International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

An Intelligent Genetic Base Algorithm for Optimal Virtual Machine Migration in Cloud Computing

Neha Sharma, Jaspreet Singh, Lokesh Pawar

Abstract: The cloud computing is the architecture in which virtual machines are involved and connect with the cloud service provider. The virtual machines connect with the cloud service provider on behalf of the users Nowadays load balancing is very big issues to solve this issue various load balancing approaches are proposed. The virtual machines get overburden due to uncertainties and to handle this challenge the genetic algorithms is applied for virtual machine migration in cloud environment. In this research work, the proposed algorithm is applied for virtual machine migration and outcomes are correlated with the existing used genetic algorithm in terms of latency, bandwidth utilization, and space utilization. The results show that our proposed approach achieves better performance for cloud user and maximal profits for cloud providers by doing optimal virtual machine migration.

Keywords: - Genetic Algorithm, Intelligent genetic algorithm (IGA), Optimal Migration Using Genetic Algorithm

I. INTRODUCTION

Cloud computing is an archetype of computing. In this paradigm, large numbers of servers are linked to public and private networks. These linked systems support a dynamically extensible framework for file storage, information and application. The term Cloud computing describes both a platform and a form of application. The development of cloud computing decreased the computation amount, content depository, application hosting, and transfer in a significant manner [1]. It is a realistic technique to understand direct rate advantages. This technology can alter an information center from a capital-intensive arrange to a changeable value setting. Cloud computing approach is established on an extremely basic principle of the utilization of IT potentials. The services models comprises by cloud computing are Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Storage as a service (STaas), Security as a service (SECaaS), etc. The users of Software as a Service give payment to utilize applications running within the Cloud's provider framework such as SalesForce. The applications are normally provided to the users through the Internet. The Cloud provider manages these applications wholly. The Cloud Providers of Platform as a Service is provide a platform to developer for example Google App Engine. This allows users to install custom software by means of the tools and programming languages recommended by the cloud provider. Infrastructure as a Service delivers hardware resources for example CPU, disk space or network devices in the form of service. The Cloud provider distributes these resources as a virtualization platform. The client can get access to these resources using the Internet. The client manages virtualized platform [2]. In this service model, a big service supplier rents space in their storage structure on the payment base. This service model is frequently utilized to resolve the offsite backup concerns. In this model, a huge service provider combines its protected services into a business framework on payment basis in a cost-effective way in comparison with several individuals or companies. In the scenario of cloud computing, load balancing offers an effective answer to different problems occurring within the cloud computing environment system and practice. Load balancing must consider two major tasks. These tasks are resource distribution and scheduling in a dispersed environment. Load balancing reassigns the whole assignment to the individual nodes for a group of arrangement for utilizing time and resources in a more effective way. Load balancing is the fundamental method used for the scaling of an application server system [3].According to the increased demand of the users the recent system are combined to the resource pool .In this situation, the load balancer, transfer the traffic directly to the novel server. Considerable enhancement in performance, constancy preservation of the system, improvement in the flexibility of the system and development of a fault tolerant system through the creation of backups are some goals of load balancer. There are two kinds of load balancing approach first one is static load balancing technique and second one is dynamic load balancing approach. In the static load balancing approach, the decision about the transferring of the weight does not rely on the existing system state. It needs awareness about the applications and resources of the system. At the time of job arrival, the performance of the virtual machines is determined. The master processor allocates workload to other slave processors as per their functioning. Therefore, the slave processors perform the assigned work and the outcome is delivered to the master processor [4]. The dynamic load balancing algorithms use the present state data of the system for making any choice intended for load balancing. Therefore, the load shifting depends upon the current state of the system.

Revised Manuscript Received on 30 May 2019.
* Correspondence Author
Neha Sharma*, M.tech (CSE) Scholar Chandigarh University Mohali, Punjab, India.
Jaspreet Singh, Department of CSE, Chandigarh University Mohali, Punjab, India.
Lokesh Pawar, Department of CSE, Chandigarh University Mohali, Punjab, India.

It permits procedures to shift from an overused mechanism to an underused mechanism animatedly for sooner implementation. Round robin is the static load balancing approach. This algorithm makes use of a time-slicing method. This algorithm works in a round manner. In a round manner, a time slice is allotted to every node and each node has to wait for its turn. The time is divided and the interval is given to every node. Every node has to perform its task in the allotted time slice. Another static load balancing technique is Opportunistic Load Balancing approach. This algorithm does not take into account the current workload of the virtual machine. This algorithm generally keeps all nodes busy. This technique supplies a load balancing agenda however does not generate a better outcome. Min-Min is a static load balancing technique. In this technique, the parameters related to the task are acknowledged earlier. The cloud manager recognizes the implementation and completion time of the unassigned task waiting nearby [5]. The jobs with minimum execution time are being assigned firstly to the processors for the completion of task in time. But the tasks with the highest execution have to delay for a particular time period. The max-min algorithm is identical to the Min-Min approach. In this algorithm, the maximum time jobs are chosen after the completion of minimum time tasks. After the completion of minimum jobs, the queued tasks are given to the processor. As it is a static algorithm, therefore the time of each task is computed in advance and performed in an accurate way. The Ant algorithm is a multi-agent approach to complex combinative development issues. Example of this approach is travelling salesman problem (TSP). Real ant colonies were the inspiration behind these algorithms. Ant's behavior is focused more to the continued existence of the colonies. They do not determine for the separate ant. Honeybee Foraging load balancing approach is one of the dynamic load balancing algorithms [6]. The design of these algorithms is based on the behavior of honey bees. The behavior of honey bee in load balancing technique has stimulated to decrease the response time of the virtual machine which decreases the waiting time as well. This algorithm does not show any development in throughput which is the main drawback of this algorithm. Active Clustering is a modified technique of random sampling. This algorithm uses the idea of clustering. Mutual grouping of similar nodes and execution of task on these grouped nodes is the main principle of this algorithm.

II. LITERATURE REVIEW

Jargalsaikhan Narantuya, et.al (2018) proposed a novel service-conscious approach for the cloud to cloud migration of services on numerous virtual machines [7]. The proposed approach analyzed the dependence of numerous virtual machines by means of network traffic strength in the determination of the migration sequence of dependent virtual machines for reducing the service downtime. The proposed migration technique was applied in an Open Stack-based test bed scenario. The tested outcomes demonstrated that the dependence amid the virtual machines was effectively recognized for the cloud to cloud migration. The proposed technique that exploited dependence amid the virtual machines considerably reduced the service downtime. It was identified that the proposed approach decreased the standard service downtime beyond 50% throughout the cloud to cloud migration. The future study will inspect the modelling and growth of cloud to cloud migration in additional complex cloud scenarios. Fei Zhang, et.al (2018) presented a summary of virtual machine migration and reviewed both its advantages and issues [8]. Three viewpoints named as manner, distance, and granularities were used to classify virtual machine migration methods. The studies performed on non-live migration were simply analyzed. Several surveys were carried out on live migration in an inclusive manner on the basis of three main issues it experienced i.e. transfer memory information and storage information transfer. To evaluate the performance of virtual machine migration, quantitative scrutiny was executed. User mobility became a significant inspiration for live virtual machine migration in some situations with the growth and advancement of cloud computing. Therefore, the works related to the linking of VM migration to user mobility were also reviewed. Uttam Mandal, et.al (2018) developed schemes for obtaining appropriate bandwidth and pre-copy redundancy calculation to enhance distinct accomplishment metrics of virtual transfer over a WAN [9]. In the initial phase, some models were formulated for the measurement of network resource utilization, migration period, and migration downtime. After that, a strategy was proposed for the determination of suitable transformation of bandwidth and amount of pre-copy iterations. Several mathematical tests were carried out in numerous cloud environments with a huge volume of migration requests. The tested outcomes depicted that the proposed approach consumed less number of network resources in comparison with highest and lowest bandwidth supplies approaches. The proposed approach achieved a considerably lesser migration period in comparison with minimum-bandwidth strategy. Umesh Deshpande, et.al (2018) introduced an advanced metric for live transformation of virtual machines termed as removal time [10]. This time was used in the eviction of one or more states of virtual machines from the source host. In this study, Scatter-Gather lives migration was also proposed. This approach decoupled the beginning and ending during transformation for reducing expulsion time in case of slow destination. The memory of virtual machines to several nodes was scattered by the source which included the destination and one or more mediators. At the same time, the destination gathered the memory of virtual machines from the mediators and the source. Therefore, the response rate of the target is no longer controlled the eviction from the source. In this study, parallel live expulsion of multiple virtual machines was supported, and reduplications were exploited for the reduction of network overhead. Weizhan Zhang, et.al (2017) proposed a cluster-aware virtual machine combined migration approach for media cloud, firmly combine group, placing, and dynamic transfer procedure [11]. The proposed approach employed a clustering technique and a placement approach for the attainment of ultimate migration schemes for recently professed media server clusters. It is also used for the efficient execution of the migration procedure of media servers. The tested outcomes showed that the proposed approach is able to transfer virtual media servers in the media cloud efficiently by decreasing the entire domestic services in DCN in the resource expenditure limitations of the media streaming operation.

In the future, the performance of other packet level simulators like SimGrid or GreenCloud will be analyzed. Fei Tao, et.al (2016) scrutinized the issue of dynamic transformation of virtual machines (DM-VM) in the cloud computing paradigm [12]. In this study, a triple-objective optimization model for DM-VM was proposed. The proposed model considered power expenditure, the interaction between virtual machines, and migration cost under the usual working of paradigm. A binary graph matching-based bucket-code learning algorithm (BGM-BLA) was intended to solve the DM-VM issue. In this algorithm, BLA is used to find the topmost solutions. Binary graph matching was utilized to evaluate the candidate solutions. The tested outcomes depicted that the determined technique execute comparatively well in the condition of the attained Pareto sets and calculation time than two optimization algorithms. Jaspreet Singh, et.al (2017) proposed an algorithm named as multi-queue scheduling technique in cloud computing. In this algorithm, the user job is divided into multi queues and accomplishes the dynamic collection of user jobs for execution. MCQ model is efficiently separate the customer job into multi queues then give more importance in formation of combine jobs pattern by merging customer task from both queues for execution, so the approach allow us to enhance energy consumption, reduce execution time and decrease overall cost. This technique accomplishes a high degree of job scheduling in cloud computing [13]. Author of [15] proposed an effective job scheduling algorithm that decreases the execution time and energy consumption to some quantity for cloud customer's jobs.

III. SYSTEM ARCHITECTURE

In this research, an intelligent genetic algorithm (IGA) is used to determine the issue of node failure and node migration within the cloud network. The proposed technique comprises several nodes. An expert node will be chosen from these nodes according to the loss rate and minimal completion time. In this situation, the master node fixes the threshold value. This threshold value involves two parameters. One parameter is the failure ratio while highest completion time is the other parameter. The expert node chooses those nodes as the candidate nodes that have equivalent or less failure rate and minimum execution time. The node N1 has lesser value as compared to the threshold value; therefore, it is selected as a candidate node. Node N2 comprises one lesser and other higher parameter thus it cannot be selected as a candidate node. The node N3 comprises value equivalent to the threshold, therefore, this node is chosen as a candidate node. Yet again, node N4 comprises higher value than the threshold value, as a result, this node so it cannot be selected as candidate node. Candidate node starts its functioning after selection. In this case, several tasks are implemented. During the completion of the task, one node can move from its location and this result in task failure. A novel methodology has been proposed to eliminate the issue of failure which occurs due to the movement of the node. A unique parameter is added in the proposed algorithm known as master node time It is the ultimate time that associate the end customers. It is used for node cooperation. The computational formulas for calculation master node time given below:

E-cost= maximum completion time + Time is taken by the expert node.. (eq1)

After that profit of each node is calculated based upon below equations:

Profit of each node = each node failure + E-cost+.. (eq2)

Burden of all nodes= No. of tasks + highest completion time/ Profit... (eq3)

The node having maximum weight is selected. Weight will be measured through above mentioned formula.

a) Computational Flowchart of Proposed Algorithm

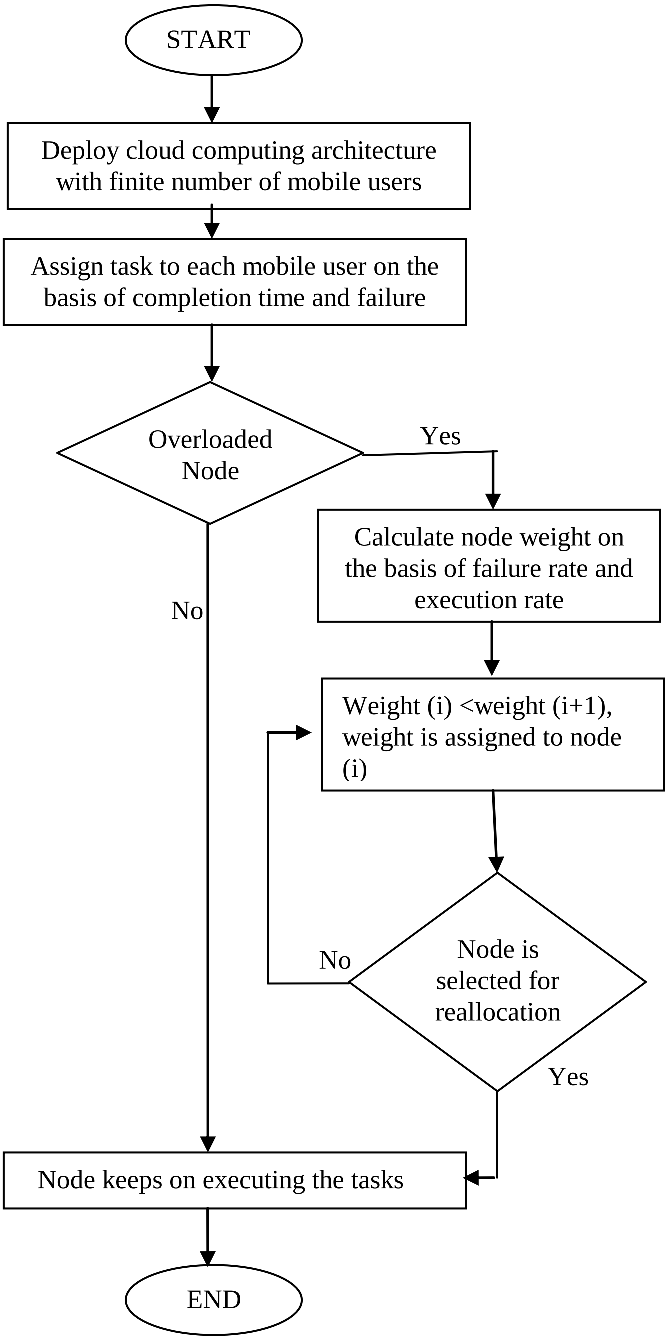

Fig 1. Flowchart of proposed system

Steps of the Intelligent Genetic algorithm are explained below:-

Step 1 Get record of all virtual machine those are active on all hosts.

Step 2 Initialize no movement is done.
Step3 Get asset consumption, failure ratio, and completion time of all system.
Step4 Create transition matrix for all virtual machines and hosts.
Step5 Loop will achieve until all machines on over used hosts are change
Step5.1 Determine the current consumption of each host for that specific virtual machine that needs migration.
Step5.2 Check creation history of the virtual machine.
Step5.3 Analyse growths in consumption of selected hosts with different hosts.
Step5.4 Select hosts for which increase in utilization is least End loop
Step5.5 if highest consumption surpasses upper use threshold go to step 5.1.
Ste 6 Else chooses that specific host for relocation.
Step7 return movement record

b) Pseudo Code of Intelligent Genetic Algorithm (IGA)
The pseudo code for proposed algorithm is given below:

1. Begin
2. Input: Virtual machine
3. Output: Task migration
4. Define Number of Tasks as TK
5. Threshold value of failure rate as FR
6. Threshold value of execution time as ER
7. Repeat while virtual machine are selected for the Task(TK)
8. If (FR of machine i> FR of machine i+1)
9. If (ER of machine i> ER of machine i+1)
10. Select i+1 as best machine
11. End if
12. End if
13. End of while
14. If (virtual i get overloaded=true)
15. Calculate weight ()
16. If (weigh of i >weight of i+1)
17. Select machine i for migration
18. Else
19. Select execute weight algorithm
20. End if
21. End

IV. RESULT AND DISCUSSION

The proposed algorithm is applied on Cloud Sim simulator due to the complexity involved in real time simulation. A comparison between the performances of earlier [14] and proposed approach is performed on the basis of power utilization and execution time and bandwidth. The simulation parameters are explain in the table 1

Number of VM	10
Number of cloudlets	60
Host Memory	2 GB
Processor	Xenon
Number of Data centers	5

Table 1: Simulation Parameters

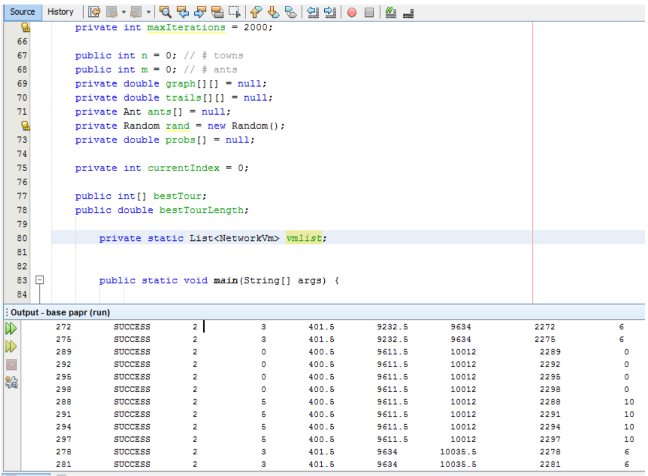

Fig 2: IGA Algorithm

As shown in figure 2, the IGA algorithm is applied for the migration of the virtual machine. The IGA algorithm transfer the task of the over loaded server to most reliable machine. The accuracy of the machine is determined on the basis of weight. The virtual machine which has maximum weight is considered as maximum reliable.

Table 2: Comparison of genetic algorithm and proposed algorithm based on some parameters:-.

Parameter	Genetic Algorithm [14]	Proposed Algorithm
Execution time	22.34 seconds	8.12 seconds

Space Utilization	13 buffers	5 buffers
Bandwidth consumption	120 mbps	80 mbps

Table 2: Comparison of genetic algorithm and proposed an intelligent genetic algorithm based on some parameters.

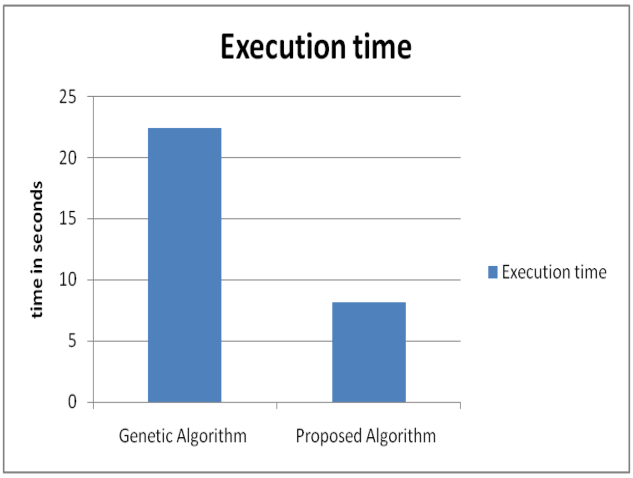

Fig 3: Execution Time

As shown in figure 3, the completion time of the genetic algorithm is compared with the proposed algorithm. Due to virtual machine migration proposed approach has short execution time as compared to the genetic algorithm.

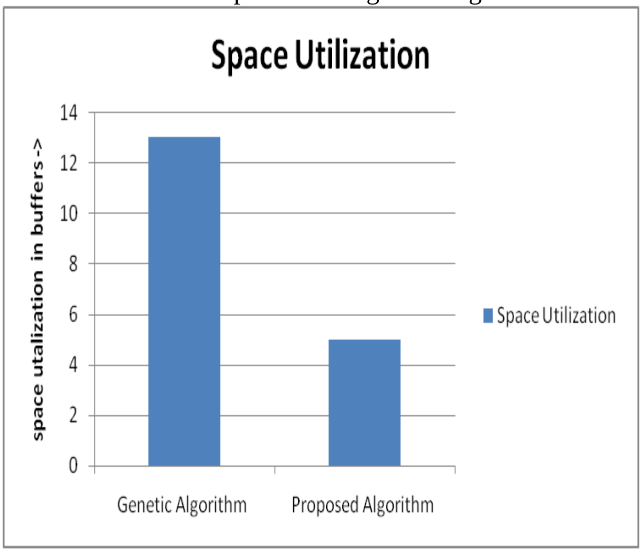

Fig 4: Space Utilization

As shown in figure 4, the space utilization of the proposed technique is correlates with existing genetic algorithm. It is analyzed that the proposed algorithm has low space utilization as compared to the genetic algorithm.

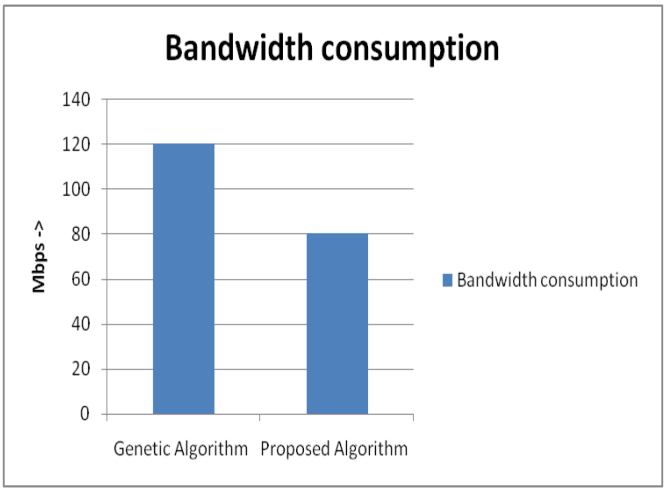

Fig 5: Bandwidth Comparison

As shown in figure 5, the bandwidth of the proposed algorithm is compared with the genetic algorithm for virtual machine migration. The proposed algorithm has low bandwidth consumption as compared to existing genetic algorithm

V. CONCLUSION AND FUTURE WORK

In this work, it is concluded that load balancing is the main issue of cloud architecture. The load balancing can increase latency in the network. The genetic algorithm is applied in the previous research work for virtual machine migration. We have presented and simulated our proposed algorithm in cloud environment using cloud sim and the experiment results have demonstrated that proposed approach performs well and can leads to a substantial improvement in terms of space utilization, bandwidth consumption and execution time when compared with earlier used genetic algorithm approach. In the future, we will combine the energy-efficient aspect in our approach for improving its further performance.

REFERENCES

1. J. srinivas, k.venkata subba reddy and a.moiz qyser (2012). Cloud computing basics. International *Journal of Advanced Research in Computer and Communication Engineering*, 1(5).
2. Pankaj, s. (2013). Cloud Computing: Types, Architecture, Applications, Concerns, Virtualisation and Role of IT Governance in Cloud. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(3).
3. Sidra, A. and Munam Ali, S. (2015). Load balancing algorithms in cloud computing: A survey of modern techniques. *National Software Engineering Conference (NSEC)*.
4. D, S. and L, S. (2015). Load Balancing Algorithms in Cloud Computing: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(7).
5. R.S., S. and Biradar Rekha, Y. (2017). Load Balancing and its Algorithms in Cloud Computing: A Survey. *International Journal of Computer Sciences and Engineering*,, 5(1).
6. Foram, F. and Jignesh, V. (2014). Load Balancing in cloud computing. *IJEDR*, 2(1).
7. Jargalsaikhan, N., Hannie, Z. and Hyuk, L. (2018). Service-Aware Cloud-to-Cloud Migration of Multiple Virtual Machines. *IEEE*, 06, pp.76663 – 76672.
8. Fei, Z., Guangming, L., Xiaoming, F. and Ramin, Y. (2018). A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues. *IEEE Communications Surveys & Tutorials*, 20(2).

9. Uttam, M., Pulak, c., Massimo, T., Charles U, M. and Biswanath, M. (2018). Bandwidth Provisioning for Virtual Machine Migration in Cloud: Strategy and Application. *IEEE Transactions on Cloud Computing*, 6(4), pp.: 967 – 976.
10. Deshpande, U., Danny, c., Steven, c., Kartik, G. and Nilton, V. (2018). Scatter-Gather Live Migration of Virtual Machines. *IEEE Transactions on Cloud Computing*,, 6(1), pp.196 – 208.
11. Weizhan, Z., Yuxuan, C., Xiang, G., Zhichao Mo Qinghua, Z. and Zongqing, L. (2017). Cluster-Aware Virtual Machine Collaborative Migration in Media Cloud. *IEEE Transactions on Parallel and Distributed Systems*, 28(10), pp.2808 – 2822.
12. Fei, T., Chen, L., T. Warren, L. and Yuanjun, L. (2016). BGM-BLA: A New Algorithm for Dynamic Migration of Virtual Machines in Cloud Computing. *IEEE Transactions on Services Computing*,, 9(6), pp.910 – 925.
13. Jaspreet, s. and Deepali, G. (2017). An Smarter Multi Queue Job Scheduling Policy for Cloud Computing. *International Journal of Applied Engineering Research ISSN*, 12(9), pp.1929-1934.
14. Monika, L. and Bhardwaj, N. (2017). Load Balancing in cloud computing using Genetic Algorithm. *International Conference on Intelligent Computing and Control system ICICCS*.
15. Jaspreet, S. and Deepali, G. (2017). Towards Energy saving with smarter Multi Queue Job scheduling Algorithm in Cloud Computing. *Journal of Engineering and Applied Science*, 12(10), pp.8944-8948.

AUTHORS PROFILE

Er. Neha Sharma is a student of M.tech in computer science and engineering from Chandigarh University, Gharuan Mohali, Punjab, India. I have done my B.tech from Global Institute of engineering & management Amritsar. Her research area of interest is Cloud Computing.

Dr. Jaspreet Singh is working with Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, (Punjab) India. He received M.tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, Punjab, India and obtained Ph.D. degree from Department of Computer Science and Engineering, Maharishi Markandeshwar University, Ambala, (Haryana),India. He also has published several research papers in national as well as international Journals and conferences of repute. He has research contributions in the area of cloud computing, networking, security and cryptography and image processing.

The author has done his M. Tech. from Kurukshetra University in year 2012. Since then he has played an active role in research and technology. He has contributed in number of Research article on Networking, Load Sharing, Delay Tolerance, Vehicular Networks, Fog computing and Edge Computing. The author has bagged few SCI, Scopus and UGC Approved Article in his count. He has internationally published 3 books with LAP with an indexing on WOS and SCI. He has written few Book Chapters with IGI Global with an Indexing on SCI and WOS. Along with this the author has 2 patents filed and published to his account. Finally he has 2 nationally published books. His future work is focused on Fog and Edge Computing.