International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.103 ISSN: 2581-4621

Metaheuristic based Task Scheduling for Load Balancing in the Cloud Computing Environment

Vijayalakshmi R[1], Sathya M[2]
[1]*Associate Professor, Master of Computer Application, Cuddalore, Tamilnadu*
[2]*Assistant Professor, Computer Science & Engineering, Pondicherry University, Puducherry*

ABSTRACT

Cloud computing has emerged as a leading technology that handles allocation of tasks in a dynamic manner. Task Scheduling highly contributes to load balancing, and scheduling tasks much adheres to the Service Level Agreement (SLA) requirements, which is a document offered by cloud service providers to the users. Load balancing serves as one of the key concepts of cloud computing that avoids the hectic situation where certain nodes are overloaded while the others are having a very smaller number of tasks and certain not at all loaded. The performance of QoS metrics such as makespan, throughput, performance, response time etc. are having some amount of impact to the degree of imbalance. Cat Swarm Optimization (CSO) and Spider Monkey Optimization (SMO) metaheuristic algorithms are integrated and CSO-SMO-LB has been proposed in Cloud Computing inorder to balance the scheduling of tasks to the available resources. This proposed metaheuristic algorithm aims to optimize resources and improve Load Balancing in terms of the Quality of Service (QoS) task parameters, the priority of Virtual Machine (VM), makespan and resource utilization. The performance of metaheuristic-based algorithms, are analysed by performing simulation and better results were produced.

Keywords – Cloud Computing, Task Scheduling, Load Balancing, Metaheuristic algorithm, Makespan, Resource Utilization

1.Introduction

Cloud computing resources provides high-speed internet services to users through an application [1]. Cloud provides hosting and [2] storage services on the Internet, with the evolution of technologies which aims at achieving optimal resource utilization and high performance [3,4]. Cloud computing consists of many resources scattered over the internet around the world and available for users through a pay-as-use model. The load over the resources increases as the number of cloud users increases day by day. The tasks are to be allocated in an efficient manner to ensure that no machine is idle or overloaded [5, 6]. Nowadays Cloud computing is widely applied in all fields and hence load balancing has become a major issue which evenly maintains tasks among different VM's by providing the requested resources to enhance the performance of the system [7-9].

In the Cloud Computing environment Load Balancing is considered as an NP-complete problem [10, 11]. To solve this problem, many algorithms have been proposed which are divided into static, dynamic, and optimization-based algorithms [12, 13]. The static algorithms are suitable for small workload and will not handle the load at runtime. The cloud computing environment is dynamic, hence in need of dynamic algorithms for efficient task scheduling and to balance the load among the Virtual Machines (VMs) [14, 15].

In this paper, a load balancing approach has been proposed, by integrating the cat swarm optimization algorithm to reduce the energy utilisation with the foraging behavior of spider monkeys using the spider monkey algorithm to optimize the load balancing problem by distributing the user task among the VMs to achieve minimum response time and makespan.

1.1 Cat Swarm Optimization

CSO algorithm is a continuous and single-objective algorithm composed of two modes, namely, tracing and seeking modes. Each cat represents a solution set, which has its own position, a fitness value, and a flag. The search space contains positions in N dimensions with each dimension having its own velocity. The flag value indicates the mode of the cat (either seeking or tracing mode) whereas

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.103 ISSN: 2581-4621

the fitness value tells the wellness of the solution set (cat). The number of cats to be engaged in the iteration is to be finalised and the best cat in each iteration is saved into memory, and the final iteration will represent the final solution [17, 18].

1.1.1. Seeking Mode

Seeking mode imitates the resting behavior of cats. Table 1.1 specifies the fundamental parameters involved in the seeking mode. These values are all tuned and defined by the user through a trial-and-error method.

Parameters	Roles
Seeking Memory Pool (SMP)	it defines number of candidate positions in which one of them is going to be chosen by the cat to go
Seeking Range of the selected Dimension (SRD)	SRD is the mutative ratio for the selected dimensions
Counts of Dimension to Change (CDC)	Defines the dimensions to be modified which is in the interval of [0, 1]
Self-Position Considering (SPC)	specifies whether the current position of a cat will be selected as a candidate position for the next iteration or not

Table 1.1 Fundamental Parameters in Seeking Mode

In seeking mode, as the first step, each cat seeks memory locations and duplicates its own position. Secondly, calculates the count of dimensions to change from the duplicated positions.

$$Pc = \frac{|FCi - FCj|}{|FC_{max} - FC_{min}|} \quad 0 < i < j$$

Where, Pc = probability of latest cat; FCi = fitness calculation value of each cat; FCmax = maximum assessment of fitness calculation FCmin = minimum significance of fitness calculation

1.1.2. Tracing Mode

Tracing mode copies the tracing behavior of cats. For the first iteration, random velocity values are given to all dimensions of a cat's position. However, for later steps, velocity values need to be updated.

Moving cats in this mode:

(1) Update velocities ($V_{k,d}$) for all dimensions according to equation (3).

(2) If a velocity value outranged the maximum value, then it is equal to the maximum velocity.

(3) Update position of Cat_k according to the following equation:

$$V_{k,d} = V_{k,d} + r_1 c_1 (X_{best,d} - X_{k,d})$$

Energy consumption is the principal parameter of load balancing process. The Power consumption of the framework is determined by the absolute Euclidean distance (ED) of all the dynamic PM simultaneously. The smaller ED is considered as the better load balanced system. The PM is turned off, when no assignment is run in relating PM. The Power efficient factor (E) of each active node is calculated based on equation (2).

$$E = \sqrt{\sum_{i=1}^{d} (U_i - Ubesti)^2}$$

$$C = \frac{1}{V} \sum_{i=1}^{v} \left(\frac{Number\ of\ migrations\ in\ VMs}{Total\ number\ of\ VMs} \right)$$

$$U = \frac{1}{PM * VM} \left[\sum_{i=1}^{PM} \sum_{j=1}^{VM} \frac{1}{2} \left(\frac{CPU\ utilized_{ij}}{CPU_{ij}} + \frac{Memory\ utilized_{ij}}{Memoryij} \right) \right]$$

The main goal of proposed technique is to obtain finer load balancing on cloud. Basically, cloud computing contains u number of PMs and each PM consist of v number of VMs. The purpose of the

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.103 ISSN: 2581-4621

proposed algorithm is to reduce Power (P), migration cost (C) and Memory Utilization (M) while the load is balanced.

The objective function is designed which is given in equation (1).

Fitness Function = Min(E + C + M) ---- (1)

Where E – Energy Consumed, C - Migration cost and M – Memory utilization

1.2. Spider Monkey Optimization SMO algorithm [16] is a swarm intelligence-inspired algorithm based on the foraging behavior of spider monkeys, which are social animals that lives in groups and follows a particular living pattern in communication and foraging for food. A female leader always leads a group of spider monkeys in foraging the food which should be enough for all group members. In case of insufficient food, the leader splits the group into smaller sub-groups which in turn foraging in different dimensions and regions to increase the opportunity of finding food sources. The group leader is called the global leader while the subgroup leaders are called local leaders.

1.2.1 SMO algorithm Optimization process

The process of SMO contains six phases, as shown in Table1.2

Local leader phase	while a sub-group is foraging, members changing their positions based on information from the local leader and from group members
Global leader phase	based on the information from the Global leader and the local leader, the subgroup members of all sub-groups are changing their positions
Global leader learning phase	Applies greedy selection to find the nearest spider monkey
Local leader learning phase	Finds the nearest foraging, within the of the sub-group domain
Local leader decision phase	to avoid stagnation, then all sub-group members will update their positions based on information from the global leader and the local leader
Global leader decision phase	to avoid stagnation, all sub-group members will update their positions based on information from the global leader and the local leader

Table 1.2 Phases of SMO

1.3 CSO-SMO-LB Algorithm

The computational complexity of the proposed approach can be analyzed by examining the updations of the positions, grouping, and assigning operations. The grouping of VMs, user tasks creates additional complexity. The creation of subgroups and groups plays a vital role in balancing the load.

Algorithm:

CSO-SMO-LB algorithm for load balancing

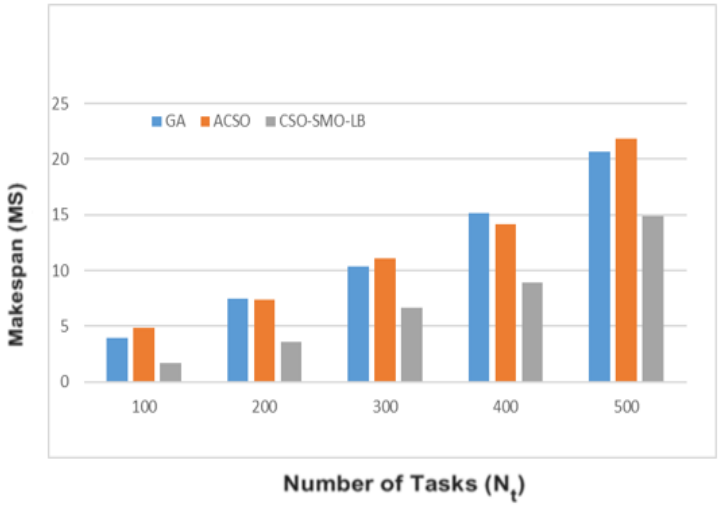
1: #PREPROCESSING PHASE
2: Virtual machine grouping
3: Expand the list of all directions to VMs
4: # THE LOAD BALANCING PHASE
5: Call Load Balancing Decision
6: while (the termination is not fit)
calculate the fitness function for all cats
7: Grouping of User Task
8: Create subgroups and identify local leader
9: Combine all subgroups and identify global leader
10: Identify better outcomes

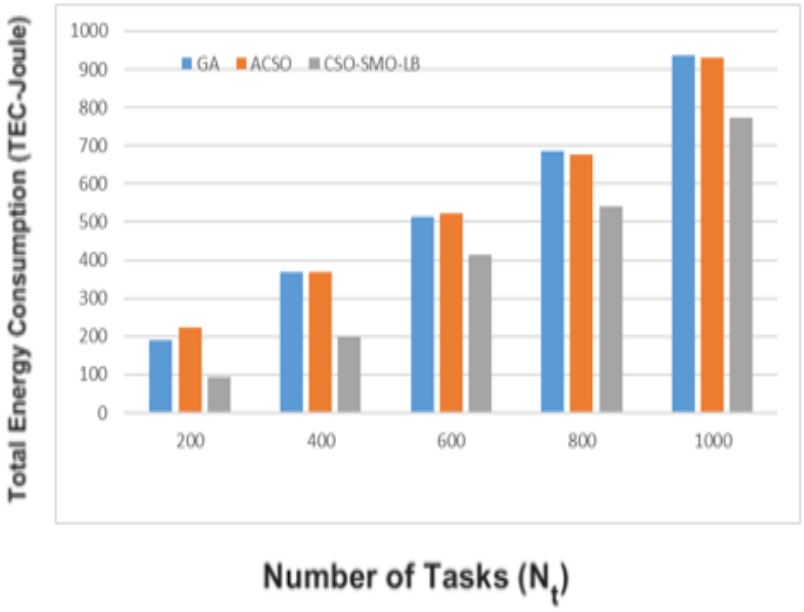
Experimental results

The main aim of this study is to reduce the energy consumption in the VM's in PM. To avoid the energy consumption and Makespan, load balancing process is used in VM. To provide evidence of

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.103 ISSN: 2581-4621

proposed algorithm comparison is done with different method namely, CSO-SMO-LB based load balancing and GA based load balancing. The performance of the CSO-SMO-LB algorithm has been evaluated through simulation done using CloudSim. The version of the system is Intel Core i7 8th Generation processor, 1.8 GHz CPU, and 16 GB RAM running on Microsoft Windows 10 platform. Different experiments were performed with different autogenerated datasets and different parameter values.





CONCLUSION

In this paper, developed a load balancing method CSO-SMO-LB for cloud computing environments inspired by the foraging behavior of Cats and Spider Monkeys. The makespan value is minimized in turn by minimizing the energy utilization. To make the Spider Monkey Optimization algorithm applicable for finding the optimal load balance of tasks via virtual machines, a mathematical model was also developed for job mapping using the CSO-SMO-LB algorithm over the cloud environment. The developed method not only handling the issue of load balancing but also takes into consideration the capability and accessibility of the resource through the proposed grouping strategies of tasks and virtual machines. The SMO-LB was tested using the CloudSim simulator with various testing scenarios and evaluated in comparison with two other methods, Genetic Algorithm(GA) and Adaptive Cat Swarm Optimization(ACSO), and the proposed method outperforms.

DOI:10.46647/ijetms.2022.v06i05.103 ISSN: 2581-4621

REFERENCES

1. V. Kunwar, N. Agarwal, A. Rana, J.P. Pandey, Load balancing in cloud—A systematic review, Big Data Analyt. (2017) 583–593.
2. Z. Goudarzi, A. Faraahi, Effective load balancing in cloud computing, Int. J. Intell. Inf. Syst. 3 (6) (2017) 1.
3. S. Chhabra, A.K. Singh, Dynamic hierarchical load balancing model for cloud data center networks, Electron. Lett. 55 (2) (2019) 94–96.
4. K. Balaji, P. Sai kiran, 2017. Efficient resource allocation algorithm with optimal throughput in cloud computing. J. Adv. Res. Dyn. Control Syst. Volume 9, 2017, Pages 1902-1910.
5. Hung P., Alam M., Nguyen H., Quan T., and Huh E., "A Dynamic Scheduling Method for Collaborated Cloud with Thick Clients," The International Arab Journal of Information Technology, vol. 16, no. 4, pp. 633-643, 2019.
6. Shukla A., Kumar S., and Singh H., "Fault Tolerance Based Load Balancing Approach for Web Resources in Cloud Environment," The International Arab Journal of Information Technology, vol. 17, no. 2, pp. 225-232, 2020
7. K. Radha, B. Thirumala Rao, S.M. Babu, K. Thirupathi Rao, V. Krishna Reddy, P. Saikiran. 2014. Allocation of resources and scheduling in cloud computing with cloud migration. Int. J. Appl. Eng. Res. Volume 9, Issue 19, 2014, Pages 5827-5837.
8. S.T. Maguluri, R. Srikant, L. Ying. (2012). Stochastic models of load balancing and scheduling in cloud computing clusters. In 2012 INFOCOM (pp. 702–710). IEEE.
9. C. Thiam, G.D. Costa, J.M. Pierson. (2013). Cooperative scheduling anti-load balancing algorithm for cloud: CSAAC. In IEEE international conference on cloud computing technology and science (pp.433–438)
10. Singh A., Sahu S., Tiwari M., and Katare R., "Scheduling Algorithm with Load Balancing in Cloud Computing," International Journal of Scientific Engineering and Research, vol. 2, no. 1, pp. 38-43, 2014.
11. Gopinath P. and Vasudevan S., "An In-Depth Analysis and Study of Load Balancing Techniques in The Cloud Computing Environment," Procedia Computer Science, vol. 50, pp. 427-432, 2015
12. Balla H., Sheng C., and Weipeng J., "ReliabilityAware: Task Scheduling in Cloud Computing Using Multi-Agent Reinforcement Learning Algorithm and Neural Fitted Q," International Arab Journal of Information Technology, vol. 18, no. 1, pp. 36-47, 2021.
13. Shafi U., Shah M., Wahid A., Abbasi, K., Javaid Q., Asghar M., and Haider M., "A Novel Amended Dynamic Round Robin Scheduling Algorithm for Timeshared Systems," The International Arab Journal of Information Technology, vol. 17, no. 1, pp. 90-98, 2020.
14. Abunaser A. and Alshattnawi S., "Mobile Cloud Computing and other Mobile Technologies: Survey," Journal of Mobile Multimedia, vol. 8, no. 4, pp. 241-252, 2013.
15. Rahman M., Hassan R., Ranjan R., and Buyya R., "Adaptive Workflow Scheduling for Dynamic Grid and Cloud Computing Environment," Concurrency and Computation: Practice and Experience, vol. 25, no. 13, pp. 1816-1842, 2013.
16. Bansal J., Sharma H., Jadon S., and Clerc M., "Spider Monkey Optimization Algorithm for Numerical Optimization," Memetic Computing, vol. 6, no. 1, pp. 31-47, 2014.
17. S. C. Chu and P. W. Tsai, "Computational intelligence based on the behavior of cats," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 1, pp. 163–173, 2007.
18. S. C. Chu, P. W. Tsai, and J. S. Pan, "Cat swarm optimization," in *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, pp. 854–858, Springer, Guilin, China, August 2006.