machines(VM) in a cloud environment. The study uses cooperative game theory for decision-making with the objective of decreasing migration time and increasing the migration options. Similarly, a cooperative game solution was proposed by Agbaje et al. (2022) for resource distribution. The author worked on workloads, SLA's and cost with objective of binding the cloud service providers and share resources cooperatively.

In cloud computing, a Q-learning technique named Reinforcement learning was explored by Tran et al. (2022) for multi-tier applications for VM Migration. The approach conceptualized the issue as a non-cooperative game, wherein individual physical machines (PMs) act as players striving to optimize their utility through efficient load and resource utilization management.

In the comparison of the above works on game theory by authors, Ghosh and De (2019) and Agbaje et al. (2022) both uses the Cooperative game theory concept whereas the first focuses on a single cloud while the later focuses on a federated cloud. While Tran et al. (2022) uses non cooperative theory approach for multi-tier application in VM migration

Although several optimizations have been performed using game theory, scalability remains the main problem in the preceding research. scalability is still a big challenge in VM migration. VM migration will be boosted by research in this area. To increase overall VM performance, a solution should be devised that incorporates scalability as a component with the previously described improvements.

2.5 Research Niche

The work done by various authors on VM migrations using various techniques and with different objectives to improve VM Migration by using hybrid solutions has been examined in the above section. Various algorithms like nature-inspired, ant colony, whale optimization, and other hybrid techniques are used with the aim of improving various parameters such as energy, cost, SLA, communication, and migration time. There has been work done on the scalability of VM migration by improving the SLA. However, focusing on the number of VM Migrations can improve the scalability as well as stability of VM Migrations. which is still a gap in the above research done by various authors. Scalability is an important factor to meet the growing dynamic demand for business. So to improve this factor, this research paper aims to improve the number of VM Migrations and execution time to improve the scalability by using a hybrid approach of Game theory strategies and Genetic algorithm.

3 Methodology

In the Related work, various methods were examined which were used by various researchers to deal with Virtual Machine Migration such as ACO, PSO, hybrid approach, Genetic, and other meta-heuristic algorithms as well as Machine Learning algorithms that aim to have multiple objectives like power, SLA, cost, communication. The survey on various techniques in one paper by Imran et al. (2022) gives the overview of all the methods used for various parameters in Virtual Machine Migration. In this paper, a new approach is proposed a hybrid approach using strategies from game theory and genetic algorithm. The proposed algorithm aims to apply game theory strategies during the crossover and mutation phase of the Genetic Algorithm. The proposed methodology aims to improve the power, execution time, and number of VM Migrations which is important for the scalability of VM migration.

The aim is to:

1. Balance the load between the overutilized and underutilized Hosts.
2. Adjust the load with the proper utilization of resources to shut down the host and save the resources for further scalability.
3. Reduce the Number of VM Migrations
4. Reduce the energy consumption for scalability.
5. Reduce the execution time.

The Process of VM Migration is triggered when the sub-optimal performance of the Hosts is detected. At the same point of time "a" VM's would need to be moved to "b" optimal Hosts within the cloud infrastructure.

Let us consider Cloud infrastructure has "a" VMs and "b" Hosts. When an overutilized host is detected, VM Migration is initiated. Lets define how it will work:

3.0.1 *Equation 1: Migration Decision equation*

The decision of migration and allocating a VMs to b Hosts can be presented by a below matrix $P(q \times b)$:

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1b} \\ p_{21} & p_{22} & \dots & p_{2b} \\ \vdots & \vdots & \ddots & \vdots \\ p_{a1} & p_{a2} & \dots & p_{ab} \end{bmatrix} \tag{1}$$

where $p_{ab} \in \{0, 1\}$,

The migration can be described as if $p_{ab} = 1$; Migration else No migration, $p_{ab} = 0$.

3.0.2 *Equation 2: Decision of allocation*

Based on equation 1, A potential assignment of x types of resources in the jth host can be represented as an allocation matrix $Q(j)(a \times x)$:

$$Q(j) = \begin{bmatrix} v(j)_{11} & v(j)_{12} & \dots & v(j)_{1x} \\ v(j)_{21} & v(j)_{22} & \dots & v(j)_{2x} \\ \vdots & \vdots & \ddots & \vdots \\ v(j)_{a1} & v(j)_{a2} & \dots & v(j)_{ax} \end{bmatrix} \tag{2}$$

where $v(j)_{ax} \in \mathbb{Z}^+$ Indicating the quantity of resource type x allocated from the jth Host to Virtual Machine (VM) a, the matrix $Q(j)(a \times x)$ encapsulates this allocation. A vector $Q = \{Q(1), Q(2), \dots, Q(b)\}$ outlines a plausible resource allocation strategy for all Host. The optimal VM migration challenge is framed as a trade-off scenario, navigating the balance between load distribution and resource utilization through the lens of non-cooperative game theory.(Source Xu and Yu (2014)

3.1 VM migration in Game theory

Xu and Yu (2014) takes a game theory approach to VM migration, with the twin goals of preserving load balance and avoiding resource waste on Hosts. Game theory, a mathematical study of strategic interactions, dictates how individuals behave in order to maximize their overall individual results or minimize their overall individual losses. The VM migration quandary is modeled as a non-cooperative game, with the hosts acting as individual players in this strategic situation.Below equations from Xu and Yu (2014)

3.1.1 *Equation 3: Game of VM Migration*

A game of VM migration can be presented as a vector of 3 tuple
$M = (P, (S(j))_{j \in P}, (g(j))_{j \in P})$.

1. P is the finite set of players in the game, i.e., $P = \{1, 2, \dots, h\}$.
2. $S(j)$ is the set of available strategies for player j.
3. $g(j) : H \to R$ is the utility function for player j.

The ultimate purpose of the migration game in the experiment is to attain load balance, with each player seeking to minimize their particular resource loss. The utility function for the (j)-th player is written as follows to achieve load equilibrium and maximize resource utilization:

$$g(j)(H) = \frac{1}{H(j) + L}$$

The Nash equilibrium in the game happens when no player can improve its utility by changing its strategy while the other players' tactics remain unchanged. In layman's words, the Nash equilibrium is a collection of tactics in which participants have no reason to change their behavior. Each element $\alpha(a) \in M(a)$ serves as the strategy for player a, $\alpha(-a) = [\alpha(j)]_{j \in P, j \neq a}$ describes the strategies of all players except a, and $\alpha = (\alpha(a), \alpha(-a))$ is indicated as a strategy profile.

3.1.2 *Equation 4: Nash Equilibrium*

A profile α^* qualifies as a Nash equilibrium for M if and only if each player's strategy is an optimal response to the strategies adopted by the other players, expressed as

$$\alpha(a)^* \in \mathrm{br}(a)(\alpha(-a)^*) \quad \text{for every player } a$$

where $\alpha(-a)$ signifies the strategies of all players except a, $\mathrm{br}(a)$ denotes the best response of player a, and

$$\mathrm{br}(a)(\alpha(-a)^*) = \alpha(a)^* \in \alpha \mid g(a)(\alpha(a)^*, \alpha(-a)^*) \geq g(a)(\alpha(a), \alpha(-a)^*)$$

Introducing the set-value function $\mathrm{br} : \alpha \to \alpha$ defined by $\mathrm{br}(\alpha(i)) = \times_{i \in P} \mathrm{br}(\alpha(-i))$, equation (3) can be recast in vector form as $\alpha^* \in \mathrm{br}(\alpha^*)$. The existence of α^* satisfying $\alpha^* \in \mathrm{br}(\alpha^*)$ is established through the application of Fixed Point theorems.

3.2 Genetic Algorithm in VM Migration

Muhammad et al. (2017) have used a Genetic algorithm for VM Migration and improved resource utilization and Energy consumption. The procedure used in the paper is defined in the below pseudo code Figure 3.

3.3 Proposed Algorithm

The new proposed optimization algorithm takes into consideration the Game theory and genetic algorithm strategies. The overall flow is shown in below Figure 4. Game theory concepts explained by authors Xu and Yu (2014) used for Resource allocation have been applied with Genetic algorithm instead of random formation. The Genetic Algorithm basically consists of Initialization, Selection, crossover, mutation, and allocation phases as described by Muhammad et al. (2017). In the proposed algorithm, the game theory equations discussed in section 3.0.1, 3.0.2, 3.1.1, 3.1.2 are applied in the crossover phase of Gentic algorithm

Instead of randomly doing a crossover, nash equilibrium 3.1.2 is being used, on the basis of which crossover will be performed. Consider the "n" number of VM's in a tournament $T = \{1, 2, \ldots, n\}$ that are selected for crossover($\imath$) be the index of the VM under consideration. The Objective of the Game is to reduce the number of VM Migration with effective load balancing and reduce execution time.

1. **Crossover:** Crossover is defined as

```
Begin
Initialize P(0); // P is physical host population
int T; // T is evolutionary algebra
t = 0;
while(t <= T) do
    findRandomHost();    //randomly selects n physical hosts as initial population
    for i = 1 to M  do     //M is the number  of individuals of the initial population
    Evaluate fitness of P(t);  //Calculate the fitness of each individual in P(t)
    end for
        for i = 1 to M  do
        Select operation to P(t);  //Apply the selection operator to the group
       end for
       for i = 1 to M/2  do
        Crossover operation to P(t); //Apply crossover to population
       end for
       for i = 1 to M  do
        Mutation operation to P(t);  //Apply the mutation operation to the group
       end for
       for i = 1 to M  do
        P(t+1) = P(t);     //Get the next generation group P(t+1)
       end for
        t = t + 1;     //End condition judgement
       end while
        AllocationVMtoHost();     //Assigning a virtual machine to the most adaptable
physical host
    End
```

Figure 3: Genetic Algorithm by Muhammad et al. (2017)

$$\text{offspring1}[i] = \begin{cases} \text{parent2}[i] & \text{with probability Probabilitycrossover} \\ \text{parent1}[i] & \text{otherwise} \end{cases}$$

$$\text{offspring2}[i] = \begin{cases} \text{parent1}[i] & \text{with probability Probabilitycrossover} \\ \text{parent2}[i] & \text{otherwise} \end{cases}$$

Probabilty of Crossover The Crossover strategy is defined as A= 0.2, B= 0.8

$$\text{Probabilitycrossover} = A \times \text{loadBalFactor} + B \times \text{neigInflParamter}$$

$$\text{loadBalFactor} = \begin{cases} A & \text{if load difference is less than a threshold} \\ B & \text{otherwise} \end{cases}$$

$$\text{neigInflParameter} = \begin{cases} A & \text{if neighbor similarity is greater than a threshold} \\ B & \text{otherwise} \end{cases}$$

$$\text{neigSimilarity} = \frac{\text{commonNeighbors}}{\text{NEIGHBORCOUNT}}$$

where commonNeighbors is the number of common neighbors for the given VM

After Crossover and Mutation phase Nash equilibrium strategy is defined to sort and find the best optimal solution for the VM based on the below strategy of neighbouring VMs

For each VM i, compare the payoffs of its left ($leftNeighborIndex$) and right ($rightNeighborIndex$) neighbors with its current payoff. Update the allocation of VM i based on the comparison.

currentPayoff=CPOF,payoffFunction= POF,newPayoffLeft= NPOFL,newPayoffRight= NPOFR

$$\text{CPOF} = \text{POF}(\text{allocationPopulation}[i])$$

$$\text{NPOFL} = \text{POF}(\text{allocationPopulation}[leftNeighborIndex])$$

$$\text{NPOFR} = \text{NPOF}(\text{allocationPopulation}[rightNeighborIndex])$$

$$\text{allocationPopulation}[i] = \begin{cases} \text{leftVMAllocation} & \text{if NPOFL} > \text{CPOF} \\ \quad \text{and NPOFL} > \text{NOPFR} & \\ \text{rightVMAllocation} & \text{if NPOFR} > \text{CPOF} \\ \quad \text{and NPOFR} > \text{NPOFL} & \\ \text{allocationPopulation}[i] & \text{otherwise} \end{cases}$$

In this mathematical model of nash equilibrium, each VM evaluates its left and right neighbors' payoffs and modifies its allocation appropriately.

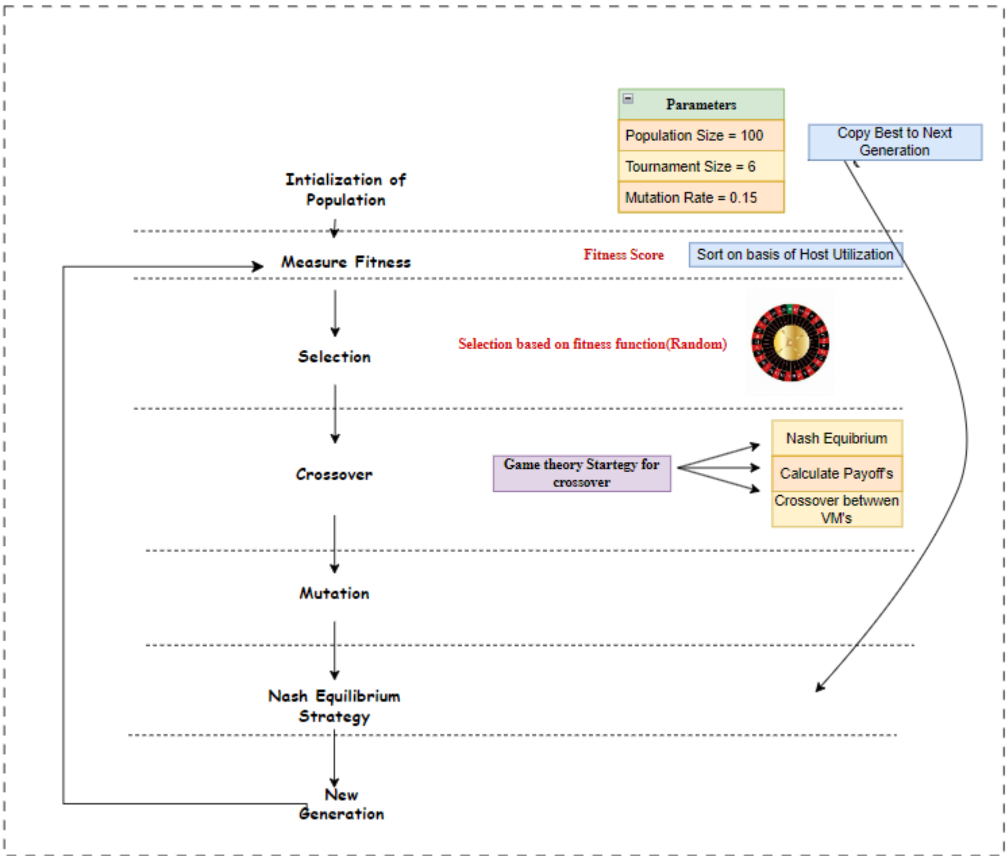

Figure 4: Proposed Game-Genetic Algorithm

4 Design Specification

An cloud Computing environment, the infrastructure is made of data center, and is the target system. The data center is made up of a maximum of n heterogeneous hosts, each of which has several VMs. Virtual Machine Monitor (VMM) is used to assign several VMs to each host. Furthermore, each host and VM are distinguished by CPU performance parameters expressed in Millions of Instructions Per Second (MIPS), RAM capacity, and network bandwidth.The system model is made up of a global and a local manager. Users input their requirements for M heterogeneous VM provisioning.Local Managers exist on each node and are responsible for monitoring a node's CPU consumption, resizing the VM with their resource demands, and making decisions of when and which VMs must be relocated from the host continuously. Resource utilization is taken care by global Manager from Master Node.They communicate with local Mangers about the status of VM's and Host.The global manager issues orders to optimize the VM location. VMMs do real resizing, VM migration, and node power state changes. The System architecture is shown in Figure 5

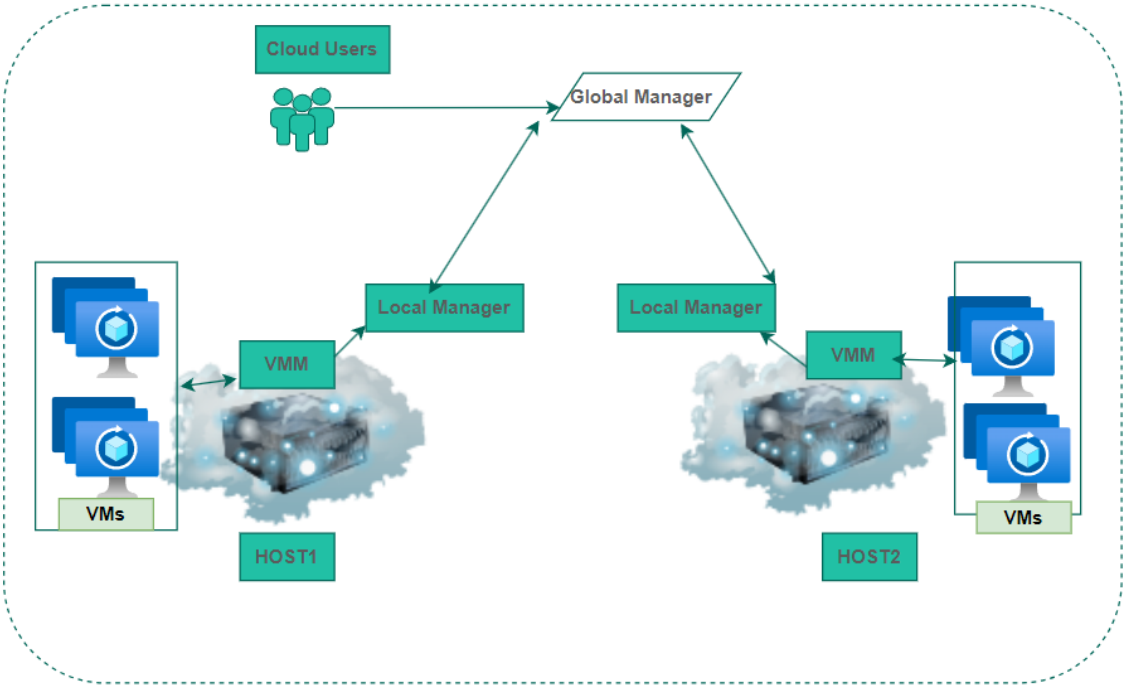

Figure 5: System Design

5 Implementation

Because of its excellent capabilities in simulating cloud computing systems, the CloudSim framework is always a preferable choice. CloudSim is a virtualization platform that simulates the behavior of real-world cloud infrastructures, offering a safe and realistic environment for testing and assessing the proposed Virtual Machine (VM) migration solution.Calheiros et al. (2010)

Cloudsim is a specially designed simulator for simulating the real-world scenario for Cloud computing before testing in real implementation. It is a great tool used by researchers which paves a great way to verify their research results. It is a great tool for simulating VM Migration scenarios with various workloads and studies how the proposed solution will work in real-world scenarios.

Table 6 and 1 contains the hardware and parameters of Simulations

Data Center	No of Data Centers	4
Virtual Machine	No. of VMs	50-400
	PES	1, 1, 1, 1
	MIPS	2500, 2000, 1000, 1500
	RAM	870, 1740, 1740, 613
	Bandwidth	100MBit/s
	Storage	2.5GB
Hosts	Number of Hosts	50-400
	MIPS per host	1860, 2660
	RAM	4096, 4096
	Bandwidth	1 Git/s
	Storage	1GB

Figure 6: Cloudsim Parameters

Table 1: Hardware and Software Configuration

Machine	HP laptop 5th Generation
OS	Windows 11
RAM	16 GB
Processor	AMD Ryzen 5 5625U with Radeon Graphics, 2.30 GHz
IDE	Eclipse
Simulator	CloudSim 3.0

The Diagram 7 shows the overall view of the objective of the proposed algorithm and the ultimate goal to reach the optimal solution. The figure shows the cloud environment with underutilized and overutilized hosts. So the Major objective is to find the solution so that after minimum migrations the load is balanced and some of the hosts that are not required are shut down for further scenarios when there is a scaling up of the system.

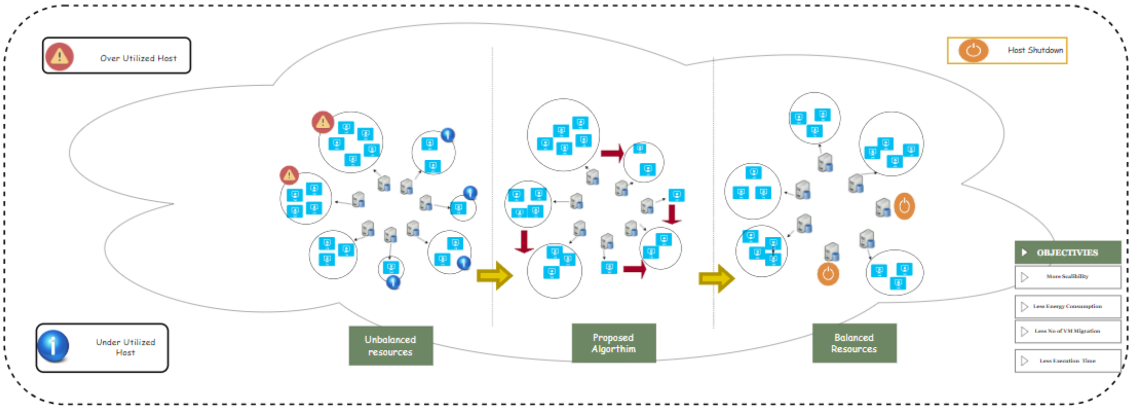

Figure 7: Objective of Proposed Game-Genetic Algorithm

6 Evaluation

In cloud computing, the efficient allocation of virtual machines (VMs) and their migration play a crucial role in enhancing Quality of Service (QoS) standards. This paper focuses on achieving an optimal solution for VM allocations and migrations, considering the following :

Now, let's consider a scenario with m Hosts and n virtual machines. The Host set $H = \{H_1, H_2, \ldots, H_m\}$ and the VM set $V = \{V_1, V_2, \ldots, V_n\}$.

Total Completion Time (TCT) for a Migration:

The total completion time for the n and m number of Host and VM is given:
Migration time = Migration complete - Migration start

Objective Function:

The objective function aims to minimize the energy consumption,number of VM Migration and execution time.

The output of the algorithm is represented in an allocation matrix (XM), indicating which VM is allocated to which Host:

$$XM = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

where $x_{ij} = \begin{cases} 1 & \text{if } V_i \text{ is allocated to } H_j \\ 0 & \text{if } V_i \text{ is not allocated to } H_j \end{cases}$

This approach considers both VM allocations and migrations to achieve an optimized solution based on the defined metrics. The proposed Game-Gentic optimization algorithm performance is the measured using simulation Tool. To validate the performance of the proposed model results are compared to the existing VM Migration algorithms Genetic algorithm, IQMC, and MAD. The traditional Model have gaps in terms of No of VM migrations, power consumption, computation time, and SLA. The proposed model tried to overcome similar limitations and the same has been verified with simulation results.

The below experiments are done with 50-400 VM's and Host.

6.1 Experiment / Case Study 1

The Figure 8 compares the energy consumption analyses for the proposed hybrid optimized solution with the current migration strategies. The suggested hybrid optimization consumes the least amount of energy for 33.29 Kwh for 50, 63.86Kwh for 100, and 127.16 kWh for 200 Host. In comparison, the suggested model consumes 10.74% less energy than the genetic Algorithm,28.90% less than IQMC and 26.95% less energy than the MAD algorithm. The Table 2 has the information of the energy consumption. This is one of the objective of the proposed algorithm which is tested and verified

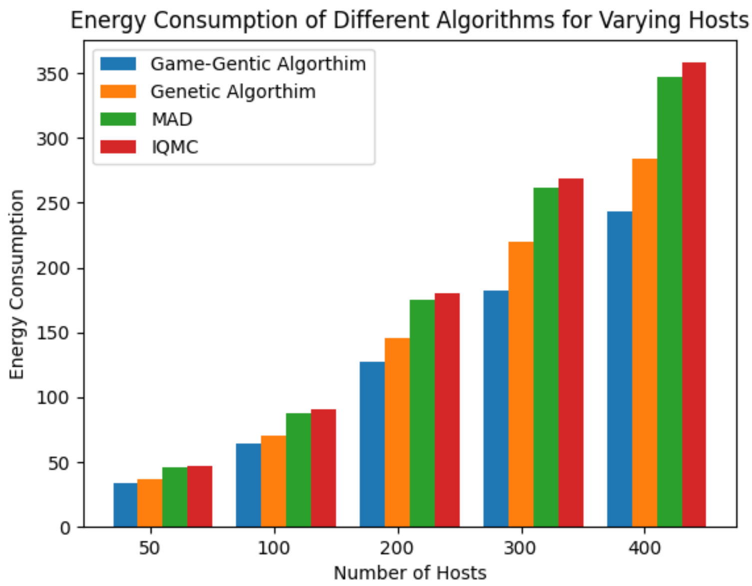

Figure 8: Energy Consumption in Kwh

Table 2: Comparison of Energy Consumption(kwh)

No of Host & VM's	Genetic	IQMC	MAD	**Proposed Game-Genetic**
50	37.30 kWh	46.86kWh	45.61kWh	**33.29 kWh**
100	69.91 kWh	90.97 kWh	88.06 kWh	**63.86 kWh**
200	145.58 kWh	179.84 kWh	174.72 kWh	**127.16 kWh**
300	220.39 kWh	268.75 kWh	261.33 kWh	**182.29 kWh**
400	283.58 kWh	358.05 kWh	347.38 kWh	**242.94 kWh**

6.2 Experiment / Case Study 2

The analysis of number of Migration for the new proposed model and the existing models is shown in Figure 9. The observations are based on the total number of real VMs and the number of VMs that have been moved to get the optimal allocation. The findings show that the suggested strategy achieves the least migration when compared to the current approaches. The optimal VM selection executes essential migrations while avoiding extraneous migration requests, resulting in fewer migrations and a more efficient migration than traditional migration techniques. The Table 3 depicts the number of VM migrations.This results in achieving more VM migrations if the system scale's UP with minimal energy consumption, thus providing an optimal scalable solution.

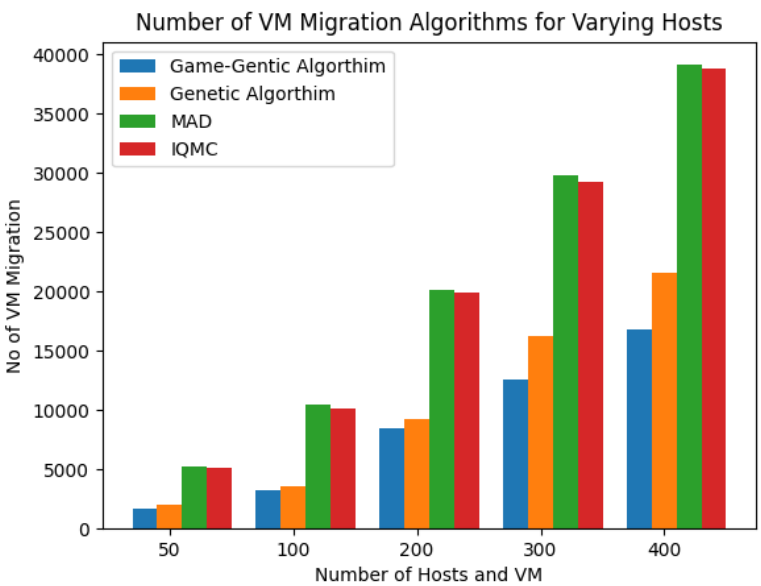

Figure 9: Comparision of No of VM Migration

Table 3: Comparison of Number of VM Migrations

No of Host and VM's	Genetic	IQR	MAD	Proposed Game-Genetic
50	2075	5085	5265	1707
100	3600	10153	10437	3201
200	9203	19939	20107	8469
300	16215	29265	29791	12581
400	21540	38857	39142	16866

6.3 Experiment / Case Study 3

The completed execution time for the approach proposed and for approaches that already exist are given in Figure 10 and Table 4. In IT the execution time is usually defined as the time consumed to migrate from one VM to another. According to the results it can be observed that the proposed hybrid Game Genetic algorithm is executed with the least computational time. Although the computational time is measured with minimum elements, in a real-world setup the computational time would vary from the optimal scores obtained in this project. But it can be assured that the total time for computation for the proposed hybrid optimization algorithm would be lesser than the already existing migration algorithms. .

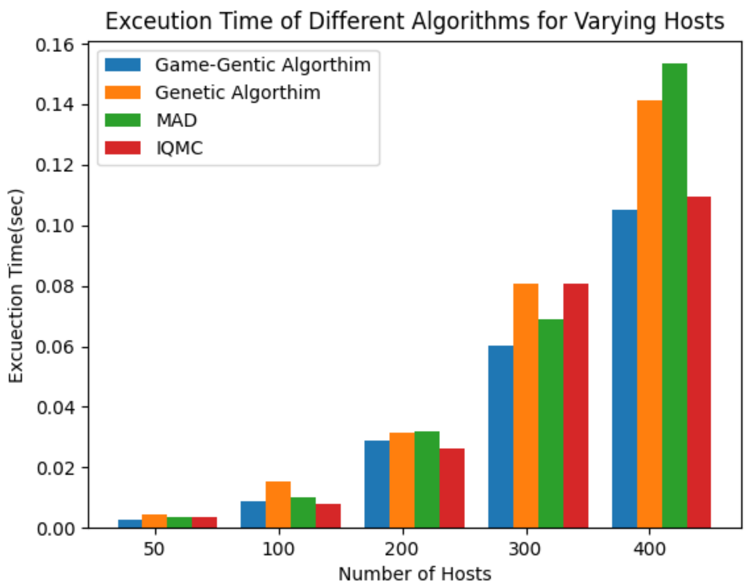

Figure 10: Excecution Time

6.4 Experiment / Case Study 4

The Figure 11 and Table 5 show that the standard deviation is relatively low and remains consistent for the proposed algorithm as compared to other algorithms which indicate that the proposed solution is more stable and has good performance even if the system size scales up. Thus the proposed algorithm is compatible for VM Migration scalability as compared to other algorithms.

Table 4: Comparison of Execution Times in Seconds

No of Host and VM's	Genetic	IQR	MAD	**Proposed Game-Genetic**
50	0.00436	0.00378	0.00379	**0.00290 sec**
100	0.01531	0.00998	0.00993	**0.00891 sec**
200	0.03137	0.03644	0.03193	**0.02886 sec**
300	0.08092	0.06192	0.06894	**0.06019 sec**
400	0.14116	0.11303	0.15341	**0.10507 sec**

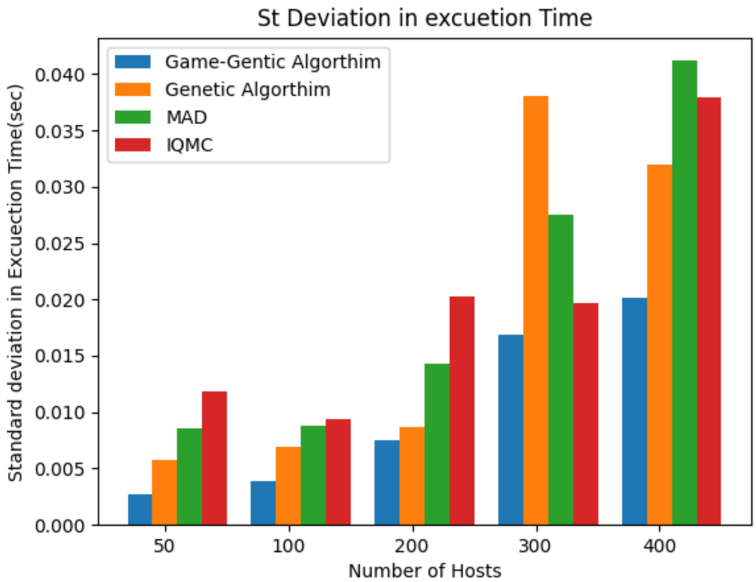

Figure 11: Standard deviation in Execution Time

Table 5: Comparison of St deviation Execution Times(sec)

No of Host and VM's	Genetic	IQR	MAD	Game-Genetic
50	0.00580 sec	0.01183 sec	0.00861 sec	0.00270 sec
100	0.00698 sec	0.00933 sec	0.00879 sec	0.00388 sec
200	0.00866 sec	0.02029 sec	0.01428 sec	0.00746 sec
300	0.03808 sec	0.01964 sec	0.02748 sec	0.01689 sec
400	0.03198 sec	0.03796 sec	0.04118 sec	0.02010 sec

6.5 Experiment / Case Study 5

The Figure 12 depicts that Game-Genetic algorithm consistently maintains a lower number of host shutdowns across different system sizes as compared to other algorithms which suggest that the algorithm offers better stability as compared to other algorithms. Game-genetic has 33% less host shutdown as compared to IQRC,32% as compared to MAD and 15% as compared to Genetic. The proposed algorithm consistently exhibits lower SLA performance degradation compared to other algorithms, which in turn has its effectiveness in minimizing the impact on SLA during migrations.

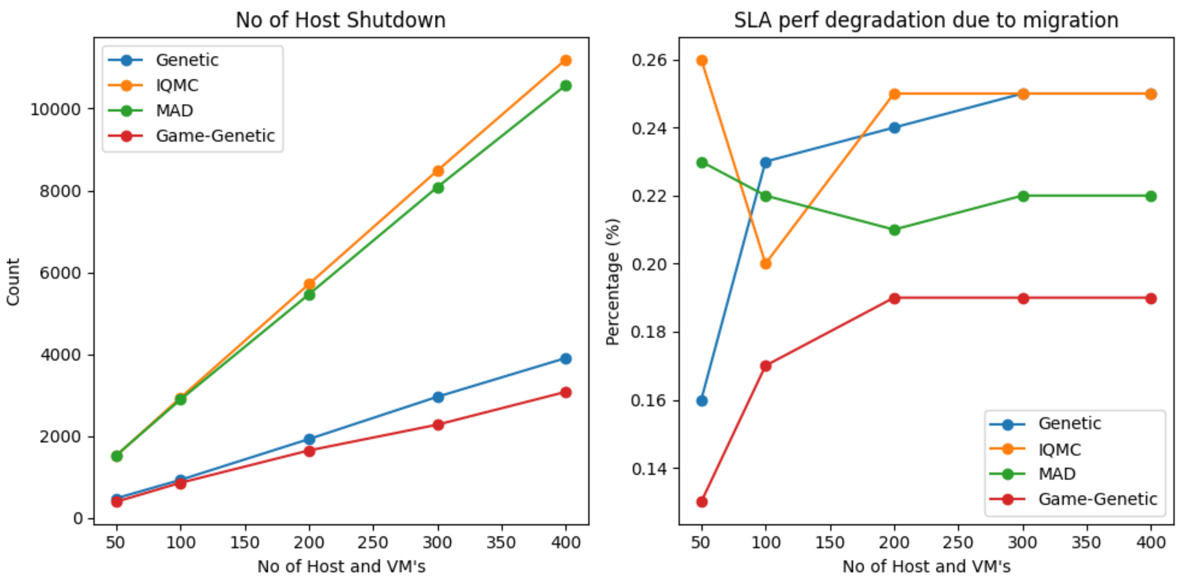

Figure 12: SLA degradation and Number of Host Shutdown

6.6 Discussion

The simulation results for the Proposed hybrid game-genetic algorithm and the other standard algorithms are compared and the proposed algorithm has proven to be more effective in case of reducing the energy consumption, number of VM Migration, and Execution Time. The reduced number Of VM Migration depicts the stability of the system and more scalability of the system in case of a Scale-up scenario. With the increasing scalability, the execution time and energy consumption are also less as compared to other algorithms. Performance degradation with increasing numbers is almost stable after a certain number which proves that the algorithm is efficient for large-scale systems. The standard deviation is less as compared to other algorithms which show that the proposed algorithm provides more stability to system performance over a large scale.

7 Conclusion and Future Work

There is a lot of potential for improving the effectiveness and efficiency of cloud resource management by developing an application on CloudSim that uses game theory to optimize virtual machine migration. According to the findings of this research, adopting game theory-Genetic Algorithm to decisions about the migration of virtual machines (VMs) has the potential to result in several significant benefits, including improved energy efficiency, decreased number of VM Migration, less execution time, and improved performance which has been proven in above evaluation section. The solution proves to be scalable for large VM Migrations. The fact that the proposed solution has been able to achieve speedy convergence to near-optimal solutions and drove typical migration procedures is evidence that the solution has the potential to be implemented in the real world utilizing cloud computing. In spite of this, it is of the utmost importance to pinpoint the constraints and challenges that have been encountered, since doing so reveals key insights that may be used to improve the creation of the next enhancements and adjustments.

Because of the solution's scalability, it is appropriate for deployment into cloud systems that are both more comprehensive and sophisticated. As cloud computing technology continues to grow at a breakneck speed, this study will act as a springboard for more investigation. It will be essential to conduct further research into game-theoretic models and investigate novel optimization targets that take communication factor and improve the SLA to keep up with the ever-changing needs of the rapidly growing cloud computing landscape. Future research needs to be done to explore solutions in heterogeneous cloud environments with dynamic workloads to check the scalability of the proposed algorithm. Other strategies of game theory can also be explored in depth to understand how the different strategies would impact the overall algorithm's performance.

References

Agbaje, M. O., Ohwo, O. B., Ayanwola, T. G. and Olufunmilola, O. (2022). A survey of game-theoretic approach for resource management in cloud computing, *Journal of Computer Networks and Communications* **2022**: 1–13.
URL: *http://dx.doi.org/10.1155/2022/9323818*

Arif, M., Kiani, A. K. and Qadir, J. (2016). Machine learning based optimized live virtual machine migration over wan links.
URL: *https://doi.org/10.1007/s11235-016-0173-3*

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F. and Buyya, R. (2010). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Practice and Experience* **41**(1): 23–50.
URL: *http://dx.doi.org/10.1002/spe.995*

Canali, C. and Lancellotti, R. (2014). Improving scalability of cloud monitoring through pca-based clustering of virtual machines, *Journal of Computer Science and Technology* **29**(1): 38–52.
URL: *http://dx.doi.org/10.1007/s11390-013-1410-9*

Choudhary, A., Govil, M. C., Singh, G., Awasthi, L. K., Pilli, E. S. and Kapil, D. (2017). A critical survey of live virtual machine migration techniques, *Journal of Cloud Computing* **6**(1).
URL: *http://dx.doi.org/10.1186/s13677-017-0092-1*

Ghosh, S. and De, D. (2019). Weighted majority cooperative game based dynamic small cell clustering and resource allocation for 5g green mobile network, *Wireless Personal Communications* **111**(3): 1391–1411.
URL: *http://dx.doi.org/10.1007/s11277-019-06922-w*

Greenberg, A., Hamilton, J., Maltz, D. A. and Patel, P. (2009). The cost of a cloud: Research problems in data center networks, *SIGCOMM Comput. Commun. Rev.* **39**(1): 68–73.
URL: *https://doi.org/10.1145/1496091.1496103*

Hummaida, A. R., Paton, N. W. and Sakellariou, R. (2022). Scalable virtual machine migration using reinforcement learning, *Journal of Grid Computing* **20**(2).
URL: *http://dx.doi.org/10.1007/s10723-022-09603-4*

Imran, M., Ibrahim, M., Din, M. S. U., Rehman, M. A. U. and Kim, B. S. (2022). Live virtual machine migration: A survey, research challenges, and future directions, *Computers and Electrical Engineering* **103**: 108297.
URL: *http://dx.doi.org/10.1016/j.compeleceng.2022.108297*

Kansal, N. J. and Chana, I. (2016). Energy-aware virtual machine migration for cloud computing - a firefly optimization approach, *Journal of Grid Computing* **14**(2): 327–345.
URL: *http://dx.doi.org/10.1007/s10723-016-9364-0*

Kaur, A., Kumar, S., Gupta, D., Hamid, Y., Hamdi, M., Ksibi, A., Elmannai, H. and Saini, S. (2023). Algorithmic approach to virtual machine migration in cloud computing with updated sesa algorithm, *Sensors* **23**(13).
URL: *https://www.mdpi.com/1424-8220/23/13/6117*

Khan, M. S. A. and Santhosh, R. (2022). Hybrid optimization algorithm for vm migration in cloud computing, *Computers and Electrical Engineering* **102**: 108152.
URL: *http://dx.doi.org/10.1016/j.compeleceng.2022.108152*

Li, P. and Cao, J. (2023). Efficient online migration mechanism for memory write-intensive virtual machines, *Journal of Cloud Computing* **12**(1).
URL: *http://dx.doi.org/10.1186/s13677-023-00433-7*

Morabito, R. and Beijar, N. (2016). Enabling data processing at the network edge through lightweight virtualization technologies, *2016 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, pp. 1–6.

Muhammad, A., Xu, Y., Yan, C. and Chen, N. (2017). A balanced virtual machine migration mechanism based on genetic algorithm.

Najm, M. and Tamarapalli, V. (2022). Towards cost-aware vm migration to maximize the profit in federated clouds, *Future Generation Computer Systems* **134**: 53–65.
URL: *https://www.sciencedirect.com/science/article/pii/S0167739X22000954*

Shahapure, N. H. and Jayarekha, P. (2018a). Distance and traffic based virtual machine migration for scalability in cloud computing, *Procedia Computer Science* **132**: 728–737. International Conference on Computational Intelligence and Data Science.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050918308159*

Shahapure, N. H. and Jayarekha, P. (2018b). Virtual machine migration based load balancing for resource management and scalability in cloud environment, *International Journal of Information Technology* **12**(4): 1331–1342.
URL: *http://dx.doi.org/10.1007/s41870-018-0216-y*

Singh, S. and Singh, D. (2023). A bio-inspired vm migration using re-initialization and decomposition based-whale optimization, *ICT Express* **9**(1): 92–99.
URL: *https://www.sciencedirect.com/science/article/pii/S2405959522000224*

Tran, C. H., Bui, T. K. and Pham, T. V. (2022). Virtual machine migration policy for multi-tier application in cloud computing based on q-learning algorithm, *Computing* **104**(6): 1285–1306.
URL: *http://dx.doi.org/10.1007/s00607-021-01047-0*

Xu, X. and Yu, H. (2014). A game theory approach to fair and efficient resource allocation in cloud computing, *Mathematical Problems in Engineering* **2014**: 1–14.

Zhang, F., Liu, G., Fu, X. and Yahyapour, R. (2018). A survey on virtual machine migration: Challenges, techniques, and open issues, *IEEE Communications Surveys Tutorials* **20**(2): 1206–1243.

Zhang, Q., Zhu, Q., Zhani, M. F., Boutaba, R. and Hellerstein, J. L. (2013). Dynamic service placement in geographically distributed clouds, *IEEE Journal on Selected Areas in Communications* **31**(12): 762–772.

Zhao, H., Feng, N., Li, J., Zhang, G., Wang, J., Wang, Q. and Wan, B. (2023). Vm performance-aware virtual machine migration method based on ant colony optimization in cloud environment, *Journal of Parallel and Distributed Computing* **176**: 17–27.
URL: *https://www.sciencedirect.com/science/article/pii/S0743731523000187*