

## IRIAL-an improved approach for VM migrations in cloud computing

G. Dalin<sup>1\*</sup> and V. Radhamani<sup>2</sup>

Assistant Professor, Department of Computer Science, Hindustan College of Computer Science, Coimbatore, Tamilnadu, India<sup>1</sup>

Ph.D. Research Scholar, Department of Computer Science, Hindustan College of Arts and Science, Coimbatore, Tamilnadu, India<sup>2</sup>

©2018 ACCENTS

### Abstract

Cloud computing is an emerging technology in internet world. Load balancing is helped to distribute the dynamic workload between multiple nodes to ensure that no single node is overloaded. The proper utilization of resources is achieved by load balancing. Resource intensity aware load balancing (RIAL) method was proposed for load balancing in cloud computing. The resources like CPU, bandwidth, and memory and storage space in physical machines (PMs) are virtualized in the form of virtual machines (VMs) in Cloud computing. The resources in PMs are consumed by each VMs. The various resources in each PM were assigned different weight values based on the resource usage intensity of the PM by RIAL method. Based on the weight values, the VMs were selected from heavily loaded PMs to migrate out other lightly loaded PMs during load balancing operation. The destination PMs were selected to migrate selected VMs by using multi-criteria decision making (MCDM) method in which only lightly loaded PMs are considered. In PM, some resources are over utilized while other resources are underutilized. So, it is possible that the heavily loaded PM nearer to PM of selected migration VM might have required resources to balance the load. So in this paper, the VMs which are selected for migration are mapped with destination PMs globally. It is achieved by considering both the lightly loaded and heavily loaded PMs as destination PMs. For global mapping process, the expected completion time of each job in VMs of heavily loaded and lightly loaded PMs are calculated which decides the destination PMs through MCDM method. Hence, load balancing in cloud is further enhanced by improved RIAL (IRIAL) method. The simulation result shows that the proposed IRIAL method has better computational cost, performance degradation and execution time for load balancing in cloud.

### Keywords

Cloud computing, Load balancing, Resource intensity aware load balancing, Global map migration.

### 1.Introduction

Cloud Computing [1,2] is a distributed internet based paradigm which is designed for remote sharing and usage of different resources and services like applications, storage and computational capabilities etc. with high reliability over the large networks. Due to dynamic incoming requests, dynamism is required in its resource allocation. This inherent dynamism in cloud computing requires efficient load balancing mechanisms. Load balancing [3] concerns distribution of resources among the users in uniform manner so that no node is overload or sitting idle. Similarly, all other internet based distributed computing tasks, load balancing is an important aspect in cloud computing. The absence of load balancing leads to nodes degradation which violates service of level agreement (SLA).

The load balancing can be achieved by virtualization technology. In this technology, VMs are created by users in PMs. In cloud, each VMs in PMs used resources from their host PMs and run they run their own operating system. One of the methods for load balancing in cloud is RIAL [4]. According to resource intensities, RIAL method assigned different weights for different resources for each PM. Those weights were used to select the VMs to migrate and find the destination PM in each load balancing operation.

The RIAL based load balancing is improved in this paper through globally map, migration VMs and destination PMs in the cloud. For selecting the destination PMs, the heavily loaded PMs are also considered along with the lightly loaded VMs. Though the processes need to wait for a long time in a queue to execute in heavily loaded VMs, it has enough resources so that can execute the processes

\*Author for correspondence

with less response time. In the RIAL method only the lightly loaded PMs are considered for selecting the destination PMs whereas in the proposed IRIAL method, both lightly loaded and heavily loaded PMs are considered for selection of destination PMs. Hence, it improves the effectiveness and efficiency of the RIAL method.

## 2.Literature survey

A new load balancing algorithm called as heuristic-based load-balancing algorithm (HBLBA) [5] was proposed for Infrastructure as a Service (IaaS) cloud. A strategy was developed which configured the servers according to the number of incoming tasks and their sizes. It determined suitable VMs to maximize the utilization and assignments of computing resources. This algorithm was divided into two phases are server configuration and task VM mapping. The main intend of server configuration phase was to host how many numbers of VM instances and their types to serve the incoming tasks in order to reduce the makespan and utilized the resources. In the task VM mapping phase, a queuing model was adopted through tasks assigned to the VM in order to reduce the completion time and waiting time of tasks. But still the SLA violations are high.

An efficient dynamic load balancing algorithm [6] was proposed to balance load in cloud. This algorithm distributed load in cloud in a balanced approach. In addition to that, the tasks were allocated uniformly and systematically through checking particular parameters such as number of request the server which was handled by this algorithm. The performance time of the system was increased and response time for the server was decreased by balancing the load on the overloaded node to under loaded node. But there was not a clear explanation about the updating of counter variable in the proposed algorithm.

A novel approach through ant colony optimization (ACO) [7] was proposed for balancing workload dynamically in a cloud. In this proposed approach, two strategies are forward-backward ant mechanism and max rules were introduced which determine the candidate nodes for load balancing. The pheromone initialization and update were formulated according to physical resources under the cloud computing environment. It also considers pheromone evaporation, incentive and punishment rules of ACO. Combined with task execution, the moving probability of ants were defined in two ways that was whether the forward ant meets the backward ant or

not in neighbor node, with the aim of accelerating searching processes.

A soft computing approach based load balancing [8] was proposed to balance the workload in cloud. The incoming jobs to the VMs or servers were allocated by a local optimization approach where stochastic hill climbing was used. It was an iteration process, where the jobs were moved in the direction of increasing value, which was uphill. This algorithm terminates when it reaches a peak where no neighbors has a higher value. However, this approach still needs an improvement it can be achieved by using other soft computing approaches like Round Robin and First Come and First Serve.

A dynamic scheduling algorithm [9] was proposed for balancing workloads among VMs. The architecture of cloud resource broker was modified and the proposed dynamic scheduling algorithm was based on the last optimal k-interval that not only reduces the makespan time of tasks but also enhanced the ratio of tasks to meet the deadline and fulfill the objective of elasticity in cloud environment. The main intend of the dynamic resource allocation scheme was to monitor and predict the resource based on the needs of the cloud applications.

## 3.Proposed methodology

In this section, the proposed IRIAL method for load balancing in cloud computing is described in detail. Similar to RIAL, the proposed IRIAL method selects the VMs to migrate out. Then, while selecting the destination PMs in RIAL method resource intensity, VM communication rate and performance degradation of lightly loaded PMs are considered. In the proposed IRIAL the expected completion time of each job in VMs of both lightly loaded and heavily loaded PMs are also considered during the selection of destination PMs. Because, huge processes are required a lot of resources which are available in heavily loaded PMs. If the calculated measures of heavily loaded PMs are lesser than lightly loaded then heavily loaded PMs are used for selection of destination otherwise lightly loaded PMs are used in selection of destination PMs.

### 3.1Design of IRIAL

In IRIAL, each PM  $P_x$  periodically checks its utilization for each of its type  $t$  ( $t \in T$ ) resources to see if it is overloaded.  $P_x$  denotes PM  $x$  ( $x = 1, 2, \dots, N$ ) where  $N$  denotes the total PM serve as a resource pool in the cloud. The non-over utilized and over utilized types of resources are represented by

NO and O respectively. Then some measures are calculated for both heavily loaded and lightly loaded PMs. Based on the measures, either heavily loaded PMs or lightly loaded PMs are used in selecting of destination PMs. Then this PM triggers VM migration to migrates its VMs to other PMs until utilization rate of type  $s$  resources in  $P_x$  ( $u_{xs}$ ) is less than or equal to predetermined utilization threshold for the  $s^{\text{th}}$  type in a PM in the cloud.

### 3.1.1 Selection of virtual machines to migrate

A method is used which determine the weight of each type of resource according to resource intensity. The main intend of this method was to reduce the workload of PM, by finding VMs to migrate out of each overloaded  $P_x$ . Then, check the  $P_x$  if it is over utilized in CPU. If it is over utilized, then the VM with the highest utilization of CPU is selected in order to relieve  $P_x$ 's load. There is no need to reduce the utilization of resources in  $P_x$  when non-over utilized resources do not overload  $P_x$ . Therefore, the VM with the lowest utilization in non-over utilized resources is selected to fully utilize the resources. By considering these two factors, the weights for each type of resources according to its overloaded status in  $P_x$  are determined. Relatively higher weights are assigned to over utilized resources than non over utilized resources. The weights for a non-over-utilized resources with resource utilization is calculated as,

$$w_{xs} = 1 - u_{xs} \quad (1)$$

For the over utilized resources which have similar but different utilization values, lower weights are assigned to resources with lower utilization and higher weights are assigned to the resources with higher utilization. Hence, based on the difference between the resource utilization the weights of resources are differentiated. So a power function is utilized with the basic form to find out the weight for an over utilized resource with resource utilization  $u_{xs}$ . The weights given to a resource is calculated by following equation.

$$w_{xs} = \begin{cases} \frac{1}{1-u_{xs}}, & \text{if } s \in O \\ 1 - u_{xs}, & \text{if } s \in NO \end{cases} \quad (2)$$

An efficient method called as Multi Criteria Decision Making (MCDM) method is used to select the VM to migrate. MCDM [10] refers to making decisions in the presence of multiple, usually conflicting, criteria. Here, MCDM method is used to calculate the weighted distances of all candidates from the ideal solution and chooses one candidate which holds the shortest weighted distance. This method creates a

$|S| \times n_x$  decision matrix  $M_{xu}$  for PM  $P_x$  with  $n_x$  VMs is given as follows:

$$M_{xu} = \begin{pmatrix} u_{x11} & \cdots & u_{xn_x1} \\ \vdots & \ddots & \vdots \\ u_{x1|S|} & \cdots & u_{xn_x|S|} \end{pmatrix} \quad (3)$$

Where,  $u_{xys}$  is the type- $s$  resource utilization rate of VM  $V_{xy}$ . In above equation 3, each column represents each VM in  $P_x$  and each row represents one type of resourc. If the heterogeneous VMs are used, a normalized VM utilization is given as follows:

$$T_{xu} = \begin{pmatrix} t_{x11} & \cdots & t_{xn_x1} \\ \vdots & \ddots & \vdots \\ t_{x1|S|} & \cdots & t_{xn_x|S|} \end{pmatrix} \quad (4)$$

$$\text{Where, } t_{xys} = \frac{u_{xys}}{\sqrt{\sum_{y=1}^{n_x} u_{xys}^2}}$$

Then the ideal migration VM ( $R_{VM}$ ) is finding out which has the lowest usage of non-overloaded resources and highest usage of over utilization resources.

$$R_{VM} = \{r_{x1}, \dots, r_{x|S|}\} = \left\{ \begin{pmatrix} \max_y t_{xys} | s \in O \\ \min_y t_{xys} | s \in NO \end{pmatrix} \right\} \quad (5)$$

For each type  $s$  resource, if it is over utilized, its  $r_{xs}$  is the largest element from  $(t_{x1s} \dots t_{xys} \dots t_{xn_x1})$  in  $T_x$  otherwise  $r_s$  is the smallest element.

MCDM used completion time of each job in VM to establish a  $|J| \times n_x$  decision matrix  $M_x$  for PM  $P_x$  with  $n_x$  VMs is given in (6). By considering completion time along with the resource utilization, the most effective VM is selected to migrate.

$$M_{xCT} = \begin{pmatrix} CT_{x11} & \cdots & CT_{xn_x1} \\ \vdots & \ddots & \vdots \\ CT_{x1|J|} & \cdots & CT_{xn_x|J|} \end{pmatrix} \quad (6)$$

Where,  $CT_{xyj}$  is the  $j$ -jobs expected completion time of VM  $V_{xy}$ .

In above equation 6, each row represents one type of jobs and each column represents each VM in  $P_x$ . The expected completion time of each job in VMs of PM is calculated by

$$CT_{pq} = \text{time}_{pq} + ET_{pq} \quad (7)$$

where,  $0 \leq p \leq j$  and  $0 \leq q \leq y$ ,  $j$  denotes the number of jobs,  $y$  denotes the number of VMs,  $\text{time}_{pq}$  denotes the time for the execution of all the

jobs that have previously been assigned to do it and ET denotes the execution time of current job. If the heterogeneous VMs are used, a normalized expected completion time of each job in VMs of PM is given as follows:

$$H_{x_{CT}} = \begin{pmatrix} h_{x11} & \cdots & h_{xn_x1} \\ \vdots & \ddots & \vdots \\ h_{x1|J|} & \cdots & h_{xn_x|J|} \end{pmatrix} \quad (8)$$

Where,  $h_{xyj} = \frac{CT_{xyj}}{\sqrt{\sum_{y=1}^{n_x} CT_{xyj}^2}}$ .

Then the ideal migration VM ( $R_{VM}$ ) is finding out resources which has the high expected completion time and resources has the low expected completion time.

$$Q_{VM} = \{q_{x1}, \dots, q_{x|J|}\} = \left\{ \begin{pmatrix} \max_y h_{xyj} | \text{Time} \in \text{High} \end{pmatrix}, \begin{pmatrix} \min_y h_{xyj} | \text{Time} \in \text{Low} \end{pmatrix} \right\} \quad (9)$$

Where, high denotes the High expected completion time and low denotes the Low expected completion time.

For each j-jobs, if it has high expected completion time, its  $q_{xj}$  has the maximum completion time element from  $(h_{x1j} \dots h_{xyj} \dots h_{xn_xj})$  in  $H_{x_{CT}}$  otherwise  $q_j$  has minimum completion time.

The Euclidean distance of each candidate  $V_{xy}$  in  $P_x$  with the expected completion time  $h_{xyj}$ , ideal  $T_{xy}$  and ideal VM is calculated as

$$l_{p,xy} = \sqrt{\sum_{s=1}^{|S|} [w_{xs}(t_{xys} - r_{xs})]^2 + [w_{xj}(h_{xyj} - q_{xj})]^2 + [w_t T_{xyp}]^2} \quad (10)$$

Where,  $w_t$  is the weight of the communication rate.

The communication rate between  $V_{xs}$  and PM  $P_p$  as

$$T_{xyp} = \sum_{q=1}^{n_p} T_{xypq} \quad (11)$$

$n_p$  denotes the number of VMs in PM.

The migration VM is the VM with the shortest Euclidean distance. Then, IRIAL checks if the selected VM  $V_{xy}$ 's  $u_{xys}$  ( $s \in S$ ) is in the ideal migration VM. If so, IRIAL replaces  $V_{xy}$ 's  $u_{xys}$  in the ideal migration VM with the updated value. This process is continued to select the VM with the second shortest Euclidean distance. Through this process, IRIAL keeps selecting migration VMs from  $P_x$  until  $P_x$  is no longer overloaded.

### 3.1.2 Selection of destination PMs by globally mapping migration VM with PMS

While selecting destination PMs to migrate the selected VMs from  $P_x$ , there are different measures such as VM communication rate  $T_{xy}$ , performance degradation  $D_{xy}$  and resource intensity are considered. Along with these measures the expected completion time of both heavily loaded PMs and lightly loaded PMs are considered. These measures are calculated for heavily loaded PMs and lightly loaded PMs. If these measures of heavily loaded PMs are lesser than lightly loaded PMs, then heavily loaded PMs are used in decision matrix. Otherwise lightly loaded PMs are used in decision matrix. The MCDM method is also used for destination PM selection. A  $|S| \times |D|$  decision matrix is built:

$$M'_{xu} = \begin{pmatrix} u_{11} & \cdots & u_{|J|1} \\ \vdots & \ddots & \vdots \\ u_{1|S|} & \cdots & u_{|J||S|} \end{pmatrix} \quad (12)$$

where, D denotes the set of either heavily loaded PMs or lightly loaded PMs, each row denotes the type of resources and each column denotes each heavily or lightly loaded PMs. The above matrix is normalized by

$$T'_{xu} = \begin{pmatrix} t'_{11} & \cdots & t'_{|J|1} \\ \vdots & \ddots & \vdots \\ t'_{1|S|} & \cdots & t'_{|J||S|} \end{pmatrix} \quad (13)$$

where,  $t'_{ys} = \frac{u_{ys}}{\sqrt{\sum_{y=1}^{|J|} u_{ys}^2}}$ .

The ideal destination PM is represented as,

$$R'_{PM} = \{r'_1, \dots, r'_s, \dots, r'_{|S|}\} = \left\{ \begin{pmatrix} \min_y t'_{ys} | s \in S \end{pmatrix} \right\} \quad (14)$$

A  $|J| \times |D|$  decision matrix is built based on expected completion time

$$M'_{x_{CT}} = \begin{pmatrix} CT_{11} & \cdots & CT_{|D|1} \\ \vdots & \ddots & \vdots \\ CT_{1|J|} & \cdots & CT_{|D||J|} \end{pmatrix} \quad (15)$$

Where, D denotes the set of either heavily loaded PMs or lightly loaded PMs, each row denotes the type of resources and each column denotes each heavily or lightly loaded PMs. The above matrix is normalized by

$$H'_{x_{CT}} = \begin{pmatrix} h'_{11} & \cdots & h'_{|D|1} \\ \vdots & \ddots & \vdots \\ h'_{1|J|} & \cdots & h'_{|D||J|} \end{pmatrix} \quad (16)$$

Where,  $h'_{yj} = \frac{CT_{yj}}{\sqrt{\sum_{y=1}^{n_x} CT_{yj}^2}}$ .

The ideal destination PM is represented as,

$$Q'_{PM} = \{q'_1, \dots, q'_j, \dots, q'_{|J|}\} = \left\{ \left( \min_y h'_{yj} \mid \text{time} \in \text{Low} \right) \right\} \quad (17)$$

The weight of j-jobs represents the priority of migrating this resource out from overloaded PM  $P_x$ . Thus, it indicates the priority of considering available resource in selecting destination PMs. Therefore, these weights are used for different resources in candidate PMs to find the most suitable PMs.

By considering different measures, the Euclidean distance of each candidate PM  $P_p$  from the ideal PM is calculated as,

$$l_{p,xy} = \sqrt{\sum_{s=1}^{|S|} [w_{xs}(t'_{ps} - r'_s)]^2 + [w_t(T_{xyp} - T_{max})]^2 + [w_d(D_{xy})]^2 + [w_{xj}(CT'_{pj} - q'_j)]^2} \quad (18)$$

where,  $w_d$  denotes the weight of performance degradation that can be adaptively adjusted like  $w_t$ .

The PM which has lowest  $l_{p,xy}$  is selected as the destination PMs. Thus, the IRIAL technique is balancing workload effectively.

## 4.Result and discussion

In this section, the performance of IRIAL and RIAL are tested in CloudSim simulator [11]. The RIAL and IRIAL are evaluated in terms of communication cost reduction, number of migrations and performance degradation.

### 4.1Communication cost reduction

The communication cost reduction is the difference between the total communication cost observed at a certain time point from the initial total cost of all VMs. The following Table 1 shows the comparison of communication cost reduction between RIAL and IRIAL for different timings.

**Table 1** Comparison of communication cost reduction

Time (hrs)	RIAL	IRIAL
8	80	120
16	130	160
24	175	200

Figure 1 shows the comparison of communication cost reduction between existing and proposed load balancing methods. The time is taken in x axis in terms of hours and communication cost reduction is taken in y axis. At 8 hours, the communication cost reduction of existing RIAL method is 80 and proposed IRIAL method is 120. From the Figure 1, it

is known that the proposed IRIAL method outperforms than the existing RIAL method in terms of communication cost reduction.

### 4.2Number of migrations

The number of migrations denotes the migration of VMs from one PM to another PM. The following Table 2 shows the comparison of number of migrations between RIAL and IRIAL for different number of VMs.

**Table 2** Comparison of number of migrations ( $\times 10^3$ )

Number of VMs	RIAL	IRIAL
2500	1.2	0.72
3000	2.6	2.0
5000	3	2.5

Figure 2, Shows the comparison of number of migration between existing and proposed load balancing methods. The number of VMs are taken in x axis and number of migrations are taken in y axis. When the number of VMs is 2500, the number of migrations of existing RIAL method is  $1.2 \times 10^3$  and proposed IRIAL is  $0.72 \times 10^3$ . From the Figure 2, it is known that the proposed IRIAL method outperforms than the existing RIAL method in terms of number of migrations.

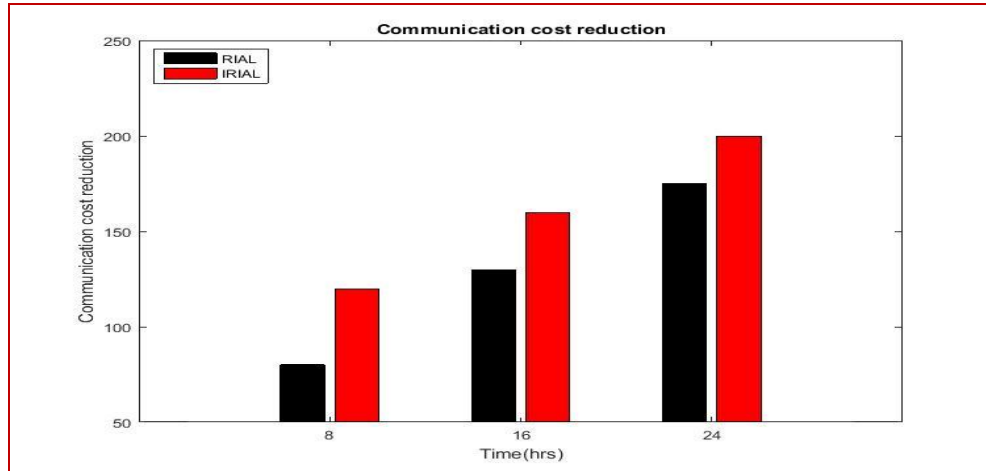
### 4.3Performance degradation

When a VM is being migrated to another PM, its performance is degraded. Based on the migration of VMs performance degradation is calculated. The following Table 3 shows the comparison of performance degradation between RIAL and IRIAL for different number of VMs.

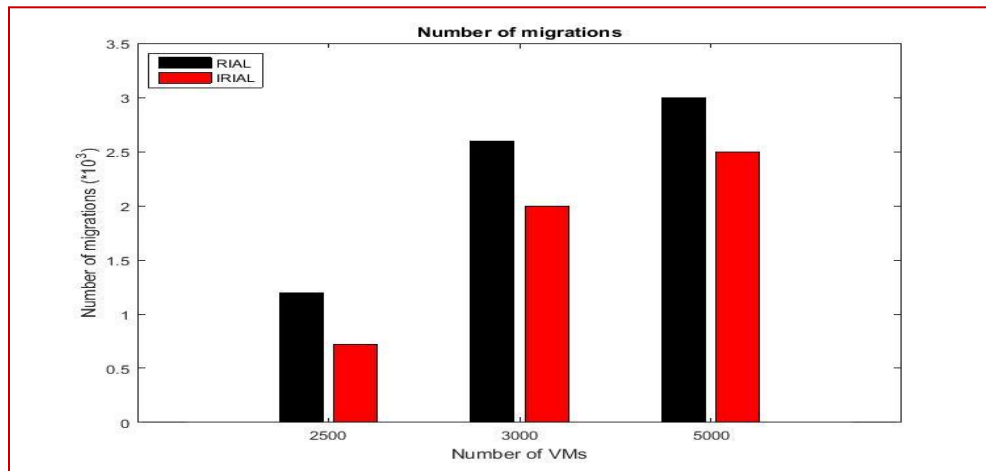
**Table 3** Comparison of performance degradation ( $\times 10^9$ )

Number of VMs	RIAL	IRIAL
2500	0.18	0.09
3000	0.87	0.72
5000	5	3

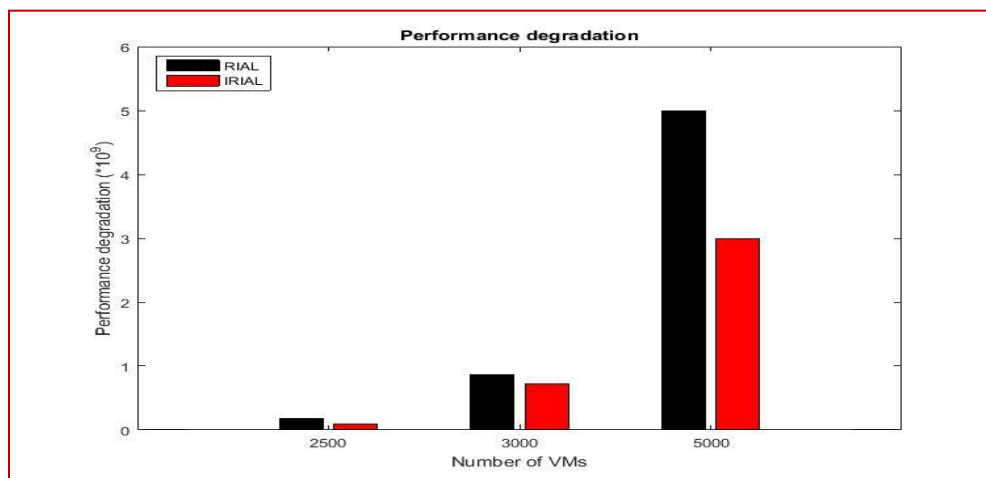
Figure 3, shows the comparison of performance degradation between existing and proposed load balancing methods. The number of VMs are taken in x axis and performance degradation are taken in y axis. When the number of VMs is 2500, the performance degradation of existing RIAL method is  $0.18 \times 10^9$  and proposed IRIAL  $0.09 \times 10^9$ . From the Figure 3, it is known that the proposed IRIAL method outperforms than the existing RIAL method in terms of performance degradation.



**Figure 1** Comparison of communication cost reduction



**Figure 2** Comparison of number of migration



**Figure 3** Comparison of performance degradation



## 5. Conclusion

This paper is focused on improving the load balancing in cloud computing. Initially the proposed IRIAL, selected the VMs which are needed to migrate from one PM to another PM. The selection is based on MCDM method. Then the destination PMs are selected based on different measures such as expected completion time, resource intensity, VM communication rate and performance degradation of both heavily or lightly loaded PMs. By considering these measures of heavily or lightly loaded PMs, the efficiency of IRIAL method is improved. The experiments are conducted in CloudSim simulator to prove the effectiveness of the proposed IRIAL. The experimental results shows the proposed IRIAL has better communication cost reduction, number of migration and performance degradation than RIAL. This research work can be extended by considering thermal power, energy consumption and bandwidth along with the completion time and resource utilization to migrate the VMs to the destination PMs which may improve the performance of IRIAL method.

## Acknowledgment

None.

## Conflicts of interest

The authors have no conflicts of interest to declare.

## References

- [1] Singh A, Juneja D, Malhotra M. Autonomous agent based load balancing algorithm in cloud computing. *Procedia Computer Science*. 2015; 45:832-41.
- [2] Dave A, Patel B, Bhatt G, Vora Y. Load balancing in cloud computing using particle swarm optimization on Xen Server. In *Nirma university international conference on engineering 2017* (pp. 1-6). IEEE.
- [3] Kaur R, Luthra P. Load balancing in cloud computing. In *proceedings of international conference on recent trends in information, telecommunication and computing 2012* (pp.374-81).
- [4] Chen L, Shen H, Sapra K. RIAL: Resource intensity aware load balancing in clouds. In *conference of computer communications 2014* (pp. 1294-302). IEEE.
- [5] Adhikari M, Amgoth T. Heuristic-based load-balancing algorithm for IaaS cloud. *Future Generation Computer Systems*. 2018; 81:156-65.
- [6] Haryani N, Jagli D. Dynamic method for load balancing in cloud computing. *IOSR Journal of Computer Engineering*. 2014; 16(4):23-8.
- [7] Gao R, Wu J. Dynamic load balancing strategy for cloud computing with ant colony optimization. *Future Internet*. 2015; 7(4):465-83.
- [8] Mondal B, Dasgupta K, Dutta P. Load balancing in cloud computing using stochastic hill climbing-a soft computing approach. *Procedia Technology*. 2012; 4:783-9.
- [9] Kumar M, Sharma SC. Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment. *Computers & Electrical Engineering*. 2018;69:395-411.
- [10] Hwang CL, Yoon K. Methods for multiple attribute decision making. In *Multiple Attribute Decision Making*, Springer, Berlin, Heidelberg; 1981, p.58-191.
- [11] Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, 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.



**Dr. G. Dalin** received Ph.D in Computer Science in 2013. He has around 12 years of teaching experience. He is currently an Associate Professor at PG & Research Department of Computer Science, Hindustan College of Arts and Science, Coimbatore. He has published more than 25 research articles in International Journals. He has acted as resource person in both national and international conferences and chaired technical sessions too. He has professional body memberships in popular International Journals. He is a reviewer and editorial member in International and National Journals. His research interests include Computer Networks, Routing Algorithms, and Cloud Computing. Email: profgdalin@gmail.com



**Radhamani V** received Diploma in Computer Science and Engineering degree from State Board of Technical Education, Madras in 1993. She has completed Under Graduate and Post Graduate degrees in Computer Science under Bharathiar University. She has received M.Phil degree in Computer Science from the same university in 2013. Currently, she is pursuing Ph.D. degree in Computer Science at Hindustan College of Arts, Coimbatore under Bharathiar University. She is currently the Assistant Professor in the Department of Computing, Coimbatore Institute of Technology, Coimbatore. Her research interest includes Data Analytics, Cloud Computing, System Software as well as Computer Architecture.