Artificial Bee Colony Optimization-Based Load Balancing in Distributed Computing Systems—A Survey

Vidya S. Handur and Santosh L. Deshpande

Abstract Distributed computing allows the interoperability of components in distributed system in which software or hardware components located at networked computers coordinate and communicate their actions by message passing. The tasks in such a system are carried out independently. In distributed computing systems, load balancing is one of the issues, which is a means to distribute the tasks such that the computational nodes are neither overloaded nor underloaded, and the performance of the system is improved. Among the various solutions proposed for load balancing, metaheuristic-based algorithms are one of them. This paper discusses the variants and recent developments of artificial bee colony optimization algorithm for solving load balancing in distributed systems. As a result of load balancing, various performance metrics measured are throughput, response time, makespan, CPU utilization, memory utilization, and network utilization. The performance of the optimization algorithms is also measured with the time taken to converge in finding the optimal solution for load balancing.

Keywords Distributed computing systems · Load balancing · Artificial bee colony optimization · Convergence rate

1 Introduction

With the development in technology and increase in the data, getting optimal solutions to data intensive applications are a challenge. To overcome this, there is need for techniques that handle the data efficiently which in turn makes it necessary that all the resources being utilized appropriately. Resource management deals with managing the resources in such a way that all the resources are utilized efficiently to improve the performance of the system. Resource management is achieved by proper resource

V. S. Handur (✉)
KLE Technological University, Hubballi, India
e-mail: vidya_handur@kletech.ac.in

S. L. Deshpande
Visveswaraya Technological University, Belagavi, India

Y.-D. Zhang et al. (eds.), *Smart Trends in Computing and Communications*, Lecture Notes in Networks and Systems 396, https://doi.org/10.1007/978-981-16-9967-2_69

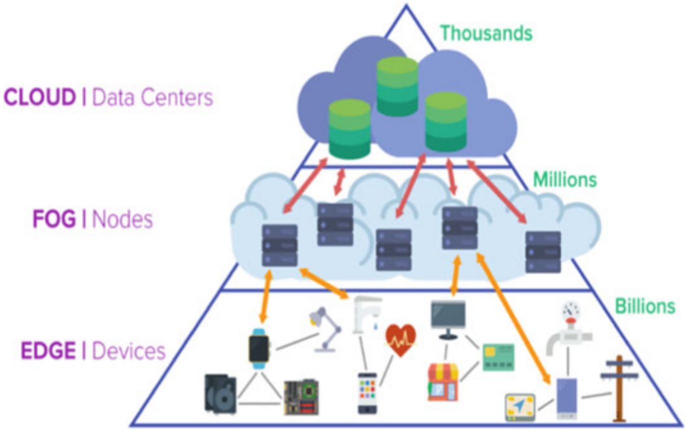

Fig. 1 Distributed system

allocation to the applications so to maximize the efficiency of the system. As the resource allocation varies with requirements of the users, there is a need to check for the availability of the resources and allocate. The resource allocation should be performed such that no resource is either underutilized or over utilized. Load balancing is a technique to discover the inefficiently used or over used resources and to balance the load among the resources proportional to their capacity [1, 2]. Load balancing is a major concern in a network that is distributed in nature. Figure 1 shows various computational nodes that communicate and coordinate in distributed manner to accomplish the task. The major concerns with distributed system are that all the computational nodes are autonomous, and there is no global clock. Other characteristics of distributed system include sharing of resources, unreliability of systems, openness, and heterogeneity of computational nodes. Due to the distributed characteristics, such systems have the advantage of better performance, availability, and scalability [3–5]. There are wide range of applications of distributed systems such as fog computing, edge computing, grid computing, cloud computing, World Wide Web, and automated banking systems.

With the increase in the technology and the data rates, there is need for techniques to handle the huge traffic to enhance the performance of the system. Also due to the differences in the computational capacities and heterogeneity of computing and network resources, the pattern of job arrival and the workloads on each node may vary significantly because of which the performance of the distributed system may deteriorate. Therefore, there is a need to distribute the workloads proportional to the capacity of the nodes. This imbalance of load can be addressed with load balancing.

The paper is systematized as follows: Sect. 2 briefs about load balancing, Sect. 3 introduces the artificial bee colony (ABC) optimization algorithm, Sect. 4 briefs the recent developments of the ABC algorithm for load balancing and followed by conclusion.

2 Load Balancing

Load balancing allows distributing the load on the system such that the computational nodes receive the load proportional to their capacity. The load balancing is either static or dynamic [6–8]. Load balancing is static when the load is assigned to the computational nodes without considering the current load of the nodes and the resources needed to carry out the tasks. Such an assignment is preferable when the load on the system is constant, and it provides good result for homogeneous environments. Due to growth in the technology usage, the traffic is not constant and is increasing at galloping speed, and static method of allocating load to the nodes is inefficient [8]. There is a need for appropriate technique to distribute the traffic among the nodes in the system. Therefore, to meet the current needs of the system, dynamic load balancing is suitable. In dynamic load balancing, the present conditions of the workload on the computational nodes are considered before executing the load. Some of the load balancing methods are round robin, min–max, game theory-based [4], throttled algorithm, etc. Nature-inspired metaheuristic techniques have also been proposed [9]. These techniques have the characteristics that they are decentralized, self-organizing, agent-based, flexible, robust, scalable, and adaptive to changes in the network. Due to ease of implementation, swarm intelligence-based techniques have been extensively employed [10].

With dynamic load balancing algorithms, there are several research questions that need to be addressed such as how frequently the load balancing should be invoked, which host initiates the decision of load balancing, how to collect the load information of the nodes, and migration of load between the hosts. Various researchers have proposed many different solutions for load imbalance. With advancement in the technology, load balancing has become more significant in fog and edge computing. The goal of load balancing is to improve the overall performance of the system by reducing the load imbalance. The various performance metrics measured are the response time, makespan, resource utilization, fault tolerance, throughput, and migration time. This paper focuses on how the artificial bee optimization algorithm and the metaheuristic-based solution can be applied for balancing the load in distributed systems.

3 Artificial Bee Colony Optimization

Artificial bee colony (ABC) optimization algorithm was originally published by Karaboga in 2005 for numerical optimization problems. ABC optimization algorithm is a nature-inspired, swarm intelligence, metaheuristic algorithm that is based on the foraging behavior of bees [11–13]. According to Ullah et al. [14], the ABC algorithm is adaptive to the heterogeneous environment and also to the varying nature of load [15]. The algorithm has few advantages over other swarm intelligence-based algorithms. The advantages are that the algorithm uses very few control parameters, and it is simple and robust, can be easily hybridized with other algorithms of optimizations, and has fast convergence rate [16]. The algorithm also has better exploration capability as compared to other swarm intelligence algorithms [17]. Some of the application areas of ABC algorithm are cluster analysis, software testing, cluster problem optimization, structural optimization, multilevel thresholding, MR brain image classification, advisory system, numerical assignment problem, bioinformatics, face pose estimation, parameter estimation in software reliability models, wireless sensors [18], big data analytics, edge and fog computing [12, 19].

3.1 Phases of Artificial Bee Colony Optimization Algorithm

Artificial bee colony optimization algorithm has three types of bees: employed bees, onlooker bees, and scout bees.

- Employed bees: The task of the employed bees is to arbitrarily search for food source which represents the potential solutions for the problem. These bees share the information of food source by dancing in the bee hive's dance area. This shared information specifies the quality of solution. The number of employed bees and the number of food sources for the bee hive are equal.
- Onlooker bees: These are the bees waiting in the dance area to assess several dances before choosing a position of food source. The selection is based on the probability proportional to the quality of that food source.
- Scout bees: These bees randomly look for possible new food source [13, 14, 20]

Figure 2 represents the flow diagram of basic ABC optimization algorithm.

4 Recent Developments of ABC Optimization for Load Balancing

With the increase in data and variations in the data rates, there is a need to manage the varying traffic across the system. Load balancing is a technique to balance the load so that all the resources are utilized efficiently. There are several solutions proposed for load balancing. This section introduces few of the recent solutions proposed using ABC optimization algorithm.

The authors in Kruekaew et al. [6] have simulated ABC optimization algorithm for cloud computing by applying a heuristic method which uses certain rules or random methods to find an optimal solution. In the proposed approach, the heuristics is based on the priority for the process selection in the system. The priority criteria used are first come first serve, shortest job first, and largest job first. The system is tested under homogeneous and heterogeneous environments and has compared with other optimization algorithms like ant colony optimization and particle swarm

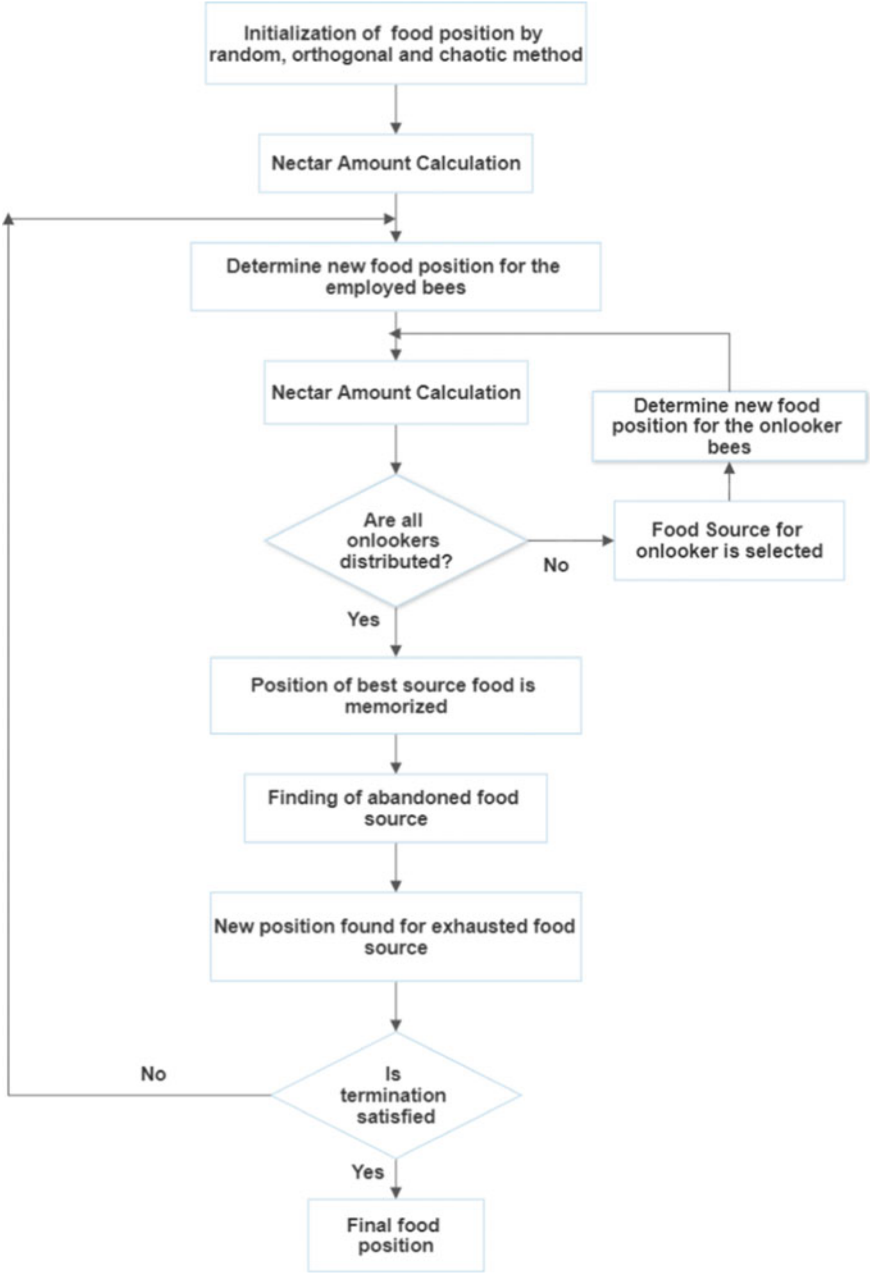
Initialization of food position by random, orthogonal and chaotic method
Nectar Amount Calculation
Determine new food position for the employed bees
Nectar Amount Calculation
Determine new food position for the onlooker bees
Are all onlookers distributed?
No
Food Source for onlooker is selected
Yes
Position of best source food is memorized
Finding of abandoned food source
New position found for exhausted food source
No
Is termination satisfied
Yes
Final food position

Fig. 2 Flow diagram of ABC optimization algorithm

optimization. In the proposed solution, the performance metric measured is makespan and algorithm with largest job first outperforms the other priority criteria.

According to the solution proposed in Hashem et al. [21], the task allocation is carried out by determining the processing time variation of each virtual machine (VM) with respect to the average processing time of all the virtual machines. The proposed solution determines the utilization of a virtual machine by setting a predefined threshold. The system is modeled to be non-pre-emptive. To solve load imbalance, some of the parameters determined are

- The processing time of host and each VM
- Average processing time of all hosts
- Average processing time of all VMs
- Load standard deviation.

The proposed algorithm is compared with round robin and throttled algorithms. The performance metric measured is the response time and the task execution time, and honey bee-based solution outperforms the other algorithms.

The solution proposed in Bhavya et al. [22] is that the authors assign the tasks dynamically according to the change in the user's demands. The algorithm works by grouping the virtual servers, and each virtual server maintains a queue of processes. After processing the request, the profit is determined. With high value of profit, the server stays and low value of profit causes a return to foraging. In this method, there is an overhead in computing the profit which affects the throughput. The load balancing is initiated by assigning the load to a virtual machine (VM) which is underloaded and with maximum throughput. If the load remains below the preset threshold, the load is assigned otherwise a VM with next highest value of throughput is selected. The performance metric measured is throughput.

The authors Korat and Gohe [23] proposed a solution based on multi-objective optimization Pareto dominance, and the weighted sum method is used for selecting the optimal virtual machine (VM) for balancing the load and for priority assignment to the task. All the virtual machines are grouped according to their load. The tasks considered are the pre-emptive tasks, and priority is used to migrate the tasks from one VM to another. To calculate the priority of the task, the parameters used are latency time of task, user type of task, expected priority of the task, and the length of the task. The performance metrics measured are the makespan and the throughput.

In the proposed study, Mallikarjuna et al. [24], the authors initially set various parameters for the virtual machines. The parameters set is million instructions per second (MIPS) which is a measure of the speed of CPU, storage space for each VM, RAM space, and the bandwidth. Each task is identified by the parameters task id and length of the task. The capacity of any virtual machine is calculated as in Eq. 1.

$$\text{Capacity} = \text{MIPS} \times \text{Number of CPUs} + \text{Bandwidth} \tag{1}$$

To balance the load, an iteration method is adopted to select the most appropriate virtual machine, and a newton gravity function is used. The performance metric measured is the resource utilization.

Table 1 Performance metrics measured

Publication	Performance metrics				
	Response time	Makespan	Resource utilization	Throughput	Task execution time
[22]				√	
[21]	√				√
[6]		√			
[23]		√		√	
[24]			√		

Table 1 summarizes the performance metrics measured by applying ABC optimization algorithm to balance load.

5 Conclusion

With the advancement in technology such as fog, cloud, and edge computing, load balancing or the offloading plays a significant role. There are a number of solutions to reduce the load imbalance in the network, however, there is still a requirement for the load balancing. Many researchers have proposed swarm intelligence-based solutions for load balancing. This paper is an attempt to show the recent developments of artificial bee colony optimization algorithm to solve load imbalance in distributed system. Among various performance metrics measured, convergence rate of the algorithm is one of them which needs to be addressed.

References

1. S. Handur Vidya, R.M. Prakash, Response time analysis of dynamic load balancing algorithms in cloud computing. in Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), vol. 2020 (2020), pp. 371–375
2. S. Handur Vidya, S. Deshpande, P.R. Marakumbi, Particle swarm optimization for load balancing in distributed computing systems–a survey. Turk. J. Comput. Math. Educ. (TURCOMAT) **12**(1S), 257–265 (2021)
3. V.S. Handur, S. Belkar, S. Deshpande, P.R. Marakumbi, Study of load balancing algorithms for cloud computing. in *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)* (2018), pp. 173–176
4. S. Penmatsa, A.T. Chronopoulos, Game-theoretic static load balancing for distributed systems. J. Parallel Distrib. Comput. **71**(4), 537–555 (2011)
5. S.B. Kshama, K.R. Shobha, A novel load balancing algorithm based on the capacity of the virtual machines. ICACDS CCIS **905**, 185–195 (2018)
6. B. Kruekaew, W. Kimpan, Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing. Int. J. Comput. Intell. Syst. **13**(1), 496–510 (2020)

7. A. Hanamakkanavar, V.S. Handur, Load balancing in distributed systems—a survey. Int. J. Emerg. Technol. Comput. Sci. Electron. (IJETCSE) **14**(2) (2015). ISSN: 0976-1353
8. S.P. Belkar, V. Handur, Comparative study of static load balancing algorithms in distributed system using cloudsim. Int. J. Adv. Res. Basic Eng. Sci. Technol. (IJARBEST) **5**, 26–30, (2017)
9. S.T. Milan, L. Rajabion, H. Ranjbar, N.J. Navimipou, Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments. Comput. Oper. Res. 159–187 (2019)
10. C. Bansal, A. Gopal, A.K. Nagar, Analysing convergence, consistency, and trajectory of artificial bee colony algorithm. IEEE Access **6**, 73593–73602(2018). https://doi.org/10.1109/ACCESS
11. Sudha Senthil Kumar, K. Brinda, R. Rathi, Angulakshmi, Jothi, Yash Vardhan Thirani: Honey-bee foraging algorithm for load balancing in cloud computing optimization. Int. J. Eng. Sci. Comput. 7(12) (2017)
12. S.S. Ilango, S. Vimal, M. Kaliappan, Optimization using artificial bee colony based clustering approach for big data. Clust. Comput. **22**, 12169–12177 (2019)
13. D. Karaboga, An idea based on honey bee swarm for numerical optimization. *Technical Report TR06* (Erciyes University, Engineering Faculty, Computer Engineering Department, 2005)
14. A. Ullah, N.M. Nawi, J. Uddin, S. Baseer, A.H. Rashed, Artificial bee colony algorithm used for load balancing in cloud computing: review. IAES Int. J. Artif. Intell. (IJ-AI) **8**(2), 156–167 (2019)
15. A.L. Bolaji, A.T. Khader, M. Azmi Al-Betar, M.A. Awadallah, Artificial bee colony its variants and applications: a survey. J. Theor. Appl. Inf. Technol. **47**(2) (2013)
16. S. Anam, Multimodal optimization by using hybrid of artificial bee colony algorithm and BFGS algorithm. IOP Conf. Ser.: J. Phys.: Conf. Ser. **893** (2017)
17. D. Yazdani, M.R. Meybodi, A novel artificial bee colony algorithm for global optimization. in *International Conference on Computer and Knowledge Engineering* (2014)
18. S. Kaswan, S. Choudhary, K. Sharma, Applications of artificial bee colony optimization technique: survey. in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (2015), pp. 1660–1664
19. S.A. Zakaryia, S.A. Ahmed, M.K. Hussein, Evolutionary offloading in an edge environment. Egypt. Inf. J. (in press)
20. L. Shen, J. Li, Y. Wu, Z. Tang, Y. Wang, Optimization of artificial bee colony algorithm based load balancing in smart grid cloud. in IEEE Innovative Smart Grid Technologies—Asia (ISGT Asia), vol. 2019, (2019), pp. 1131–1134
21. W. Hashem, H. Nashaat, R. Rizk: Honey bee based load balancing in cloud computing. KSII Trans. Internet Inf. Syst. **11**(12) (2017)
22. V.V. Bhavya, K.P. Rejina, A.S. Mahesh: An intensification of honey bee foraging load balancing in cloud computing. Int. J. Pure Appl. Math. **114**(11), 127–136 (2017)
23. C. Korat, P. Gohe, A novel honey bee inspired algorithm for dynamic load balancing in cloud environment. Int. J. Adv. Res. Electr. Electron. Instrum. Eng. 4(8), (2015)
24. B. Mallikarjuna, P. Venkata Krishna, A nature inspired bee colony optimization model for improving load balancing in cloud computing. Int. J. Innov. Technol. Exploring Eng. (IJITEE) 8(2S2) (2018). ISSN: 2278-3075