International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Dynamic Load Balancing Ant Colony Optimization (DLBACO) Algorithm for Task Scheduling in Cloud Environment

Arvinda Kushwaha , Mohd Amjad, Arvind Kumar

Abstract: ***Cloud computing is a framework which provides on-demand services to the user for scalability, security, and reliability based on pay as used service anytime & anywhere. For load balancing, task scheduling is the most critical issues in the cloud environment. There are so many meta-heuristic algorithms used to solve the load balancing problem. A good task scheduling algorithm should be used for optimum load balancing in cloud environment. Such scheduling algorithm must have some vital characteristic like minimum makespan, maximum throughput, and maximum resource utilization, etc. In this paper, a dynamic load balancing and task scheduling algorithm based on ant colony optimization (DLBACO) has been proposed. This algorithm assigns the task the VM which has highest probability of availability in minimum time. The proposed algorithm balances the whole system by minimizing the makespan of the task and maximizing the throughput. CloudSim simulator is used to simulate the proposed scheduling algorithm and results show that the proposed (DLBACO) algorithm is better than the existing algorithms such as FCFS, LBACO (Load balancing ACO), and primary ACO.***

Keywords: CloudSim, Task scheduling, cloud environment, ACO, load balancing.

I. INTRODUCTION

A process which is running on a remotely located computer system instead of a user personal computer is known as Cloud computing. Cloud computing is an environment which provides infrastructure as a services (IaaS) for sharing infrastructure, platform as a service (PaaS) for sharing platform and software as a service (SaaS) for sharing software among multiple users [1-3]. The data stored on the cloud environment is cloud data, services provided by remote machines is cloud services, etc. Cloud computing offers many attractive services/ applications to the people to switch from the mono-centric computing system to the cloud computing system.

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author
Arvinda Kushwaha*, Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India. Email: arvindakush@gmail.com
Mohd Amjad, Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India. Email: mamjad@jmi.ac.in
Arvind Kumar, Department of CSE, KEC, Ghaziabad, India. Email: arvinddagur@gmail.com

Cloud computing provides unlimited access/ unlimited storage/ unlimited data processing, etc. at meager and optimized cost. Due to low cost and unlimited access, cloud data center (CDC) becomes highly dynamic. The load in the CDC is unbalanced due to random and unpredictable client request [4-5].

Cloud provides virtual machines for computation to each user. Many virtual machines can be assigned to a single user. The assignment of virtual machines to each user is not possible manually, and it is challenging. To assign virtual machines, we need an optimum task scheduling algorithm for cloud computing [5].

The user request for virtual machines is very unpredictable and unstable due to which a CDC becomes overloaded or under-loaded (idle). The performance of computation degrades and violation of the agreement. The unbalanced load on CDC may decrease throughput, reduced resource utilization, and increase power consumption & response time. Improper and uneven task scheduling is the main reason which can be addressed by applying a suitable and optimal scheduling algorithm [6-7].

A scheduling algorithm is said to be the best scheduling algorithm if they change their scheduling strategy according to system requirement like environment change as well as the type of task. For the same purpose, an optimum scheduling algorithm is required. Ant colony optimization (ACO)[8-9] algorithm has been used which is best for cloud computing environment.

ACO algorithm is a meta-heuristic search algorithm [10].This algorithm work like real ant colony. They search their food and connect during traveling based on their pheromone lay down on the path. There are many problems such as traveling salesman (TSP) problem [10], routing problem in vehicles [12] and graph coloring problem [11], all are solved by ACO algorithm.

We proposed a dynamic load balancing Ant colony optimization (DLBACO) algorithm to optimize the QoS parameter of the cloud system. The proposed algorithm is based upon Ant Colony Optimization technique to optimize the schedulability of tasks. We improved the existing LBACO (Load balancing ACO) and primary ACO (Ant colony optimization) and compared the results. The results presented in this paper show that the proposed algorithms perform better than the same of existing LBACO and ACO approaches.

A. Some Important Definitions

Definition 1: User accesses services such as platform, software or infrastructure for his use as the end user. The user may be a human, or a machine which use the services.
Definition 2: CIS (Cloud Information Service) is a registry used to register each datacenter entity and datacenter broker entity. It is also used to discover resources.
Definition3: Data center is a network of computers which can provide the facility to user to store, process, organize, and distribute a large amount of data.
Definition 4: Data center broker is accountable for intermediating between SaaS and cloud providers. It is work as decider which task execute first and in which VM.
Definition 5: Host is a physical server in cloud computing which provides services.
Definition 6: Virtual Machine (VM) is a software program or operating system. It can work as an individual computer.
Definition 7: VM allocation refers to allocation of VMs to physical hosts.
Definition 8: VM scheduler is implemented by a host, and It has policies like time-shared, space-shared to allocate processing power to VMs.
Definition 9: cloudlet is a prototype for cloud-based application services.
Definition 10: cloudlet scheduler determines the share (policies: times-shared, space-shared) of processing power between cloudlets in a VM.

B. CloudSim Architecture

CloudSim architecture is shown in figure 1.It shows that all users have an independent task. We consider there are n users such as user1, user2 ….user n. All users having p separate task, and they are represented as T_1, T_2 …T_p. A user can schedule the tasks which are distributed to q number of VMs. There are q virtual machines with n data centers. Each data center may contain any number of hosts. We consider there are m hosts for each data center. A CIS provides data center broker, Virtual machine scheduler, and VM allocation policy. Each data center broker is responsible for task scheduling, which communicates to each other. VM Scheduler delivers to each data center broker, virtual machines, and VM allocation policy. VM policy decides the scheduling ability of tasks on each virtual machine. To optimize the task scheduling, an efficient algorithm is applied and improve the quality of services.
Organization of the Rest part of this paper is given as follows: Section 2 represents the related work, in section 3, the problem has been described. The proposed DLBACO Algorithm has been included in section 4. Simulation results are shown in section 5 and finally, section 6 conclude the paper with future work.

II. RELATED WORK

Many researchers have been worked for load balancing in the cloud environment to enhance cloud performance. Most of the researchers worked on the basis to two primary criteria, first is system aware virtual machine migration, and second is network aware virtual machine migration. A prediction based load balancing algorithm has been presented. This algorithm was based on a live virtual machine migration approach. Gao et al.[13] proposed the Virtual Rank algorithm, which determined when and where to migrate virtual machines. The migration of these virtual machines predicted load tendency and migrated on the basis of weighted probability and Markov stochastic process method.
The responsiveness of the cloud data center is improved by dynamic resource allocation. Optimal data center configuration has been achieved by live VM migration. The proposed algorithm minimized the migration time and migration overhead [14].
The migration problem of tasks has been proposed by Chandrasekaran et al. [15] and finds the optimized solution for load balancing in cloud. The authors proposed the genetic algorithm for load balancing and results compared with existing algorithms such as Round Robin and Greedy, etc. A dynamic management algorithm (DMA) approach was developed by Gunjan et al. [16]. In this approach, the residual capacity of a physical machine is considered and used when the resource usages violate the threshold of a virtual machine. An algorithm for cluster-wide load balancing distributed resource scheduler (DRS) algorithm has been proposed. The authors [17] used VMware distributed resource scheduler cluster and conducted the experiments were conducted to check the performance. Through experimental results, authors observed that the proposed algorithms worked for system level resource constraint.
Modified Central Load Balancer (MCLB) has been proposed by sheetyet all in 2019 [22] for load balancing in virtual machines. In this algorithm job allocation is done based on priority and state of the virtual machine. This algorithm has been simulated on CloudSim and compared with existing algorithms. It has been observed that the MCLB perform better than that the same of FIFO and Round Robin algorithms.
A problem for migrating virtual machine storage also has been designed. Authors in [18-19] proposed the algorithms and found the solution to migrate the virtual machines. There are some other approaches which have been proposed for improving the VM migration. [20-21].
VM migration technique is generally used to solve the load balancing problem in cloud environment. In migration technique VMs are migrated at host level and ensured that all machines are running approximately at equal load and load is balanced [23]. Two types of migration were developed, first was cold migration in which virtual machines are suspended before migration. It was easy and simple to implement but caused much service downtime. Second migration approach i.e. live migrations were developed to overcome such problem. Live migration approach also has been divided into two categories pre-copy and post-copy. These migration techniques improves system level performance and overload network performance [24][25].

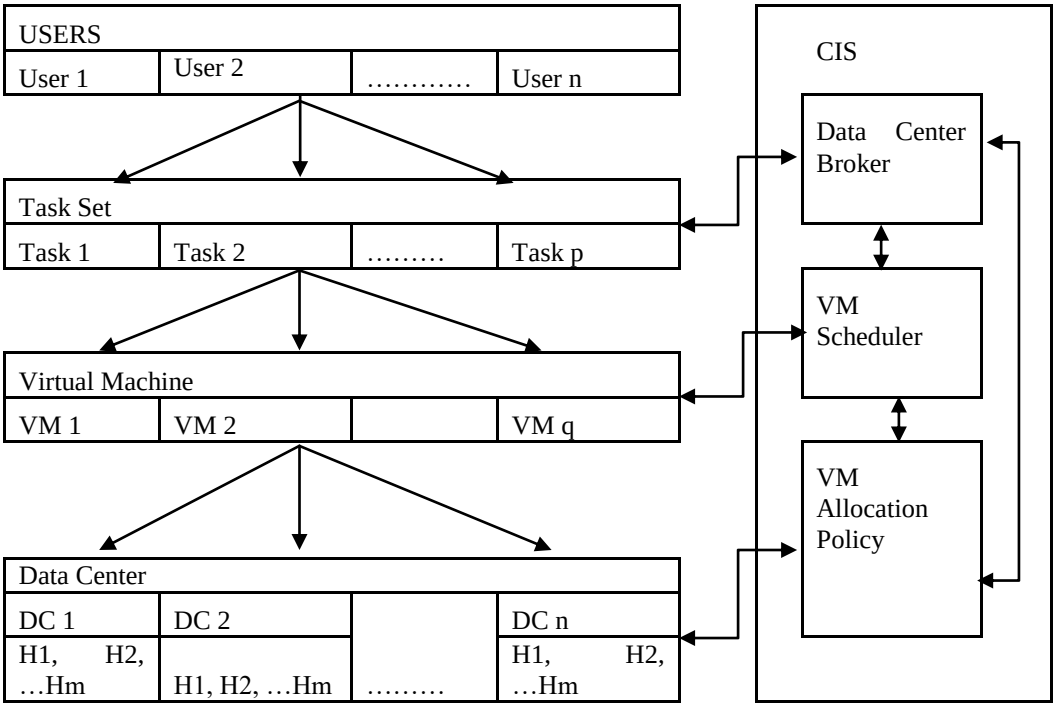

Figure 1: Cloud Architecture

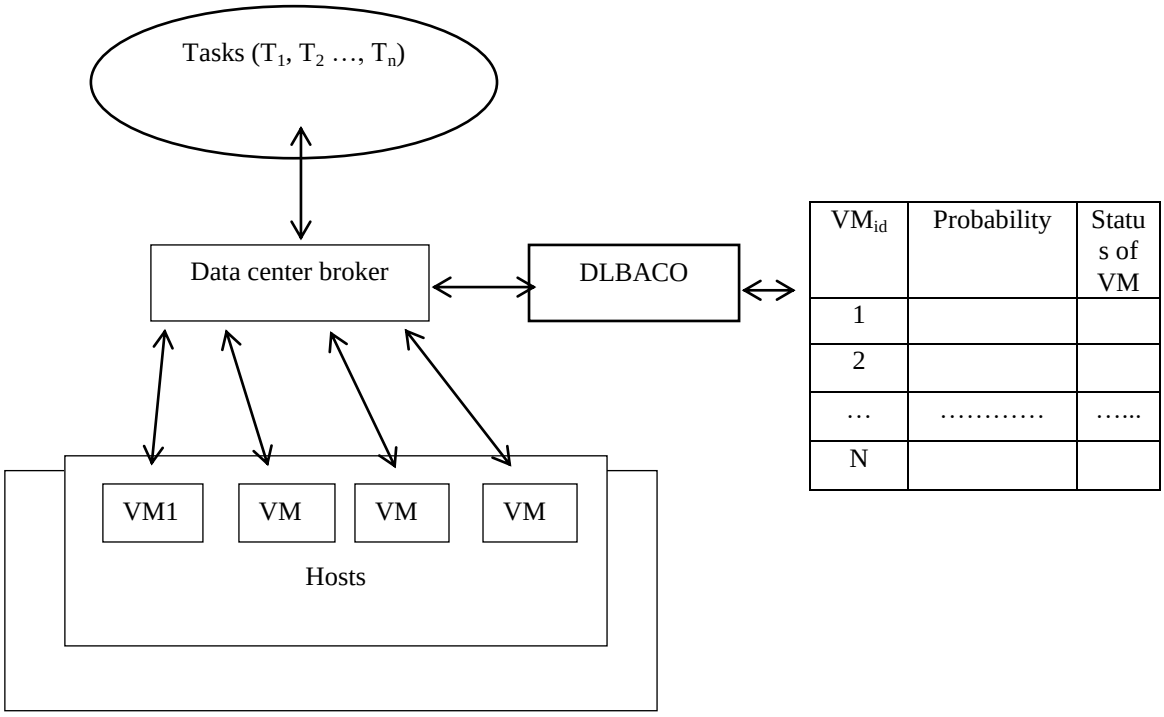

Figure 2: DCB model for the proposed DLBACO algorithm

In [2], authors designed a new approach for even distribution of VMs using VM migration technique. In this approach the cloud data center was modeled such that a random walk initiated from overloaded to migrated VMs. This approach migrated only single VM at a time.

There are some other approaches designed for network aware VM migration [22-25]. Many authors have designed the various approaches to enhance the throughput of overloaded machines. The work also has been done for reducing the number of migration and traffic of the network. The authors also found another solution of migration to enhance the throughput. Ant colony optimization technique has been applied to schedule the tasks for proper load balancing which was not so efficient for data center.

To enhance the ant colony optimization, authors proposed a new approach in which load has been balanced. The load balancing is achieved by proper scheduling. The authors also proposed dynamic load balancing ant colony optimization approaches for load balancing in cloud environment [20][21].

III. PROBLEM FORMULATION

Cloud computing is used as a metaphor of Internet. It is also a distributed computing and is dynamic. It's challenging to coincide between scheduled task actual execution time and expected execution time in the real cloud environment. Therefore, the ultimate challenge is the task scheduling in a virtual machine to balance the load at VM. The other problem is in creating the difference between the actual execution time and expected execution time in the proposed algorithm.

Met heuristic algorithm will be used to solve the task scheduling algorithm to balance the cloud environments. There is some assumption regarding task and machines: here we consider n task such that T_1, $T_2 \ldots T_n$ and these task scheduled on m virtual machines VM_1, VM_2 …VM_m.

Inspired by the above mention fact, the objective of this paper is to optimization of resource of cloud computing. We proposed a task scheduling algorithm for load balancing based on ant colony optimization algorithm that can optimize resource selection technique, which can find the resource to perform the task in optimal time and maximize the performance of the system.

IV. PROPOSED ALGORITHM AND ARCHITECTURE

In this proposed algorithm, we utilize the basic features of ACO for scheduling the tasks [2, 27]. We can design a new task scheduling algorithm experiences of previous task algorithm because they are very helpful in developing a new algorithm.

The Datacenter broker (DCB) model for the proposed DLBACO algorithm has been shown in figure 2, in which all tasks arrive at DCB for processing.DCB forwards the requests to the DLBACO algorithm for assigning the tasks to the available virtual machines.DLBACO algorithm maintains a table which has three components they are Id of virtual machine (VM_{id}), probability of virtual machine (P_{vm}) and status of the virtual machine. The DLBACO algorithm searches the table to find the virtual machine which is free or which has the highest probability to free as earliest to assign the task. If the free virtual machine is located then searching is complete and acknowledge to DCB with the id of that virtual machine (VM_{id}) and the DCB assign the task to the VM, else it will wait to be free of the virtual machine and once the status of the virtual machine is free then assign the task.

A. Flow chart for proposed DLBACO algorithm

The flow chart of proposed DLBACO algorithm shown in figure 3.Sequence of the algorithm is initializing with a start then request is generated from tasks group .when tasks arrived at DCB then it forward task to the DLBACO algorithm, the function of DLBACO algorithm is to find the free VM on the basis of highest probability at that time. If it found the free VM assign the task by the DCB else wait for free the VM.

B. Initialization of Pheromone

At the starting, all ants are distributed randomly on VM, and then we find out the pheromone value of VM_i by the following formula

$\rho_i(o)=\mu_i * \Phi_i + BW(VM_i)$

Where μ_i is a number of VM_i processor, Φ_i processor speed in MIPS of every processor in VM_i, BW (VM_i) is communication bandwidth concerning the capacity of VM_i.

C. VM selection Method for next task

The virtual machine VM_i is selected for n-ant for next task based on probability that is as follows:

$$P_i^n(t)=\begin{cases} \dfrac{[\rho_i(t)]^{\alpha}[W_i]^{\beta}[L_i]^{\gamma}}{\sum[\rho_n(t)]^{\alpha}[W_n]^{\beta}[L_n]^{\gamma}} & \text{if } i \epsilon\ 1\ldots..n \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Where

- α is the relative importance of pheromone trails
- β is the relative importance of computing capacity of VM
- γ is the relative impotence of load balancing factor on VM
- n is the number of VM
- ρ (t) is the initial pheromone value of VM_i at time t.

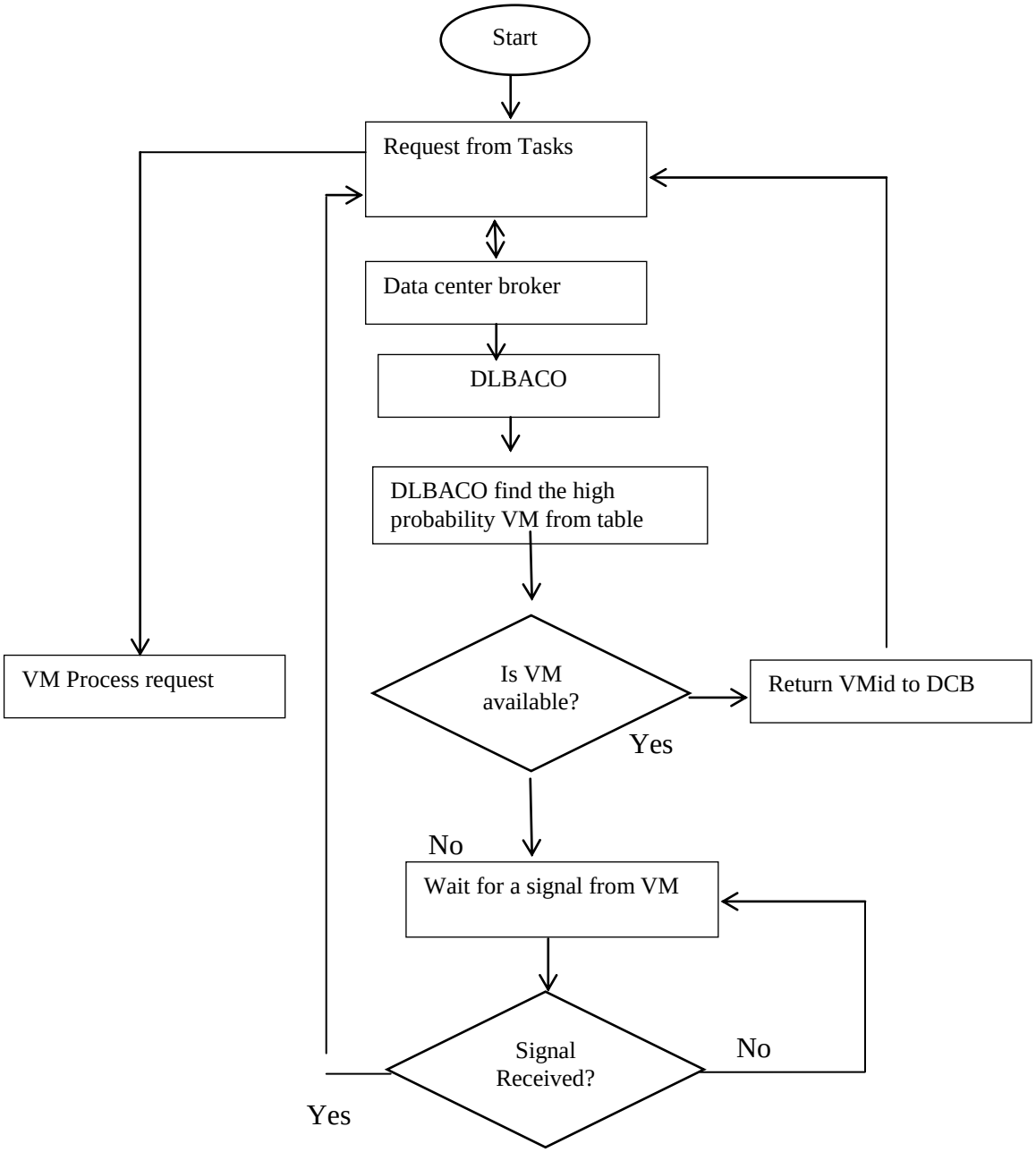

Figure 3 Flow chart for the proposed DLBACO algorithm

- W is the capacity of each VM that can be computed as

$W_i = \sqrt{((\rho_m X\ S_m\ X\ R_m) + (\rho_c X\ S_c X\ R_c))}$

Where

- ρm and ρc are proportional constants, and their addition should be one that is $\rho m + \rho c = 1$
- S_m is the memory size of VM
- R_m is the ideal rate of every VM memory
- R_c is the rate of every CPU of VM
- S_c is the speed of CPU in MHz
- L is load on Virtual Machine (VM) that can be computed by the following formula:

$\rho_i(t+1) = (1-\mu)\ \rho_i(t) + \Delta\ \rho_i$

$L_i = N(t)/E(t)$

Where

- N (t) is the number of tasks at VM_i at time t
- E (t) is the time required to execute the task at time t

D. Pheromone updating

Assume that $\rho_i(t)$ be the intensity of pheromone of VM_i at time t. We can update the pheromone by the following formula

Where μ is the pheromone trail decay coefficient (0< μ<1).if the value of μ is higher then the previous solution impact is less.

$\Delta \rho_i$ Can be computed by the following formula

$$\Delta \rho_i = 1/d_{jn}$$

Where d_{jn} is the shortest distance that is found by an n^{th} ant in the j^{th} iteration.

DLBACO Algorithm

Algorithm 1:

Proposed DLBACO Algorithm
I/P: Request for VM to assign the task
O/P: Receive ID of VM and assigned the task

```
Do
        {
        For (i=0; i<n;i++)
        {
        Check the status of VM
        If(VM=Free)
        Compute the probability of as per equation 1
        Else
        Set probability is 0
        Endif statement
        Assign probability
        }
        }
        While (there exists the request for VM)
```

V. SIMULATION RESULTS

The CloudSim toolkit package has been used to perform the simulation of the proposed algorithm. The simulation has been performed for the comparison of following existing scheduling algorithms such as FCFS, primary ACO, and LBACO with proposed DLBACO algorithm.

A. System Model

The following assumptions have been used for simulation.

- All tasks are a periodic and homogeneous.
- There are no dependencies and precedence constraints between tasks.
- All tasks are computationally exhaustive.
- All tasks are non-primitive, and they are not interrupted during execution.

The goal of this scheduling algorithm is to optimize the total execution time of tasks, and another most important goal is to maintain the balance of load among all VMs. In this work, we consider two factors: first, optimization of tasks execution time and second evenly distribution of workload among all VMs.

B. Simulation Environment

The following assumption has been made to simulate the proposed algorithm. We used CloudSim toolkit package for experiment and simulation. The experiment has been conducted on Intel® Core™ i3-4005U CPU @ 1.70GHz × 4 machine with 3 GB RAM on Ubuntu 14.04LTS.Table 1 shows the parameters taken to evaluate and simulate the overall performance

Table 1 Simulation parameters

Parameter	Value
DC configuration	
Architecture of DC	X86
Operating system of DC	Linux
VMM	Xen
Host configuration	
Number of host	20-50
MIPS	2000
RAM	4GB
Storage	2TB
Bandwidth	1Gbps
Configuration of VM	
Number of VMs	50-250
Size	Varying
MIPS	1000
RAM	512MB
Bandwidth	5Mbps
Number of PEs	Varying
Number of Tasks	50-250

VI. EXPERIMENTAL RESULTS

Figure 4 shows the comparison of average makespan for various numbers of tasks simulated using existing FCFS, basic ACO, and LBACO with our proposed DLBACO algorithm. The graph shows that average makespan is always less in DLBACO as compared to the same of FCFS, LBACO and basic ACO.The minimum average makespan is always better show DLBACO is better than the same of FCFS, basic ACO and LBACO.

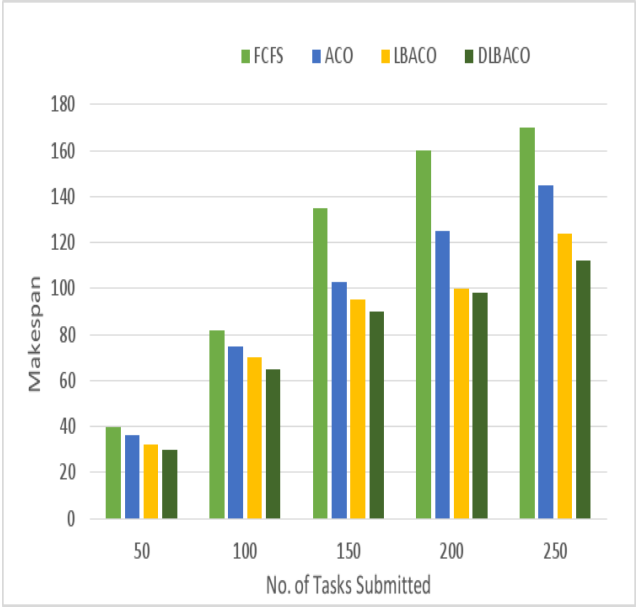

Figure 4: Comparison of makespan on different number of tasks

Figure 5 shows that when we increased the number of iteration then the average makespan has been reduced in all algorithms (FCFS, basic ACO, LBACO and DLBACO), but for the proposed algorithm DLBACO average makespan reduced fast after 80 iteration.

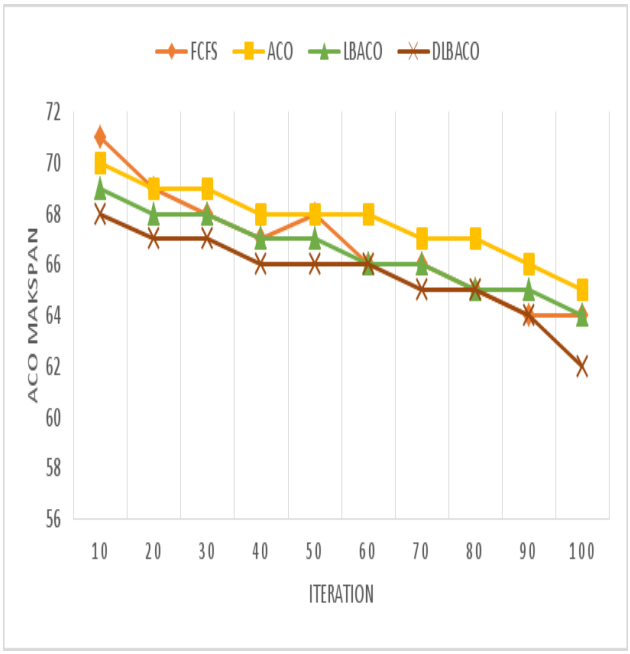

Figure 5: Effect of number of iteration on average makespan

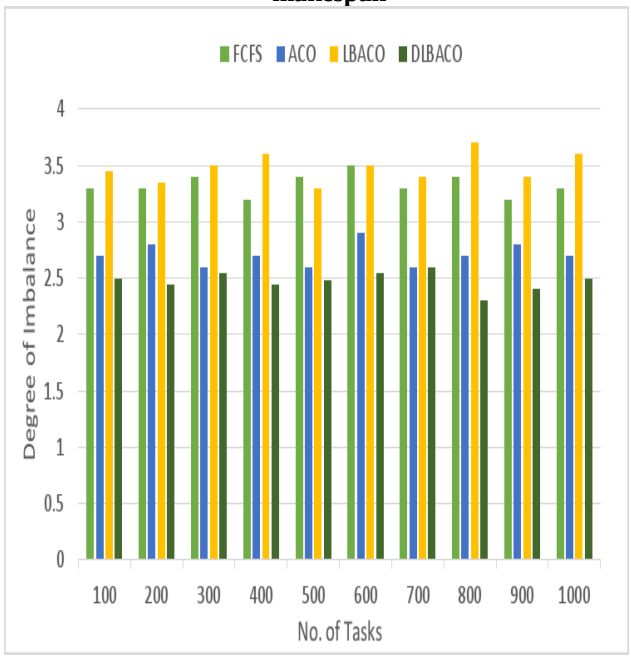

Figure 6: Comparison of degree of imbalance on different number of tasks

The degree of imbalance (DI) checks the imbalance among all virtual machines. DI of a task can be calculated as given below

$$DI = \frac{T_{max} - T_{min}}{T_{avg}}$$

Here T_{max}, T_{min} and T_{avg} are the maximum time, minimum time and average time along all Virtual Machine (VM).

The comparison of degree of imbalance of different number of processors has been shown in above figure 6. The figure shows the degree of Imbalance of various tasks. We have decided through the experimental results that the performance of our proposed algorithm in different aspects is better than that of other existing FCFS, ACO, and LBACO algorithms. We can be conclude from above graphs that the performance if our proposed DLBACO algorithm is better than the FCFS, basic ACO and LBACO algorithm.

VII. CONCLUSION

In this paper, a new dynamic load balancing ant colony optimization (DLBACO) algorithm for task scheduling in cloud environment has been presented. We designed a new algorithm and compare the results with existing algorithms such as FCFS, basic ACO and LBACO algorithms. We simulate the proposed algorithm using different parameters using CloudSim and then compared the results. We analyzed that the proposed approach perform better than the same of existing algorithms. Experimental results shows that using developed DLBACO algorithm, the makespan has been reduced approximately up to 15% when we increase the number of task as compared to FCFS, basic ACO and LBACO. The experimental results also show that the developed algorithm, DLBACO, reduces the makespan up to 20% on increasing the number of iterations when we compared with existing algorithms. The degree of imbalance also has been reduced 10-15% as compared to existing algorithms.

REFERENCES

1. Raj Kumar Buyya, Chee Shin Yeo, Srikumar Venugopal,James Broberg, and Ivona Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. 25, 6 (June 2009), 599-616.
2. Kumar, N., Agarwal, S.: Self-regulatory graph based model for managing VM migration in cloud data centers. In: 2014 IEEE International Advance Computing Conference (IACC) India, pp. 731–734. (2014)
3. Aruna, P., Devi, L. Y., Devi, D. S., Priya, N., Vasantha, S., & Thilagavathy, K. (2013). Private cloud for organizations: an implementation using OpenStack. International Journal of Scientific & Engineering Research, 4(10), 82.
4. Mahapatra, D., Saini, G. K., Goyal, H., & Bhati, A. (2016). Ant Colony Optimization: A solution of load balancing in cloud. International Journal of Engineering Applied Sciences and Technology, 1(3), 76-79.
5. Nipane, N. S., &Dhande, N. M. (2014). ABC—load balancing techniques—in cloud computing. Int. J. Innov.Res. Adv. Eng., 1(2), 2278–2311.
6. Liu, X. F., Zhan, Z. H., Deng, J. D., Li, Y., Gu, T., & Zhang, J. (2016). An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Transactions on Evolutionary Computation, 22(1), 113-128.
7. Khan, S., & Sharma, N. (2014). Effective scheduling algorithm for load balancing (SALB) using ant colony optimization in cloud computing. International Journal of Advanced Research in Computer Science and Software Engineering, 4(2)
8. Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. Theoretical computer science, 344(2-3), 243-278.
9. Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011, August). Cloud task scheduling based on load balancing ant colony optimization. In 2011 Sixth Annual ChinaGrid Conference (pp. 3-9). IEEE.
10. Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on evolutionary computation, 1(1), 53-66.
11. Salari, E., & Eshghi, K. (2005, December). An ACO algorithm for graph coloring problem. In 2005 ICSC Congress on computational intelligence methods and applications (pp. 5-pp). IEEE.
12. Zhang, X., & Tang, L. (2005, December). CT-ACO-hybridizing ant colony optimization with cyclic transfer search for the vehicle routing problem. In 2005 ICSC Congress on Computational Intelligence Methods and Applications Membership, achievements, with photo that will be maximum 200-400 words. (pp. 6-pp). IEEE.

13. Gao, Q., Tang, P., Deng, T., & Wo, T. (2011, July). Virtualrank: A prediction based load balancing technique in virtual computing environment. In 2011 IEEE World Congress on Services (pp. 247-256). IEEE.
14. K ochut, A., & Beaty, K. (2007, October). On strategies for dynamic resource management in virtualized server environments. In 2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (pp. 193-200). IEEE.
15. Chandrasekaran, K., & Divakarla, U. (2013). Load balancing of virtual machine resources in cloud using genetic algorithm. National Institute of Technology KarnataNa, Surath Nal, 156-168.
16. Khanna, G., Beaty, K., Kar, G., & Kochut, A. (2006, April). Application performance management in virtualized server environments. In 2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006 (pp. 373-381). IEEE.
17. Gulati, A., Shanmuganathan, G., Holler, A. M., & Ahmad, I. (2011). Cloud Scale Resource Management: Challenges and Techniques. HotCloud, 11, 3-3.
18. Devi, R. K., Murugaboopathi, G., & Muthukannan, M. (2018). Load monitoring and system-traffic-aware live VM migration-based load balancing in cloud data center using graph theoretic solutions. Cluster Computing, 21(3), 1623-1638.
19. Mashtizadeh, A. J., Celebi, E., Garfinkel, T., & Cai, M. (2011, June). The Design and Evolution of Live Storage Migration in VMware ESX. In USENIX Annual Technical Conference (pp. 187-200).
20. Tsakalozos, K., Verroios, V., Roussopoulos, M., & Delis, A. (2017). Live VM migration under time-constraints in share-nothing IaaS-clouds. IEEE Transactions on Parallel and Distributed Systems, 28(8), 2285-2298.
21. Chien, N. K., Dong, V. S. G., Son, N. H., & Loc, H. D. (2016, March). An efficient virtual machine migration algorithm based on minimization of migration in Cloud Computing. In International Conference on Nature of Computation and Communication (pp. 62-71). Springer, Cham
22. Shetty, S. M., & Shetty, S. (2019). Analysis of load balancing in cloud data centers. Journal of Ambient Intelligence and Humanized Computing, 1-9.
23. Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Xia, F., & Madani, S. A. (2015). Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. The Journal of Supercomputing, 71(7), 2473-2515.
24. Khanna, G., Beaty, K., Kar, G., & Kochut, A. (2006, April). Application performance management in virtualized server environments. In 2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006 (pp. 373-381). IEEE.
25. Wood, T., Shenoy, P., Venkataramani, A., & Yousif, M. (2009). Sandpiper: Black-box and gray-box resource management for virtual machines. Computer Networks, 53(17), 2923-2938.
26. P. Mell, T. Grance, The NIST definition of cloud computing (draft), NIST Spec. Publ. 800 (145) (2011) 7.
27. Reuters (SCI & Scopus) and conferences including IEEE and it's also available online. His research interest includes Network Security, Internet and mobile computing, Mobile Ad hoc Networks and wireless sensor networks. Dr. Amjad has more than 16 Years of teaching experience at U.G and P.G. Level.

AUTHORS PROFILE

Arvinda Kushwaha received the bachelor's degree in Computer Science and Engineering, from Bundelkhand University, Jhansi, Uttar Pradesh, India in 2002 and master's degree in Software Engineering from RGPV Bhopal, Madhya Pradesh, India in 2012. Currently he is pursuing his Ph.D. degree in computer engineering at Jamia Millia Islamia, Delhi, India. He has published more than 15 research papers in reputed international journals including Thomson Reuters (SCI & Scopus) and conferences including IEEE and it's also available online His research interests include, wireless sensor networks, and cloud computing.

Dr. Mohd. Amjad is currently working as Associate Professor in the Department of Computer Engineering, F/o Engineering & Technology, Jamia Millia Islamia (Central University), New Delhi. He received B.Tech. Degree from A.M.U. Aligarh in computer Engineering, M.Tech. Degree in Information Technology from GGSIP University New Delhi and Ph.D. from Jamia Millia Islamia, from the Department of Computer Engineering New Delhi. He has published more than 50 research papers in reputed international journals including Thomson