

# Virtualized Load Balancer for Hybrid Cloud Using Genetic Algorithm

S. Manikandan<sup>1,\*</sup> and M. Chinnadurai<sup>2</sup>

<sup>1</sup>Department of Information Technology, E.G.S. Pillay Engineering College, Nagapattinam, 611 002, India

<sup>2</sup>Department of Computer Science and Engineering, E.G.S. Pillay Engineering College, Nagapattinam, 611 002, India

\*Corresponding Author: S. Manikandan. Email: profmaninvp@gmail.com

Received: 10 August 2021; Accepted: 13 September 2021

**Abstract:** Load Balancing is an important factor handling resource during running and execution time in real time applications. Virtual machines are used for dynamically access and share the resources. As per current scenario cloud computing is played major for storage, resource accessing, resource pooling and internet based service offering. Usage of cloud computing services is dynamically increased such as online shopping, education, ticketing, etc. Many users can use the cloud resources and load balancing is used for adjusting the virtual machine and balance the node. Our proposed virtualized genetic algorithms are to provide balanced virtual machine services in Hybrid cloud. The proposed algorithm and experiments are implemented by using Cloud simulator. In this paper the experiments are done with cloud computing models, Virtual Machine allocation, load balancing and simulations. Also compare the results using response time, throughput and turnaround time using cloud sim. The accuracy can be compared with existing load balancing techniques.

**Keywords:** Load balancing; hybrid cloud; virtual machine; genetic algorithm; cloud simulator

## 1 Introduction

Cloud computing is a delivery service model based on requirements such as infrastructure, platform and software as services. Cloud is a internet storage model and provide plenty of services. Cloud computing application can access services from remote location data, sitting one place and access the resources from anywhere and any place. User retrieves the data from the storage and shares the resources [1]. Cloud service provider is an important for all cloud accessing. Cloud service provider enables the service offering, service level agreement and load balancing. Load Balancing is major issues for handling resources. At the same time multiple users can access the cloud means need to balance the service [2].

In general cloud computing has no need of physical location, configurations, required space and administration capabilities. The major features are virtualization, on demand service, pay for use, geographic access model, access privileges and service level agreements. It is internet based technology, provides dynamic reliable and efficient services [3]. On demand services is important key features because whenever the service is required that time request the service and pay for usage. Nowadays



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Amazon Web Services, VMWare, Google, IBM are providing efficient cloud services. In Cloud have public, private and hybrid cloud models. Private means only specified users will have access privileges, Public any use can access and hybrid means private and public cloud model [4].

Cloud facilitate an online platform for storage, compilation, online gaming, social media access, online shopping, online ordering, musical services, etc. Cloud computing has Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). IaaS have compute, network services, PaaS have IaaS and storage services and SaaS has PaaS and application services [5]. The primary objective of cloud is Load balancing. Load balancing is used for improving utilization, resource sharing and better response time [6].

Load balancing has static and dynamic model. Static has fixed assignment and execution time. Dynamic means response time, throughput and overhead are need to balanced while execution [7]. In this paper virtualized genetic algorithm is proposed for handling dynamic resource allocation and responses. This paper comprises, Section 2 explain related works, Section 3 proposed virtualized genetic algorithm, Section 4 gives experimental discussion and Section 5 explains conclusion and future plan.

## 2 Related Works

Check point algorithm is proposed by Cheug et al. for handling resource dynamically based on access point values. This cloudlet based access specification model which enables cloud can migrate one resource to another resource [8]. It has minimizing response time, threshold and execution time. While in this case not possible allocate resource during run time. Wong et al. honey bee algorithm has proposed with round robin circular queue method resource provisioning and trap polling procedure. So it has minimized execution time and bandwidth increased [9,10].

Quing et al. gives balanced load balancing mechanism for handling virtual machines, tasks and priority. The major issues are low priority tasks can hold the resource at longer time. Genetic algorithm can be used for selecting operator, storage levels, service level agreements and virtual provisioning as dynamically [11,12]. So the large scale operations are handled easily and increased efficiency. Ant colony based virtual provisioning method proposed by Manikandan et al. dynamically update the operation frequently. So we measure fitness function and throughput.

Yuvang et al. synchronized hill climbing algorithm is implemented handling each node values and their fitness function of measuring threshold values. This round robin based circular execution so the low tasks wait for more time to complete [13]. So quality of service cannot be calculated. Virtual machine arrives more request means surely need to provide load balancing system. Because cloud service provides enable load balancing system for handling multiple requests. So this situation the utilization factor is increased and quality of service well maintained. A major issue is at the time it will handle only on virtual machine and load balancing [14].

Enhanced virtualized load balancer is proposed handle more requests and select priority based resources. In this case each nodes are selected and apply load balancer for handling request and response system. The virtual machines are clustered based on their response and execution time. This is homogenous model so resource utilization and measuring response time need to wait for completing all tasks. The genetic based model is available means can we specify process and stop-wait model can be implemented [15].

## 3 Load Balancing

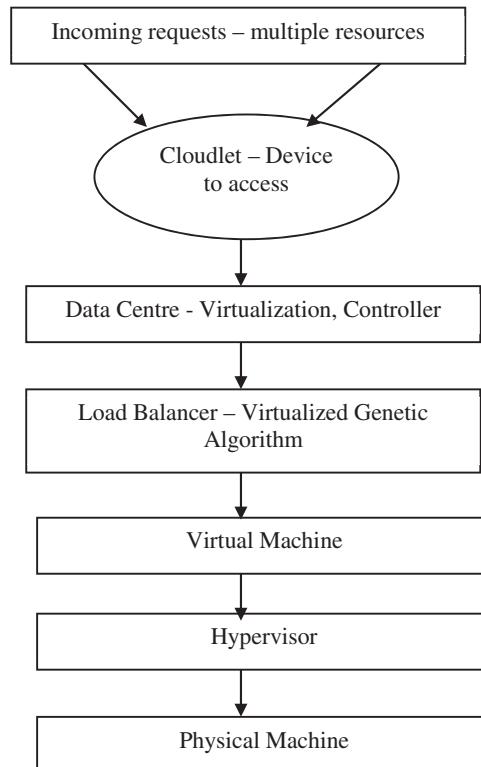
Load balancing is the mechanism to allocate resources to virtual machine in equal manner as dynamically. Cloud service providers are major key role playing for selecting load balancing operations. Our proposed virtualized genetic algorithm has following motivations such as response time, lesser

turnaround time, fully utilized resources, faulty tolerance, improved efficiency and stability. Load balancers is used for managing resources and balance the resource in equal manner.

*If load balance is in static mode—the cloud environment is in stable mode and so we need to fix all the cloud service requirements such as compute, network, storage and application.*

*In case load balance can be in dynamic—change the services as per request and requirements. Also resource provisioning system is enabled on demand and elasticity.*

*Our proposed algorithm has both static and dynamic provisioning method.*



**Figure 1:** Load balancing–process

In the above process, load balancer is handle the data centre inputs and virtual machine operations. The load balancer operates in virtual machine by using below code.

**Table 1:** Algorithm–load balancing in virtual machine

---

*Algorithm: Balancing Load to Virtual Machine*

*Begin*

*Connect -> resource()*

*Load*

*If(resource == available)*

*calculate connected\_values()*

*select virtual\_machine()*

*return resource*

*Else*

*Select next\_resource()*

*End*

---

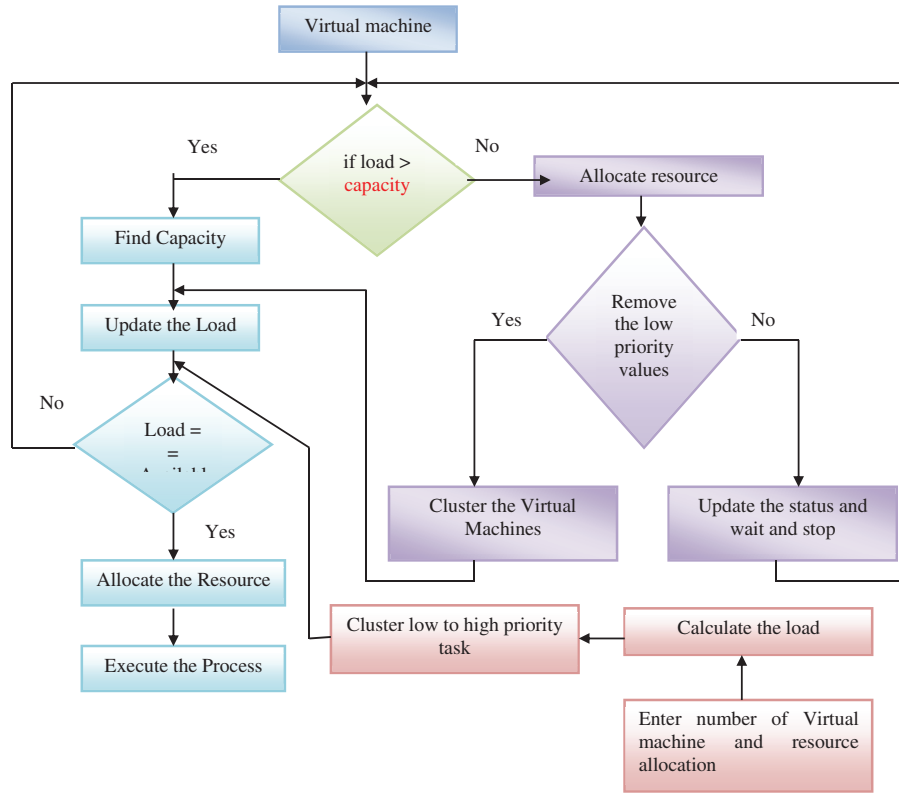


Figure 2: Proposed method-load balancer

#### 4 Experimental Setup-Genetic Algorithm

The evaluation process is done by using cloud simulator. This simulator is used for measuring burst time, turnaround time, load balancing results and capacity represented in Fig. 1. In this case CPU Processing time and task completion time are compared with existing load balancing algorithms shown in Tab. 1.

##### 4.1 Initial Process

Initial stage execution time and failure rate of each virtual machine are recorded. Select number of resource available for executing in virtual machine also taken. Guest, Host and Virtualization are three major factors are considered. Guest-Find the components and communicate to Virtualized Layer, Host-Manage the Guest, Virtualization-Compute, network and storage resources are handled and shown in Fig. 2.

The estimation function  $E_1, E, \dots E_n$  has an effect on target  $G$ . The selection matrix is represented as  $E = (e_{ij})n*n$ .

$$E = \begin{Bmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \dots & \dots & \dots & \dots \\ e_{n1} & e_{n2} & \dots & e_{nn} \end{Bmatrix} \tag{1}$$

So all column transactional matrix is calculated as follows

$$\text{Weight\_Consistency} = \sum_1^n \bar{e}_{ij}(i = 1, 2, \dots n) \tag{2}$$

**Table 2:** Algorithm-scheduling and virtualized genetic algorithm*Algorithm***Scheduling and Round Robin Process***Step 1: Create Virtual machine and Data centre for allocating task**Step 2: Check the current status of load in virtual machine and record the load value**Step 3: Cluster Virtual machine based on lowest and highest load**Step 4: Assign the incoming request to lowest cluster virtual machine and apply round robin scheduling**Step 5: Calculate burst time and set dynamic quantum time for rounding process which is  $\geq 1$* *Step 6: Allocate and start the process from least value and continue the process until vacant**Load Balancing**Step 7: Find the capacity and load value of each virtual machine**Step 8: If check virtual machine balanced or not and measure status overloaded/under loaded**Step 9: Remove heavily load virtual machine and repeat the round robin process until process will complete***Virtualized Genetic Algorithm***Step 10: Cluster the virtual machine based on loading, capacity and burst time values**Step 11: Sort the Virtual machine based on low to high priority constraints**Step 12: Calculate maximum execution time and allocate virtual machine***Threshold\_High** -> **Virtual\_Machine(d) | min** ( $\sum^{Threshold} \mathbf{High}$ )–**Minimum Task Completion time****Threshold\_Medium** -> **Virtual\_Machine(d) | min** ( $\sum^{Threshold} \mathbf{High} + \sum^{Threshold} \mathbf{Medium}$ )–**Minimum Task Completion time****Threshold\_Low** -> **Virtual\_Machine(d) | min** ( $\sum^{Threshold} \mathbf{Low}$ )–**Minimum Task Completion time***Step 13: Based on above result modify the process and repeat until process has to be suspend***Weighted Round Robin–Genetic Status***Step 14: Calculate all the virtual machine burst time, turnaround time and completion time**Step 15: Assign task to higher to lower burst time values**Step 16: Add the task to all the virtual machine and repeat all the above process from step 1 after each virtual machine completion*

$$\text{Combined weights} = W_i^* = \frac{W_i W_i'}{\sum_{i=1}^m W_i W_i'} \quad (3)$$

The transaction time measured as,

$$T(i) = \exp\left(\frac{\text{time}(W_i, CS_j) - \min(U_i)}{\max(W_i) - \min(W_i)}\right) \quad (4)$$

Based on above transaction time is recorded and allot to next section based on [Tab. 2](#).

**4.2 Cross Over-Virtualized Genetic Process**

Each cloud service attenuation factor is calculated as follows

$$P(i) = \frac{T_i^\tau}{\sum_{j=1}^n T_j^\tau} \quad (5)$$

where as  $\sum_{i=1}^n P(i) = 1$  and T–each burst values

Based on above representation the trust can be recorded as

$$Tr = \sum_1^m Sim_{T,w} \times T(i)_m \times P(i) \quad (6)$$

So quality factor measured by

$$Q = (Tr_j^*)^T * Sim(T, W) = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1k} \\ q_{21} & q_{22} & \cdots & q_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ q_{j1} & q_{j2} & \cdots & q_{jk} \end{bmatrix} \quad (7)$$

$$St(i) = Q * E(q)^T \quad (8)$$

So the above value is used for selecting simulating result of turnaround computation.

### 4.3 Best Value Computation–Task Completion

The finite number of virtual machine are selected and set least probability values. In case any delay fault occurred means genetic algorithm is used for selecting virtual machine from clustered groups,

$\phi$  is a factor to measure completed transaction.

$$\phi = Q * \left[ \frac{1}{(Tr + e^{-n})} \right] \quad (9)$$

This case eliminate minimum failure and transaction status is recorded. The objective weight function is calculated by using completed transaction and burst time of each process. The each transaction recorded and turnaround time is calculated as follows,

$$TT_j(W_j) = \frac{\text{sig}(Q_j)}{\sum_{j=1}^m \text{sig}(Q_j)} \quad (10)$$

Based on above result CPU processing time and task completion time is recorded.

From the above [Tab. 3](#) shows that the result of Burst time, Turnaround time and accuracy factors. In this case different virtual machine are classified and capacity level are fixed in 4 levels 5, 10, 15, 20 respectively. Virtualized genetic algorithm is applied and accuracy is calculated. Our proposed system average accuracy is 96% achieved at different virtual machine and different capacity values with load as 100 from [Tab. 4](#). Based on [Tab. 4](#), our proposed system accuracy is above 90% compare with other existing methods.

**Table 3:** Accuracy value, turnaround and burst time of virtual machines and various capacities

Virtual machine	Capacity	Load	Burst time	Turnaround time	Accuracy
5	5, 10, 15, 20	100	0.78, 0.89, 0.85, 0.91	0.98, 0.97, 0.91, 0.94	94
10	5, 10, 15, 20	100	0.89, 0.90, 0.85, 0.89	0.94, 0.97, 0.98, 0.96	95
20	5, 10, 15, 20	100	0.78, 0.88, 0.91, 0.88	0.98, 0.98, 0.98, 0.98	98
50	5, 10, 15, 20	100	0.82, 0.87, 0.88, 0.90	0.97, 0.95, 0.96, 0.96	96
100	5, 10, 15, 20	100	0.91, 0.92, 0.93, 0.91	0.96, 0.97, 0.95, 0.96	96
200	5, 10, 15, 20	100	0.91, 0.90, 0.89, 0.90	0.95, 0.97, 0.98, 0.97	96

**Table 4:** Proposed system and existing algorithms results

Virtual machine	Load	Accuracy				
		Proposed method	RNN classifier	SJF	Round robin	Eigen values
5	100	94	87	75	87	85
10	100	95	85	77	84	83
20	100	98	83	78	85	81
50	100	96	82	79	84	83
100	100	96	78	77	82	81
200	100	96	75	78	81	79

## 5 Conclusion

Cloud computing is one of the major IT developments for storing and retrieving resource via online. Most of the industries are using online based resource sharing mechanism for storage, infrastructure and developing applications. It is reduced the cost and pay for usage method. Load balancing is major important factors for implementing and efficient cloud services. At the same time more numbers of resources are access and used means load balancing is very important. In this paper virtualized genetic algorithm is proposed for implementing efficient load balancing mechanism. Our experiments are done by using cloud simulator and achieved good accuracy factors. Our proposed system compared with existing methods and our system gives good accuracy compare with others. In future the same method will be used for federated cloud environment applications.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] S. Sahu and P. Manish, "Efficient load balancing algorithm analysis in cloud computing," in *Proc. of the Fourth Int. Conf. on Communication and Electronics Systems (ICCES 2019)*, China, vol. 15, no. 6, pp. 177–184, 2019.
- [2] Karan and Bhalodia, "An efficient dynamic load balancing algorithm for virtual machine in cloud computing," in *Proc. of the Int. Conf. on Intelligent Computing and Control Systems (ICICCS 2019)*, India, vol. 11, no. 4, pp. 215–224, 2019.
- [3] K. Sheetal, "Enhanced improved genetic algorithm for load balancing in cloud computing," *Journal of Research in Computer Science and Engineering*, vol. 4, no. 1, pp. 275–286, 2020.
- [4] S. Manikandan, K. Raju, R. Lavanya and R. Hemavathi, "Energy efficiency controls on minimizing cost with response time and guarantee using EGC algorithm," *International Journal of Information Technology Insights & Transformations*, vol. 3, no. 1, pp. 131–137, 2017.
- [5] K. Manikanda Kumaran, M. Chinnadurai, S. Manikandan, S. Palani Muruganm and E. Elakiya, "An IoT based green home architecture for green score calculation towards smart sustainable cities," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 7, pp. 2377–2398, 2021.
- [6] K. Sheetal and G. Anshika, "Performance evaluation of check pointing and threshold algorithm for load balancing in cloud computing," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 5, pp. 2347–2693, 2018.

- [7] D. Nithin, S. Melvin, S. George and S. Jaya “Incorporating weighted round robin in honeybee algorithm for enhanced load balancing in cloud environment,” in *2017 IEEE Int. Conf. on Communication and Signal Processing*, Japan, vol. 15, no. 21, pp. 1971–1976, 2017.
- [8] S. Manikandan and M. Chinnadurai, “Effective energy adaptive and consumption in wireless sensor network using distributed source coding and sampling techniques,” *Wireless Personal Communication (2021)*, vol. 118, pp. 1393–1404, 2021.
- [9] N. Calheiros, R. Ranjan, A. Beloglazov César, D. Rose *et al.*, “Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2015.
- [10] S. Vinayak, D. Anas and L. Muhib. “Load balancing algorithms in cloud computing,” *International Journal of Computer Science Trends and Technology*, vol. 19, no. 7, pp. 957–966, 2016.
- [11] S. Manikandan, M. Chinnadurai, D. Maria Manuel Vianny and D. Sivabalaselvamani, “Real time traffic flow prediction and intelligent traffic control from remote location for large-scale heterogeneous networking using tensorflow,” *International Journal of Future Generation Communication and Networking*, ISSN: 2233-7857, vol. 13, no. 1, pp. 1006–1012, 2020.
- [12] S. Kripa and K. Kosal, “SIQ algorithm for efficient load balancing in cloud,” in *2017 IEEE Int. Conf. on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, Singapore, vol. 21, no. 13, pp. 1756–1766, 2017.
- [13] C. Priyanka and S. Sankari. “Comparative analysis on virtual machine assignment algorithm,” in *2017 IEEE Int. Conf. on Computing and Communication Technology*, Tokyo, vol. 17, no. 3, pp. 526–533, 2017.
- [14] D. Menno Dobber, M. Rob van dermei and K. Gerkoole “Dynamic load balancing and job replication in a global-scale grid environment: A comparison,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 2, pp. 207, 2016.
- [15] C. Zenon, M. Venkatesh, S. Shahrzad and M. Christopher, “Availability and load balancing in cloud computing,” in *2011 Int. Conf. on Computer and Software Modeling (IPCSIT)*, India, vol. 14, no. 3, pp. 219–224, 2011.