Indonesian Journal of Electrical Engineering and Computer Science
Vol. 27, No. 2, August 2022, pp. 1119~1126
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v27.i2.pp1119-1126

In-depth analysis of dynamic degree load balancing technique in public cloud for heterogeneous cloudlets

Aparna Shashikant Joshi[1,2], Shyamala Devi Munisamy[1]
[1]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India
[2]Department of Information Technology, Army Institute of Technology, Pune, India

Article Info

Article history:

Received Jan 31, 2022
Revised Jun 10, 2022
Accepted Jun 24, 2022

Keywords:

Bandwidth utilization
Cloud computing
CloudSim
CPU utilization
Load balancing
Memory utilization
Task allocation

ABSTRACT

Load balancing is one of the challenges of the distributed computing worldview. With the enormous development in clients and their interest for different administrations on the distributed computing stage, compelling or productive asset usage in the cloud climate has turned into an urgent concern. Load balancing is critical to keeping cloud computing running smoothly. This study examines the research using four scheduling algorithms: dynamic degree balance CPU based (D2B_CPU), dynamic degree balanced membership based (D2B_Membership), dynamic degree memory balanced allocation (D2MBA) and hybrid dynamic degree balance (HDDB) algorithm. Central processing unit (CPU) utilisation, bandwidth utilisation, and memory utilisation are used as performance measures to verify the performance of these algorithms. The CloudSim simulation programme was used to simulate these algorithms. The primary goal of this work is to aid in the future construction of new algorithms by researching the behaviour of various existing algorithms.

Corresponding Author:

Aparna Shashikant Joshi
Department of Computer Science and Engineering,
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
Chennai, 600062, India
Email: aparna.joshi82@gmail.com

1. INTRODUCTION

Load balancing is a procedure for conveying responsibility among numerous hubs in a specific climate, guaranteeing that no hub in the framework is over-burden or inactive at some random time [1]. A good load balancing algorithm ensures that each node in the system accomplishes roughly the same amount of work. The load balancing calculation's goal is to plan occupations that are shipped off the cloud space to empty assets, working on generally speaking reaction time and expanding asset usage [2]. Because we cannot forecast the number of requests issued per second in a cloud environment, load balancing has become one of the most important challenges in cloud computing. The cloud's ever-changing behaviour is responsible for the unpredictability. Load balancing in the cloud is primarily concerned with dynamically assigning load among nodes in order to meet user needs and maximise resource efficiency by distributing the total available load over multiple nodes [3], [4].

Load balancing seeks to improve user satisfaction. Because the number of users and their demands are growing by the day, the clouds should deliver services to clients that meet their highest expectations [5]. A good or ideal load balancing algorithm ensures that available resources are used efficiently and that no node is overloaded or underloaded. Load balancing enables for scalability, reduces bottlenecks, and speeds up response times [6]. Many load balancing methods have been created in order to schedule the load among multiple machines [7]. However, no perfect load balancing algorithm that distributes the load evenly across the system has yet to be developed. The problem of distributing work fairly across the system has been proven to be an NP-complete problem [8]. In existing system, there are various types of scheduling algorithm. The dynamic degree balance cpu based (D2B_CPU) based, dynamic degree balanced membership based (D2B_Membershp), dynamic degree memory balanced allocation (D2MBA) and hybrid dynamic degree balance (HDDB) algorithms are discussed in this paper, as well as their performance over various performance parameters such as central processing unit (CPU) utilisation, bandwidth utilisation, and memory utilisation. The following is the format of this paper: In section 2, related work is provided. Section 3 shows the simulation setup. Section 4 presents the algorithm's performance analysis, and section 5 summarises the task completed.

2. RELATED WORK

This section examines cloud computing scheduling techniques. Task duration, task deadline, and resource utilisation are all important factors to consider when evaluating scheduling algorithms. With these parameter CPU, Bandwidth and memory utilization are also important factor to consideres when evaluating scheduling algorithms.

Joshi and Devi [9] implemented the D2B_CPU based algorithm. The degree of imbalance and the time it takes for cloudlets to wait are improved using this algorithm. VMs are assigned to hosts with the least number of CPUs in use using this approach. Underloaded, laden, and balanced virtual machines are categorised according to their load.

Ghanbari and Othman [10] implemented priority-based job scheduling algorithm. This approach lowers a critical performance parameter, namely make span. Priorities are taken into account for scheduling in this algorithm. Each job makes a resource request based on its priority. Other performance characteristics, such as consistency and complexity, are taken into account in addition to make span.

Josh and Munisamy [11] implement (D2B_Membership) algorithm. The degree of imbalance, execution cost, throughput time, execution time, makespan, and CPU time all improve with this method. This approach tweaks the virtual machine and task allocation policies to find the best resource allocation. The value of a host's membership is used to impact VM allocation policy. The under and overutilization of virtual machines (VMs) was analysed in order to improve the work allocation policy.

Shokripour *et al.* [12] implemented modified round robin algorithm. The response time of this algorithm is reduced. It is based on a master-slave relationship that is maintained in divisible load balancing theory. The master processor divides jobs into smaller jobs, and VMs are set up. Smaller jobs are entrusted to VM to do. Jobs are dispatched to users after execution, and fresh jobs are assigned to the virtual machine for execution.

Xiao and Wang [13] implemented priority-based scheduling strategy for VM allocation. This method increases the service provider's benefit while also improving resource use. It suggests that virtual machines be scheduled based on priority. This technique assigns a ranking to requests based on how profitable they are. It has been discovered that using this method can boost the benefits.

Patel and Patel [14] used the GP algorithm (generalized priority-based algorithm). This strategy reduces the amount of time it takes to complete tasks. Virtual machines are prioritised according to their million instructions per second (MIPS), while jobs are prioritised according to their size and length. The highest-priority task is assigned to the highest-priority virtual machine. In this strategy, cloudlet size and priority are also used as scheduling factors.

Pawar and Kapgate [15] used priority based earliest deadline first scheduling algorithm. Combining two scheduling algorithms: earliest deadline first and priority-based scheduling, the average task waiting time is lowered. The major purpose of this strategy is to optimise resource allocation and memory usage.

Kaur and Kinger [16] implemented opportunistic load balancing algorithm. This strategy increases resource efficiency and productivity. This is a static load balancing approach that ignores the virtual machine's current workload. Using this method, unfinished jobs can be addressed in a random order.

Joshi and Munisamy [17] used D2MBA algorithm. Based on the host's random-access memory (RAM) and microprocessor without interlocked pipelined stages (MIPS), the algorithm allocates a virtual machine (VM) to the best-suited host, which subsequently assigns tasks to the best-suited VM based on the VM's balanced state. This method reduces the degree of imbalance, the cost of execution, and the time it takes to complete a task.

Joshi and Munisamy [18] used HDDB algorithm. The purpose of this study is to determine the best virtual machine hosts based on CPU availability and host membership value. To produce a hybrid technique capable of outstanding workload balance, the HDDB scheduling technique combines two algorithms: D2B_CPU based and D2B_Membership. This paper looked at the experimental and performance analysis of four algorithms: D2B_CPU, D2B_Membership, D2MBA and HDDB, taking into account performance parameters such as CPU utilisation, bandwidth utilisation, and memory utilisation.

3. SIMULATION SETUP

Because real testing limit experiments on the scale of the infrastructure, we used a CloudSim simulator to conduct the studies. Experiments are also more difficult to repeat [19]. The procedure of measuring the system's performance in a real-world cloud environment is time-consuming. To have access to a genuine cloud's infrastructure, you must pay a fee [20]. We used Facebook as a cloud-based social networking service on the internet for the simulation. According to the data published in [21], Table 1 shows the estimated distribution of Facebook users among the three major areas.

Table 1. User distribution (approximately)

Region	Users (approx in millions)
South America	277
Asia	893
Europe	323

We've assumed a system comparable to this, but on a normalised scale $(1/300)^{th}$. We've created three user bases to represent the people who live in the three locations mentioned [22], [23]. Table 2 represents user base characteristics. We've assumed that just 5% of users are online during non-peak hours for the sake of simplicity [24], [25]. Core i3-7100U CPU, 2.40 GHz, 4 GB RAM machine were used for experiment and Java programming language is used to write algorithm. Simulating was done by using the same VMs but different cloudlets.

Table 2. User base characteristics

Region	Time Zone (GMT)	Peak Hours	Average peak users	Average off-peak users
South America	GMT-4.00	15.00-17.00	273,000	27,300
Asia	GMT + 6.00	01.00-03.00	340,000	34,000
Europe	GMT + 1.00	20.00-22.00	210,000	21,000

4. METHODS

This section explains detail methodology of D2B_CPU based, D2B_Membership, D2MBA and HDDB algorithms:

4.1. D2B_CPU based method

This method assigns VMs to hosts based on the CPU usage of the hosts and assigns tasks based on the VMs' balancing status. The workload will be shifted to an underloaded VM if the VM is overloaded. The method D2B_CPU based is work in two phases viz. task allocation phase and VM allocation phase. The task allocation phase's main goal is to distribute the dynamic workload among all VMs in order to avoid resource underutilization or overutilization. VM allocation is carried out in the VM allocation phase by assigning VMs to a suitable host with a lower number of processors [9].

4.2. D2MBA

This approach assigns VMs to hosts based on the RAM and MIPS value of the host, and then assigns tasks based on the VMs' balancing condition. The workload will be shifted to an underloaded VM if the VM is overloaded. The D2MBA is work in two phases viz. task allocation phase and VM allocation phase. The task allocation phase's main goal is to distribute the dynamic workload among all VMs in order to avoid resource underutilization or overutilization. VM allocation is accomplished by assigning virtual machines to a suitable host with sufficient RAM and MIPS [17].

4.3. D2B_membership

This approach assigns VMs to hosts based on the membership value of the host and assigns tasks based on the VMs' balance state. Tasks will be shifted to an underloaded VM if the VM is overloaded. The D2B_Membership is work in two phases viz. task allocation phase and VM allocation phase. The task allocation phase's main goal is to distribute the dynamic workload among all VMs in order to avoid resource underutilization or overutilization. VM allocation is accomplished by assigning VMs to a suitable host with a capacity greater than the VM's requirement [11].

4.4. HDDB

Using the hybrid dynamic degree balance (HDDB) algorithm, this technique allocates VMs to Hosts based on the CPU availability of the Host and the membership value of the Host, as well as job allocation by considering the balance state of the VMs. To balance the overall workload in the cloud environment, the load balancing algorithm seeks to balance the system load by transferring workload from overcommitted resources to undercommitted resources. The goal of this research is to apply the HDDB algorithm to assign virtual machines to the best host based on CPU availability and host membership values. The proposed strategy proposes a hybrid method for balancing workloads that combines two algorithms: D2B CPU based and D2B Membership. The HDDB has been tested using the CloudSim simulation tool. The HDDB is work in two phases viz. task allocation phase and VM allocation phase. The task allocation phase's main goal is to distribute the dynamic workload among all VMs in order to avoid resource underutilization or overutilization. The VM is assigned to a host with a higher membership value and fewer processors [18].

5. PERFORMANCE ANALYSIS OF ALGORITHM

This section presents the output and inferences observed from the entire four scheduling algorithms. For the purpose of studying the behaviour of the four algorithms, two different scenarios are explored. Each scenario has its own set of datacentres (DCs), virtual machines (VMs), and broker policy. For examining the findings, the simulation is performed for D2B_CPU based, D2B_Membership, D2MBA and HDDB. The performance parameters are:

$$CPU\ utilization = \sum_{i=1}^{m} CPU\ utilization_i \quad (1)$$

$$Bandwidth\ utilization = \sum_{i=1}^{m} BW\ utilization_i \quad (2)$$

$$Memory\ utilization = \sum_{i=1}^{m} Memory\ utilization_i \quad (3)$$

In our investigation, we looked at two alternative scenarios with varying amounts of virtual machines: 500 VMs and 800 VMs. The input tasks taken are of the range; 1,000 to 5,000. This has been implemented using Java with the Cloudsim tool for simulation. A series of experiments were conducted on a desktop system. The results obtained shows that HDDB technique produces better performance when compared to D2B_CPU based, D2B_Membership and D2MBA algorithm.

5.1. Analysis of CPU utilization

CPU utilization of all the methods is compared and given, including D2MBA, D2B_CPU based, D2B_Membership based and HDDB. The number of virtual machines (VMs) used was between 500 and 800, and the number of cloudlets was between 1,000 and 5,000. Table 3 gives a value of the CPU Utilization of 500 VMs on varying number of tasks for all the four algorithms. It is observed that the HDDB algorithms have lowest values for CPU Utilization. Table 3 shows that the HDDB method reduces CPU use by 40.66% on average when compared to the D2B_CPU algorithm, 41.86% when compared to the D2B_Membership based algorithm and 11.73% on average when compared to the D2MBA approach. The changes in CPU utilisation as the number of cloudlets increased were plotted and presented in Figure 1.

For each of the four techniques, Table 4 shows the CPU Utilization of 800 VMs on a variable number of tasks. The HDDB algorithm is shown to have the lowest CPU Utilization values. The HDDB algorithm reduces CPU use by an average of 24.89% when compared to the D2B_CPU algorithm, 25.18% when compared to the D2B_Membership based strategy and 7.46% when compared with D2MBA approach, as shown in Table 4. Figure 2 depicts changes in CPU utilisation as a function of the number of cloudlets.

Table 3. CPU Utilization of 500 VMs

	CPU utilization				
No of tasks	1,000	2,000	3,000	4,000	5,000
D2MBA	2.1	1.89	1.6	1.2	1.56
D2B_CPU based	2.14	2.8	1.81	1.9	1.69
D2B_Membership	2.40	1.4	2.59	1.3	2.7
HDDB	2.1	1.4	1.45	1.2	1.5

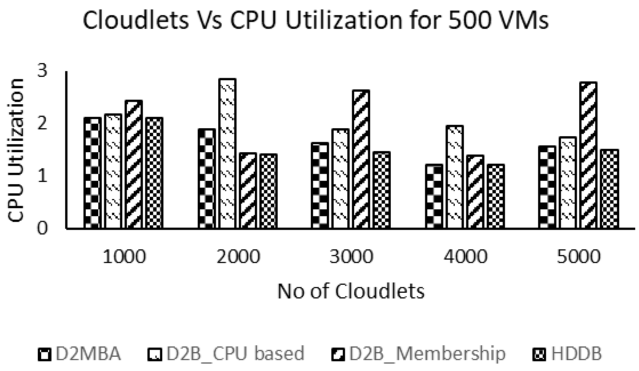

Figure 1. CPU utilization of 500 VMs

Table 4. CPU utilization of 800 VMs

	CPU utilization				
No of tasks	1,000	2,000	3,000	4,000	5,000
D2MBA	3.1	2.89	2.63	2.2	2.56
D2B_CPU based	3.11	3.8	2.8	2.9	2.7
D2B_membership	3.4	2.4	3.6	2.3	3.7
HDDB	3.1	2.39	2.5	2.1	2.49

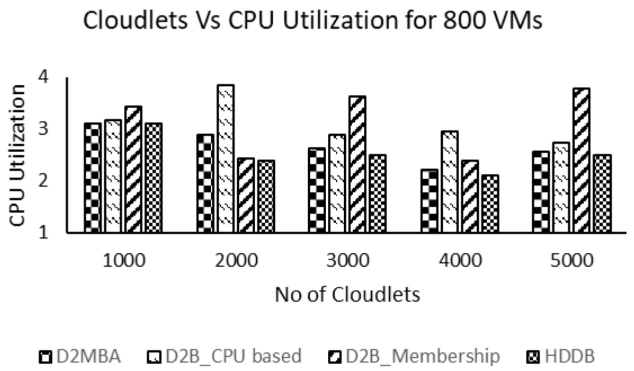

Figure 2. CPU utilization of 800 VMs

5.2. Analysis of bandwidth utilization

All four methods, D2MBA, D2B_CPU based, D2B_Membership based and HDDB, are compared and their bandwidth utilisation is provided. The number of VMs studied here was between 500 and 800, with the number of cloudlets ranging from 1,000 to 5,000. For each of the four techniques, Table 5 shows the Bandwidth Utilization of 500 VMs on a changing number of tasks. The HDDB algorithms are found to have the lowest Bandwidth Utilization values. The HDDB method reduces bandwidth use by an average of 11.49% when compared to the D2B_CPU algorithm, 8.4% when compared to the D2B_Membership based strategy and 7% when compared to the D2MBA approach, as shown in Table 5. Variations in Bandwidth Utilization were plotted and shown in Figure 3 as a function of the number of cloudlets.

Table 5. Bandwidth utilization of 500 VMs

	Bandwidth utilization				
No of tasks	1,000	2,000	3,000	4,000	5,000
D2MBA	2.71	2.43	2.34	1.01	1.13
D2B_CPU based	2.7	2.7	2.8	1.24	1.87
D2B_membership	2.8	2.47	2.4	1.01	1.21
HDDB	2.7	2.41	2.3	1	1.1

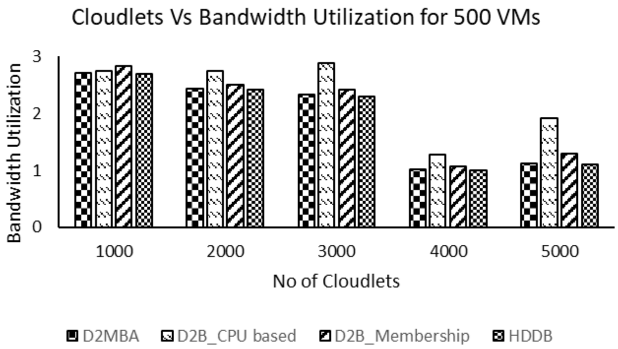

Figure 3. Bandwidth utilization of 500 VMs

Table 6. Bandwidth utilization of 800 VMs

	Bandwidth utilization				
No of tasks	1,000	2,000	3,000	4,000	5,000
D2MBA	3.71	3.43	3.34	2.01	2.13
D2B_CPU based	3.7	3.7	3.8	2.2	2.9
D2B_membership	3.8	3.5	3.4	2.07	2.29
HDDB	3.7	3.41	3.32	2	2.1

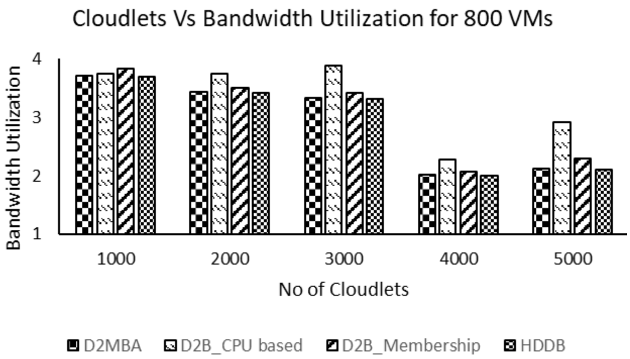

Figure 4. Bandwidth utilization of 800 VMs

5.3. Analysis of Memory Utilization

All four methods, D2MBA, D2B_CPU based, D2B_Membership based and HDDB, are compared and their memory utilisation is provided. The number of VMs studied here was between 500 and 800, with the number of cloudlets ranging from 1000 to 5000. For each of the four techniques, Table 7 shows the memory utilization of 500 VMs on a variable number of tasks. The HDDB algorithms are found to have the lowest memory utilization values. The HDDB method reduces memory use by an average of 12.08% when compared to the D2B_CPU algorithm, 8.44% when compared to the D2B_Membership based strategy and 4.8% when compared to D2MBA approach, as shown in Table 7. Variations in memory utilization were plotted and shown in Figure 5 as a function of the number of cloudlets.

Table 7. Memory utilization of 500 VMs

	Memory utilization				
No of tasks	1,000	2,000	3,000	4,000	5,000
D2MBA	1.89	1.43	2.17	1	1
D2B_CPU based	2.72	2.72	2.26	1.58	1.76
D2B_membership	1.93	1.49	2.88	1.01	1.91
HDDB	1.88	1.4	2.1	1	1

For each of the four techniques, Table 8 shows the memory utilization of 800 VMs on a changing number of tasks. The HDDB algorithms are shown to have the lowest Memory Utilization values. The HDDB algorithm reduces memory utilization by an average of 6.88% when compared to the D2B_CPU algorithm, 4.96% when compared to the D2B_Membership based strategy and 0.05% when compared to D2MBA approach, as shown in Table 8. Variations in memory utilization were plotted and shown in Figure 6 as a function of the number of cloudlets.

Table 8. Memory utilization of 800 VMs

	Memory utilization				
No of tasks	1,000	2,000	3,000	4,000	5,000
D2MBA	2.89	2.43	3.17	2	2
D2B_CPU based	3.72	3.72	3.26	2.58	2.76
D2B_membership	2.93	2.49	3.88	2.01	2.91
HDDB	2.87	2.4	3.12	2	1.9

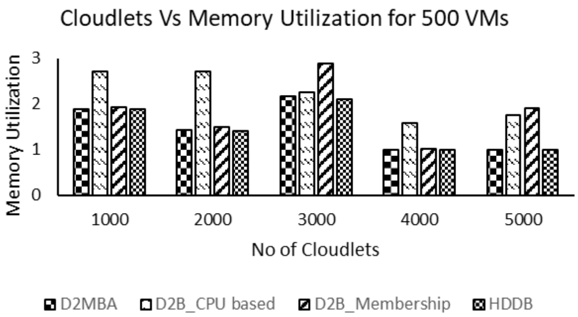

Figure 5. Memory utilization of 500 VMs

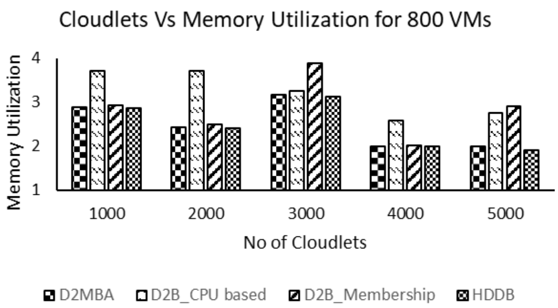

Figure 6. Memory utilization of 800 VMs

6. CONCLUSION

The performance of existing load balancing techniques such as D2B_CPU, D2B_Membership, D2MBA and HDDB algorithms is investigated in this study. To assess the system's performance, various performance parameters such as CPU utilisation, bandwidth consumption, and memory utilisation were employed. The simulation was run using the CloudSim simulation programme with a number of VMs between 500 and 800 and a number of cloudlets between 1,000 and 5,000. The findings of the analysis demonstrate that the HDDB algorithm has lower CPU, bandwidth, and memory consumption than the D2MBA, D2B_Membership and D2B_CPU based algorithms. As a result, the HDDB method outperforms the D2B_CPU, D2MBA and D2B_Membership algorithms in terms of performance.

ACKNOWLEDGEMENTS

Authors thanks to Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Chennai, India and Army Institute of Technology for providing an infrastructure to carry research on mentioned topic.

REFERENCES

[1] A. Singh and M. Hemalatha, “An approach on semi-distributed load balancing algorithm for cloud computing system,” *Int. J. Comput. Appl.*, vol. 56, no. 12, pp. 5–10, Oct. 2012, doi: 10.5120/8941-3083.
[2] G. Gupta, V. K. Kumawat, P. R. Laxmi, D. Singh, V. Jain, and R. Singh, “A simulation of priority based earliest deadline first scheduling for cloud computing system,” in *2014 First International Conference on Networks & Soft Computing (ICNSC2014)*, Aug. 2014, pp. 35–39. doi: 10.1109/CNSC.2014.6906659.
[3] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, “Load-balancing algorithms in cloud computing: a survey,” *J. Netw. Comput. Appl.*, vol. 88, pp. 50–71, Jun. 2017, doi: 10.1016/j.jnca.2017.04.007.
[4] T. Yeboah, P. O. I. Odabi, and C. A. A. Abilimi, “Utilizing divisible load scheduling theorem in round robin algorithm for load balancing in cloud environment,” *Comput. Eng. Intell. Syst.*, vol. 6, no. 4, pp. 81–91, 2015.
[5] X. Liu, B. Chen, X. Qiu, Y. Cai, and K. Huang, “Scheduling parallel jobs using migration and consolidation in the cloud,” *Math. Probl. Eng.*, vol. 2012, pp. 1–18, 2012, doi: 10.1155/2012/695757.
[6] C.-L. Hung, H. Wang, and Y.-C. Hu, “Efficient load balancing Algorithm for cloud computing network,” in *International conference on information science and technology*, 2012, pp. 28–30.
[7] S. S. Moharana, R. D. Ramesh, and D. Powar, “Analysis of load balancers in cloud computing,” *Int. J. Comput. Sci. Eng.*, vol. 2, no. 2, pp. 101–108, 2013.
[8] D. Fernandez-Baca, “Allocating modules to processors in a distributed system,” *IEEE Trans. Softw. Eng.*, vol. 15, no. 11, pp. 1427–1436, 1989, doi: 10.1109/32.41334.

[9] A. S. Joshi and S. D. Munisamy, "Dynamic degree balanced with CPU based VM allocation policy for load balancing," *J. Inf. Optim. Sci.*, vol. 41, no. 2, pp. 543–553, Feb. 2020, doi: 10.1080/02522667.2020.1724618.
[10] S. Ghanbari and M. Othman, "A priority based job scheduling algorithm in cloud computing," in *Procedia Engineering*, 2012, vol. 50, pp. 778–785. doi: 10.1016/j.proeng.2012.10.086.
[11] A. Joshi and S. D. Munisamy, "Enhancement of performance parameter of cloud using dynamic degree balanced with membership value algorithm," *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 8, pp. 664–676, 2020, doi: 10.34218/IJARET.11.8.2020.065.
[12] A. Shokripour, M. Othman, H. Ibrahim, and S. Subramaniam, "New method for scheduling heterogeneous multi-installment systems," *Futur. Gener. Comput. Syst.*, vol. 28, no. 8, pp. 1205–1216, Oct. 2012, doi: 10.1016/j.future.2012.03.008.
[13] J. Xiao and Z. Wang, "A priority based scheduling strategy for virtual machine allocations in cloud computing environment," in *2012 International Conference on Cloud and Service Computing*, Nov. 2012, pp. 50–55. doi: 10.1109/CSC.2012.16.
[14] S. J. Patel and U. R. Bhoi, "Improved priority based job scheduling algorithm in cloud computing using iterative method," in *2014 Fourth International Conference on Advances in Computing and Communications*, Aug. 2014, pp. 199–202. doi: 10.1109/ICACC.2014.55.
[15] A. Pawar and D. Kapgate, "A review on virtual machine scheduling in cloud computing," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 4, pp. 928–933, 2014.
[16] A. Kaur and S. Kinger, "A survey of resource scheduling algorithms in green computing," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 4886–4890, 2014.
[17] A. S. Joshi and S. D. Munisamy, "Enhancement of cloud performance metrics using dynamic degree memory balanced allocation algorithm," *Indonesia. Journal Electrical Engineering and Comput. Scince.*, vol. 22, no. 3, p. 1697, Jun. 2021, doi: 10.11591/ijeecs.v22.i3.pp1697-1707.
[18] A. Joshi and S. D. Munisamy, "Evaluating the performance of load balancing algorithm for heterogeneous cloudlets using HDDB algorithm," *Int. J. Syst. Assur. Eng. Manag.*, vol. 13, no. S1, pp. 778–786, 2022, doi: 10.1007/s13198-022-01641-1.
[19] A. Dhari and K. I. Arif, "An efficient load balancing scheme for cloud computing," *Indian J. Sci. Technol.*, vol. 10, no. 11, pp. 1–8, Mar. 2017, doi: 10.17485/ijst/2017/v10i11/110107.
[20] K. R. Jackson *et al.*, "Performance analysis of high performance computing applications on the amazon web services cloud," in *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, Nov. 2010, pp. 159–168. doi: 10.1109/CloudCom.2010.69.
[21] "Facebook Stats," *Internet World Stats Usage and Population Statistics*, 2017. http://www.internetworldstats.com (accessed May 25, 2022).
[22] E. Asano, "How much time do people spend on social media? [infographic]," *SocialMediaToday*, 2017. https://www.socialmediatoday.com/marketing/how-much-time-do-people-spend-social-media-infographic (accessed May 25, 2022).
[23] M. Keutelian, "The best times to post on social media in 2022," *Sproutsocial.com*, 2022. https://sproutsocial.com/insights/best-times-to-post-on-social-media (accessed May 25, 2022).
[24] V. Velde and B. Rama, "Simulation of optimized load balancing and user job scheduling using CloudSim," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, May 2017, pp. 1379–1384. doi: 10.1109/RTEICT.2017.8256824.
[25] H. Saleh, H. Nashaat, W. Saber, and H. M. Harb, "IPSO task scheduling algorithm for large scale data in cloud computing environment," *IEEE Access*, vol. 7, pp. 5412–5420, 2019, doi: 10.1109/ACCESS.2018.2890067.

BIOGRAPHIES OF AUTHORS

Aparna Shashikant Joshi is a research scholar at Department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Chennai, India. Author working as an Assistant Professor at Department of Information Technology, Army Institute of Technology, Pune. Authors area of interests is cloud computing. She can be contacted at email: aparna.joshi82@gmail.com.

Dr. Shyamala Devi Munisamy is working as a Professor at Department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Chennai, India. Her area of interests is cloud computing, Machine learning and image processing. She can be contacted at email: shyamalapmr@gmail.com.