ISSN:2229-6107

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY

E-mail :
editor.ijpast@gmail.com
editor@ijpast.in

www.ijpast.in

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST

ISSN 2229-6107 www.ijpast.in

Vol 11,Issuse 4.Nov 2021

Cloud-Based Load Balancing with an Altered ABC

Dr. V Venkata Ramana[1,] Dr. V Lokeswara Reddy[2] Dr K Sreenivasa Rao[3], Dr.M Ramanjeneya Reddy[4]

Abstract

Through the Internet, users of cloud computing may get on-demand, scalable virtual resources at a low per-use cost. Today, many organizations and individuals use cloud services to get access to the Internet's many data stores. To handle the massive demands and clients they provide, cloud service companies need to carefully manage their data centers. As a result, providers need load balancing to ensure optimal use of available resources. Researchers have created a plethora of meta-heuristic algorithms to address the NP-hard optimization problem of cloud-based load balancing. The purpose of this research is to offer a mutative bacterial foraging algorithm that may shorten make-span time while simultaneously enhancing productivity. The approach not only shortens the life cycle, but also improves health and vitality. By accounting for work variability, the suggested technique outperforms existing methods in the literature when applied to a heterogeneous cloud environment.

Introduction

Every industry today, from social media to IT to other applications, is shifting to the cloud. One efficient way to pool internet resources in response to user demand is via the usage of cloud computing. Making the most of available resources is crucial for optimizing system performance. Many businesses see this need and respond by providing sophisticated online services. However, users choose a single cloud provider and then submit a processing request to that service. In cloud computing, the algorithm must transmit the execution request to any virtual machine. While there are many virtual computers capable of running such a system, the question of how, where, and on which machine a given request should be made remains.

[1,2,3, 4]Professor,

Department of CSE, K.S.R.M College of Engineering(A), Kadapa

ISSN 2229-6107 www.ijpast.in

Vol 11,Issuse 4.Nov 2021

The whole system performance may suffer if, for instance, all requests have been directed to only one or two computers while the other machines are idle. Since improving cloud computing's overall performance is a major problem for both businesses and academics, the introduction of the term "load balancing" highlights this issue.

Load balancing is the process of dividing incoming requests across many virtual nodes in such a manner that each request has the least possible delay. Several cutting-edge algorithms for balancing loads, both static and dynamic, have been built and proposed. Static algorithms are straightforward and effective in a consistent setting, but they don't allow for parameter tweaks mid-execution [1]. Dynamic algorithms are superior because their execution may be tailored to the user's specific changes in parameters. These algorithms are able to function under a diverse set of conditions. However, it is more difficult for algorithms of any kind to accept changes in characteristics. Research into load balancing algorithms may be facilitated by their limitations, which are both static and dynamic. This problem must be addressed before cloud computing can be used effectively.

Literature Survey

There are many challenges and issue in load balancing algorithm. Such as tolerance of high delays [2] because of distance between nodes, network bandwidth etc. It is location-based issue. Sometimes high storage is required in case of full replication algorithm. The replication of data at various nodes increases the cost [3]. So, algorithm is required to up the level as to manage cost. In terms of operation complexity, load balancing algorithm operation complexity must be minimum. The negative performance occurs with high complexity. Similarly, delay creates problem in large data communication [4]. Another challenge is POF (point of failure) [5]. In central node-based topology would be fail be controller node fails. In this distributed system algorithm provide a new way to solve this POF issue. In 2008, a famous algorithm which was mostly used in various applications was implemented by Soto mayor et al [6]. An improved version in 2010 with name central load balancing decision model (CLBDM) [7] has been suggested by Radojevic et al. They worked on application layer session switching. In CLBDB, connection time has been calculated, they have used thresh hold value and if time exceeded the thresh value then connection switch to next node.

In 2012, ant's behaviors have been taken in consider in [8, 9] to solve load balancing problem. As we know, ant's having a good habit to collect information in fast way, same used by instant et al. and hang et al. As there is no central node, the problem of single point of failure is avoided. Another

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST
ISSN 2229-6107 www.ijpast.in
Vol 11,Issuse 4.Nov 2021

advantage of this research is that it searches under loaded nodes fast. To overcome the load balancing issue, map reduce is also used. But it takes high processing time [12]. They have also not considered reduction in task. The reliable connection method proposed by in VM mapping method [13]. But they have not measure node capabilities and network load. In 2017, Singh et al. presented an improved version of algorithm suggested by Kalra and Singh in 2015. They remove the limitation of meta-heuristic techniques by extending meta-heuristic—based task scheduling. Hashed et al. [19] used foraging strategy of honey bee and proposed an algorithm for cloud load balancing. They actually worked on response time and try to improve to improvise cloud services. The suggested algorithm finds overloaded virtual machine with threshold value which equals to average processing time. In 2020, Signal et al. [20] proposed MPSO mutation based PSO as a modified algorithm. This algorithm worked on data canter to utilize the services of servers in cloud. They propagate task at various data centers with division in subtask as individual job. They select appropriate data centers for allocation of subtask as per their availability.

Proposed Work

In this paper, a new modified artificial bee colony (ABC), called Mutation Based ABC (MABC) is described. The proposed algorithm gives emphasis to process of finding under-utilized servers available in the data centers. For using the resources on cloud, the users use Internet for sending their job request. Cloud Service Provider ensures that each submitted resource is allocated to some VM for execution. For this, the submitted task may move from one data centre to other looking for an under-utilized resource. N turns these data centers may divide the submitted task to sub-tasks commonly known as jobs. The data centers search for the under-utilized resources in the data centers for allocating the jobs to them. The proposed algorithm is able to minimize the make span time of the jobs by assigning it to the available under-utilized data centers. Artificial Bee Colony: For each job, fitness function is calculated after updating the positions of bee. The flow of the algorithm is shown in figure 1.

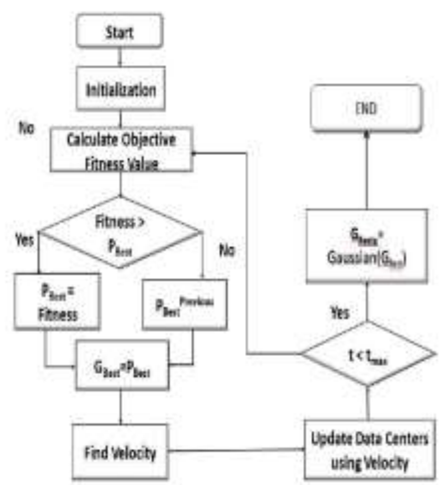

Fig 1. Flow Chart of proposed algorithm As the value of test improves; the entire population of bee gets better. If the mutation is applied to this value of test the system will convert the value of the Best value and hence the overall value of Best can improve. The value of Best will be accepted only if the value improves. A Gaussian mutation is applied

ISSN 2229-6107 www.ijpast.in

Vol 11,Issuse 4.Nov 2021

as the mutation technique on the search space to improve value of Best.

Simulation

Implementation and Parameters settings

To implement the proposed algorithm, CloudSim 3.0.3 simulator is used. It has been configured with eclipse tool. Using Clouds, the researchers can model power aware and energy aware cloud solution. Proposed algorithm minimizes the make span time while improving the fitness function. Make span time is the time duration of executing the total submitted tasks to the environment. In cloud computing, it defines the maximum time for executing cloudlets running on different data centers. Make span time has been considered as the parameter for measuring the performance of proposed algorithm. The proposed algorithm is compared with other nature inspired algorithms like ACO and PSO. The parameters that have been considered for experimental setup are shown in table 1.

Table 1. Experimental Parameters for Clouds

Entity	Type	Parameters Values
User	Number of Cloudlets	10-300
Cloudlets	Length	500-10000
Host	Number of Hosts	2
	RAM	8GB
	Storage	20 GB
	Bandwidth	100
	Numbers of VMs	4
Virtual Machine	RAM	2 GB
	Storage	5 GB
	Operating System	Windows
	Type of policy	Time Sharing
	Numbers of CPUs	1
Data centers	Numbers of Data Centers	5-15

Experimental result

The result obtained by running proposed algorithm are shown in figure 2,3 and 4 for the value of data centers 5, 10 and 15 respectively.

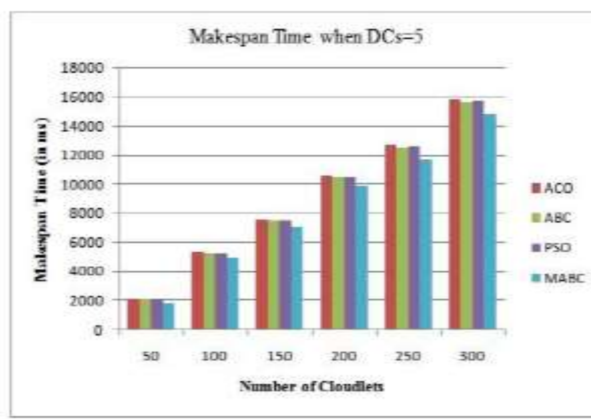

Fig. 2. Make span Time comparison when DC is 5

ISSN 2229-6107 www.ijpast.in

Vol 11,Issuse 4.Nov 2021

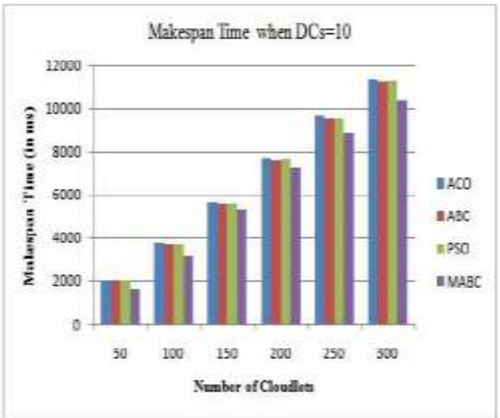

Fig. 3. Make span Time comparison when DC is 10.

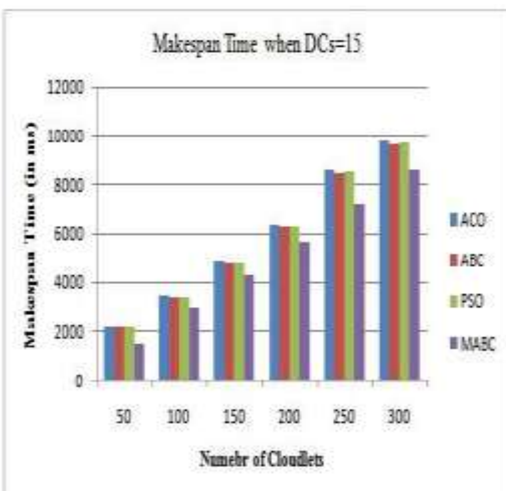

Fig. 4. Make span Time comparison when DC is 15

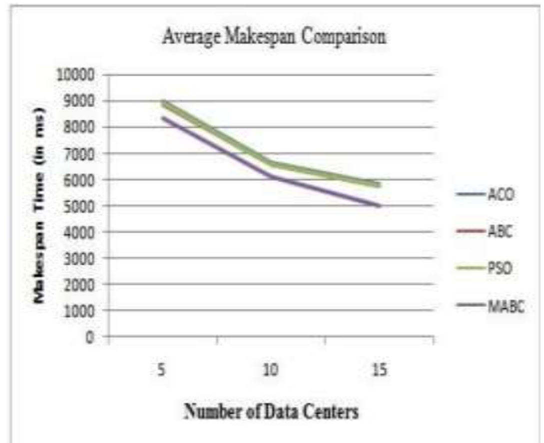

Fig. 5. Make span Time comparison when DC is constant

It has been observed the proposed algorithm gives a better value of make span time than the algorithm present in literature. For different set and values of cloudlets and data centers the result are improves in the range of 5%-15%. The result in figure 5 show the comparison of algorithms when the value of data centers is kept same and the value of cloudlets is varied.

Conclusion and Future Work

Since cloud service providers must manage a massive volume of assignments and customers, load balancing has emerged as a critical challenge in the cloud computing industry. Since the rate at which new jobs occur fluctuates greatly, load balancing is made more difficult. In this article, we will investigate the problem of load balancing in the cloud and make some attempts to suggest a solution. Overall make span is reduced and fitness is enhanced by the suggested load balancing computation MABC. In this method, assets scattered throughout a large network of data centers are monitored and managed with

ISSN 2229-6107 www.ijpast.in

Vol 11,Issuse 4.Nov 2021

precision. The suggested approach yields better results in both the data center and worker count situations. In the future, we may investigate a wide range of characteristics that can be used to estimate load balancing displays.

References

A Study of Distributed Load Balancing Algorithms for Cloud Computing, by Michael Rundles, David Lamb, and Ali Taleb-Bendiab, Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, Australia, April 2010.

In the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), held in Busan, South Korea, in 2010, R. Bunya, R. Raman, and RN. Cahiers presented "Interclub: Utility-oriented federation of cloud computing environments for scaling of application services."

In proc. Grid Computing Environments Workshop, pp. 99-106, 2008, I. Foster, Y. Zhao, I. Riau, and S. Lu evaluated cloud computing versus grid computing from every angle.

In "Cooperative load balancing in distributed systems," by D. Gross, A.T. Chronopoulos, and M. Leung, available in

Journal of Parallel and Distributed Computing, Volume 20, Issue 16, Pages 1953–1976, 2008.

"Puerto-peer cloud provisioning: Service discovery and load-balancing," by Raman, R., Zhao, L., Wu, X., Liu, A., Quiroz, and M. Parishes, in Cloud Computing: Principles, Systems, and Applications, pages 195-217, 2010.

Reference: [6] Radojevic, B., and M. Agar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." Presented at the 34th Annual IEEE Convention on MIPRO (2011).

"Virtual Infrastructure Management in Private and Hybrid Clouds," by B. Soto mayor, RS. Montero, IM. Llorente, and I. Foster, published in IEEE Internet Computing, Volume 13, Issue 5, Pages 14–22, 2009.

Based on the work of Nishant, K. P. Sharma, V. Krishna, C. Gupta, KP. Singh, N. Nitin, and R. Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization." March 2012: Presented at the 14th IEEE International Conference on Computer Modeling and Simulation (UKSim), pages 3-8.

Reference: [9] Zhang, Z., and X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation." 240–243 in Proceedings of the 2010 IEEE 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), May 2010.

Osmotic bio-inspired load balancing method in cloud computing. [10] Gamal, M., R. Rizk, H. Mahdi, and B.E. Elnaghi. (2019) IEEE Access 7, 42735-42744