Cloud-Based Load Balancing with an Altered ABC

NIBEDITA SAHOO
ASSISTANT PROFESSOR, Mtech,Ph.D
Department of CSE
Gandhi Institute for Technology,Bhubaneswar.

Abstract

Cloud computing enables users to access Internet-based, elastically scalable virtual resources on a pay-as-you-go basis. Many people and businesses now rely on cloud services to access a wide range of online resources. Cloud service providers must efficiently manage their data centers to cope with the vast numbers of requests and customers they support. Therefore, suppliers need load balancing to guarantee efficient resource usage. Numerous scholars have developed numerous meta-heuristic algorithms to solve the NP-hard optimization issue of load balancing in the cloud. The study presents a mutative bacterial foraging algorithm with the goals of reducing makes span time and increasing efficiency. The method does more than only reduce the make span time; it also boosts fitness. Results from a comparison of the proposed method to other algorithms in the literature show that it achieves better performance in a heterogeneous cloud setting while taking job variability into account.

Introduction

In today's scenario every business-like social media, IT industries and other application are moving on Cloud computing. Cloud computing is one of high-performance mechanism to share resources on internet by user demand. Maximum use of utilization of resources is important and it increases system level performance. For all this reasons, various companies are offering online services with high level systems. However, people choose one cloud service and send Request to process a task. In cloud computing, the request for execution has to be transferred to any virtual machine by algorithm. There is number of virtual machines supporting this kind of system but problem is, how, where and on which machine the request has to be sent. For example, if all requests have been sent to one or two machine and other machines are free then overall system performance will be down. Therefore, load balancing term arrive in cloud computing to enhance overall system performance which is a big challenge for industry and researcher.

To distribute various incoming request to many virtual nodes in such a way that all request will have minimum waiting time can be defined as load balancing. There are number of static and dynamic load balancing algorithm has been implemented and suggested as state-of-the-art methods. The static algorithms are easy and easily work in homogenous environment but cannot support changes in parameters at run time [1]. The dynamic algorithms are more efficient and adjust execution according to the changes in parameters as per request. These algorithms work in heterogeneous environment. However, acceptance of changes in attributes is more complex in all types of algorithm. The limitation of static and dynamic algorithm of load balancing algorithm gives a way to do research on this area. So, to use cloud efficient the issue must be resolve.

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 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.

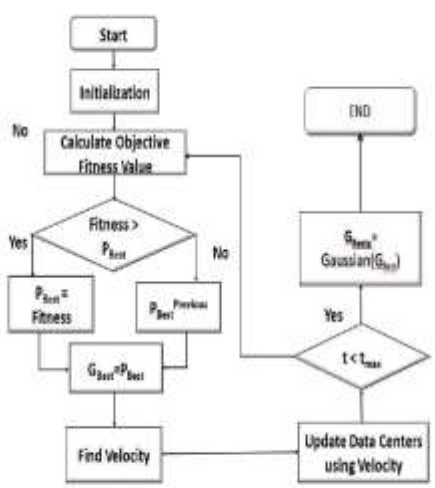

Fig 1. Flow Chart of proposed algorithm As the value of test improves; the entire population of bee gets better. If the mutation is applied to this value of test the system will convert the value of the Best value and hence the overall value of Best can improve. The value of Best will be accepted only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

Copyright @ 2020 Authors

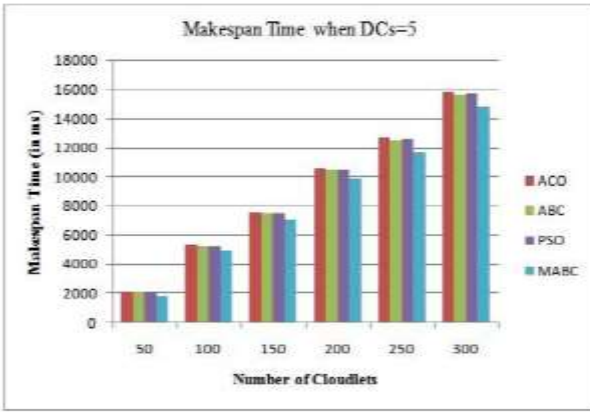

Fig. 2. Make span Time comparison when DC is 5

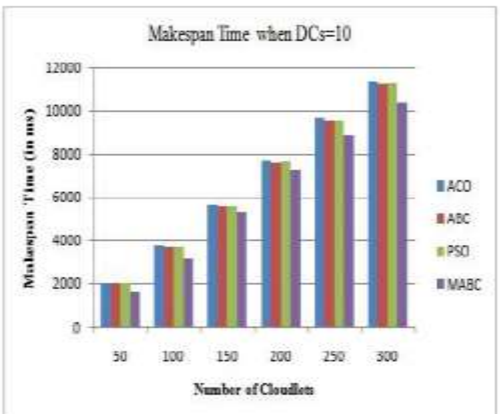

Fig. 3. Make span Time comparison when DC is 10.

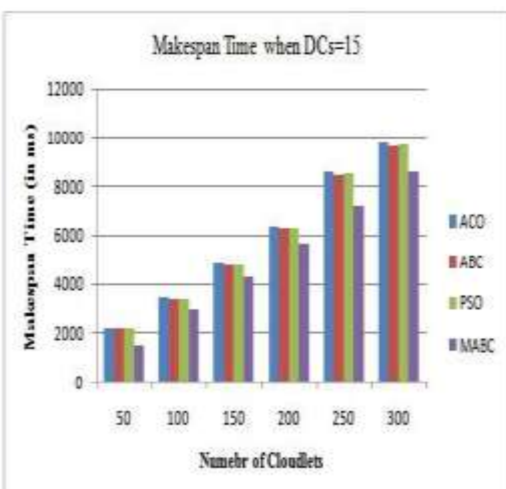

Fig. 4. Make span Time comparison when DC is 15

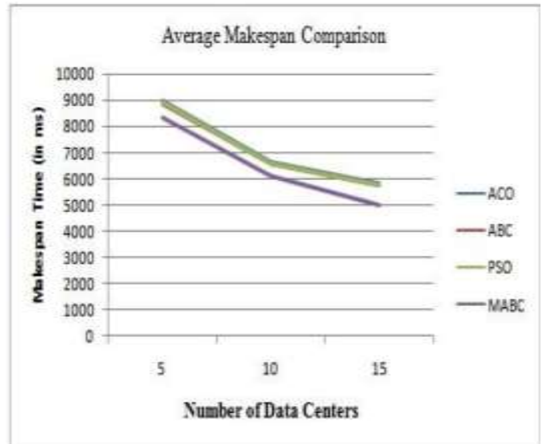

Fig. 5. Make span Time comparison when DC is constant

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 precision. The suggested approach yields better results in both the data center and worker count situations. In the future, we may investigate a wide

Copyright @ 2020 Authors

range of characteristics that can be used to estimate load balancing displays.

References

[1] Rundles, M., D. Lamb and A. Taleb-Bendiab, ―A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, ‖ in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, Australia, April 2010.

[2] Bunya R., R. Raman and RN. Cahiers, ―Interclub: Utility-oriented federation of cloud computing environments for scaling of application services,‖ in proc. 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Bussan, South Korea, 2010.

[3] Foster, I., Y. Zhao, I. Riau and S. Lu, ―Cloud Computing and Grid Computing 360-degree compared,‖ in proc. Grid Computing Environments Workshop, pp: 99-106, 2008.

[4] Gross, D., A.T. Chronopoulos and M. Leung, "Cooperative load balancing in distributed systems," in Concurrency and Computation: Practice and Experience, Vol. 20, No. 16, pp: 1953-1976, 2008.

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

[6] Radojevic, B. and M. Agar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." In proc.34th International Convention on MIPRO, IEEE, 2011.

[7] Soto mayor, B., RS. Montero, IM. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," in IEEE Internet Computing, Vol. 13, No. 5, pp: 14-22, 2009.

[8] 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." In proc. 14th International Conference on Computer Modelling and Simulation (UKSim), IEEE, pp: 3-8, March 2012.

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

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