International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Distributed Load Balancing Architecture for IAAS Cloud Computing

Avi Chauhan[1], Dr. Manish Jha[2]

[1]M. Tech Scholar, CSE Department, IITM, Sonipat, Haryana
[2]Professor, CSE Department, IITM, Sonipat, Haryana

Abstract
Cloud Computing has been evolved with various techniques for satisfying the needs of the users with an aim to reduce the cost and offer the better result. The users' need may include sharing of resources like memories, processors, apps, information, data, applications etc., whereas, performance should be in terms of improved DC processing period & response-time. Apart from above-said, there is also a requirement to manage the stability of the system and flexible to make the amendments in the system. Static data is used to optimize the project schedule. However, the traditional ACO does not take into account the personnel allocation matrix when scheduling projects. The ACO model is not a suitable solution to the scheduling problem. The classic ACO methodology operates in two phases: the first phase uses an event-based scheduler to address the complex planning issue.

Keywords: Cloud computing, PSO, ACO, GA

I. INTRODUCTION

With the widespread use and rapid growth of internet technologies in the present era, Cloud Computing has gained the popularity for its use in the industry and academia. Its infrastructure is easily accessible for business or any other purpose across the globe as per the demand of the user. The word "Cloud Computing" is derived from two unique words, one is "Cloud" which is basically related to network and another is "Computing" which defines itself as calculation. Therefore, Cloud Computing (CC) may be referred as process and compute data using computers. More precisely, CC is an IT deployment model used for sharing the resources including servers, hardware, software, data, analytics, information, intelligence, memory, storage space, web-services, emails, networking, apps, desktop accessibility, printers, audio, video etc. The internet as a service provided by a single or a group of providers. The special features like scalability, virtualization and easy use of CC has made it popular among the many industries for storing and executing their applications [1]. Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure (Azure) are some of the public cloud service providers that can be found in Figure 1.

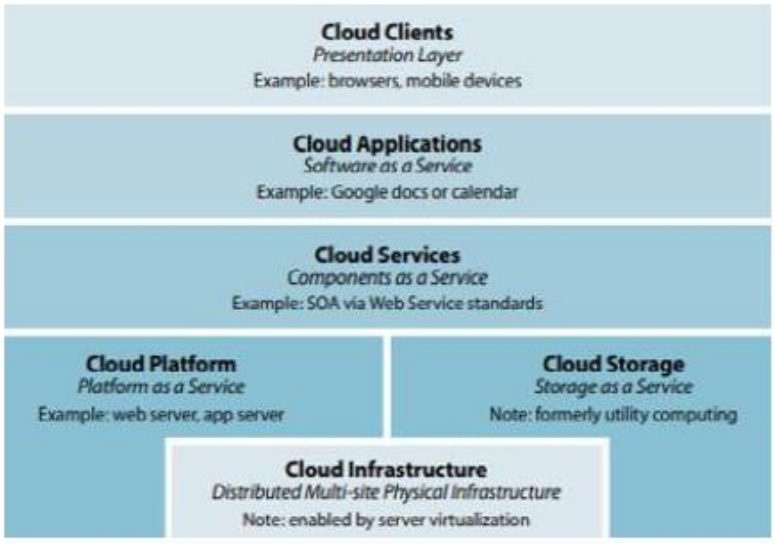

Figure 1: Schematic Diagram of Cloud Computing

The motivation behind this work is to know about the performance factors of load balancing techniques used in Cloud Computing. For reducing the time to get executed the assigned tasks, it is necessary to distribute the workload homogenously to the entire available resources after keeping their processing speed in the mind.

The paramount objective of a load balancing technique is to optimize the response time, DC processing time of the assigned applications and also reduce the required cost. In Cluster Computing, many computers are employed together, usually Personal Computers or UNIX workstations with redundant connectivity and various storage media to create a single highly available system. This type of computing is used fort high availability. Cluster proponents claim that in some situations, clustering can help a company reach 99.99 percent availability. One of the basic concepts of cluster computing is that the cluster appears as a single unit or system to the outside world. [6][7].
When there is a lot of traffic on a website, cluster computing is used to balance the load. A request for an internet page is routed to a "manager" server for election of similar kind of internet servers for responding the request. Cluster computing can also be utilised as a low-cost method of multiprocessing for scientific and other applications that benefit from parallel processing [8].

II. Literature Review

G. Singh Chhabra et al. [1] has explained about the multiple-machine system, where, the chances of anyone of the machine may be remain being idle, on the other hand, other processors may have multiple tasks. When such type of situation arises in the system load then the performance can be enhanced by migrating the tasks from the machines with the overloads to the machines with the less loads. Further, he categorized the Load Balancing algorithms as dynamic and static. DLB algorithms are flexible to changing scenarios and make judgments at run time, whereas, SLB algorithms make decisions about job assignment to processor and delays at compile time.
A. Singh et al. [2] has demonstrated about the LBA which is very simple to implement and passes requests to the registered nodes in a cyclic manner after getting the requests from distinct users. The

proposed algorithm is not able to assign any preference to higher capacity systems during the allotment of the tasks and best suited for the servers of similar kind of specifications.

According to **Shu-Ching et al. [3],** hardware technology and required network bandwidth are fast developing. Therefore, in order to use the network's computing resources to do complex activities that demand large-scale calculation, the elected nodes must be taken care of, and nodes must be sensibly picked according to the parameters of the given job in order to maximize the usefulness of the resources. Further, they also suggested Load Balance Min-Min scheduling techniques for improving system load balancing and execution efficiency.

A. Alnowiser et al. [4] has explained the weighted RR algorithm which executes exactly in the similar way as the conventional RR algorithm but with a twist. In this technique, the greater number of requests will be catered by the node which will have higher specifications. Every node will get assigned with the weights well in advance and also get registered with in the load balancer for fulfilling the required conditions.

V. Mohammadian et al. [5] have stated Fault tolerance is also important in load balancing algorithms, however there is still a need for more research in this area. This gap inspired authors to conduct the current study, which intended to gather and assess existing articles in the subject of fault tolerance load balancing. The available algorithms are being classified as; centralised and distributed, and then these are evaluated on the basis of important qualitative factors.

G. Liu et al. [6] have elucidated the Min-Min Algorithm and notify the time it takes for pending jobs in a queue to be executed and completed. The cloud adminstartor allocates tasks to the processors which are efficient to execute the task within the stipulated period of time. There will a long wait for the process with maximum execution.

Li et al. [7] have explained Max-Min algorithm containing job status table for measuring the real-time load of VMs along with expected time required for accomplishment of the jobs.

O. M. Elzeki et al. [8] have premeditated the Max-Min algorithm which assigns the task or job named T on the resource said to be R, where, bigger jobs in the task pool have been assigned the highest priority in comparison to smaller tasks. Initially, it starts with completion time has to computed for individual task in the resource pool. After that, task is assigned to the resources which will have least execution time for the completion of the task, then prepared time of the available resource is being altered and the new planned task is also eliminated from the meta job or task. This procedure is continuously repeated till the meta-task become idle. The idea behind is to propose this method to get decreased the stand by time of the assigned jobs.

Maheswaran et al. [9] intended to improve the makespan and balancing the load of data centre by allotting the jobs in the optimised way. The vital disadvantage of proposed technique is that it does not pay attention to the machine ready time and also indicates various changes in the load across the virtual machines.

S. C. \Wang et al. [10] have demonstrated a hybrid method by integrating the Opportunistic and Min-Min LBAs. The purpose to propose this new technique is to manage the load by decreasing the execution time and improving the efficiency.

S. Penmatsa et al. [11] have projected a game theoretic approach for providing the solutions to static load balancing problems. Authors have considered the single channel communication in a distributed system which comprises distinct computers of different configurations. A cooperative game has been proposed for solving the load balancing issues.

O. M. Elzeki [12] et al. have developed an algorithm in which jobs are being assigned purely on random basis without considering the availability of the machines but focus on execution of the assigned job in minimum time. The basic motive of the authors to propose this algorithm is to assign a job which has to be completed in the minimum execution time, however, this approach may lead to imbalance across the devices.

D. Grosu et al. [13] have anticipated a game for getting the client-optimal load balancing approach in non-homogenous dispersed systems, and named noncooperative load balancing game. The authors have also reported the non-complexity and optimal allocation of the tasks for timely execution and they have also juxtaposed the proposed approach with existing load balancing approaches

N. A. Mehdi et al. [14] have perceived an algorithm which executes the job in minimum completion time. The request of the user is being forwarded to the available VM by the Data Centre Controller and this approach is chiefly based on speed of the processor along with load bearing capacity of the virtual machines.

Moly et al. [15] have expounded a new Modified spherical Robin formula in which authors have focused on arrival time and burst time. The burst time and number of processors are considered as the input. Authors incline to arrange all procedures in ascending order as per predefined explode time and also select for altered time slice depending upon the number of processes, if processors vary then time slice may also get vary.

III. PROPOSED ALGORITHM

Each iteration of an optimization problem improves a set of possible solutions known as individuals and their abstract representations as chromosomes until the optimum solution is found. The survival of each individual is determined by a selection operation in the Genetic Algorithm (GA). In other words, the overall health of the population is assessed. A representative sample of the current population is selected at random, and those with greater levels of fitness are more likely to be passed down to the next generation, while those with lower levels of fitness are less likely to be passed down to the next generation. Natural selection and mutation result in a new population being formed with each generation.

3.1 Selection Operation

In the previously described natural selection process, the most often employed way of picking people is the roulette selection [20, 21] approach. For example, a selection operator, or proportional selection, sets the likelihood of an independently being picked in ratio to its fitness-function magnitude, as illustrated in Figure 3. The chance xi that an individual xi will be chosen by our operator if the population size N and the fitness of that individual xi are assumed to equal

$$p(x_i) = \frac{f(x_i)}{\Sigma_{j=1}^{N} f(x_j)}$$

Because the size of each chromosomal sector corresponds to an individual's level of fitness, those with higher levels of adaptation are more likely to be selected.

3.2 Genetic Operations.

(a) crossover

Based on the crossover chance, two chromosomes swap genes to generate two new people.
Before crossover: 01000|0100001000
Then: 01000|1111000010 11000|0100001000
(c) Mutate
A mutation transforms a chromosome's coding string to a new gene value. The mutation procedure helps sustain population variety.
Earlier: 01000|1111000010
After 01000|111001010

A fundamental tenet of genetic algorithms is that they are based on the biological principle of natural selection to produce a population. The field of artificial intelligence (AI) known as general artificial intelligence (GA) is on the rise. Originating from Darwinian evolution, the Genetic Algorithms (GAs). Scheduling activities based on fitness-function values for every features of the process is a principle advocated by Darwin's evolutionary theory (Darwin's theory of evolution). Listed below are the GA's core ideas. [1] [2].
a) The earliest settlers
The GA's initial population is the total number of people that participated in the search for the best answer. In the general population, the term "individual solutions" is used. An individual's chromosome is used as a model for genetic testing processes. From the initial population, a limited number of people are selected for each generation. Some precise criteria are used to choose the mating chromosomes.
2) The Function of Fitness
A person's productivity is directly related to their level of fitness. It's a way to gauge how much of an advantage a person has over the rest of the population. It is a measure of an individual's overall performance in the population. As a result, each individual is either alive or dead depending on his or her ability to do a certain task. As a result, the GA's fitness function is the driving force behind it.
3) Selection
Based on Darwin's rule of survival, the selection process is utilized to pick an intermediate solution for the following generation. The GA uses this action as a compass to direct its course. Selection tactics include the roulette wheel, the Boltzmann strategy, tournament selection, and ranking the chromosomes depending on their quality.

IV. RESULT ANALYSIS

The result of cloud scheduling is concluded in terms of different weight task , execution time , estimated cost etc.

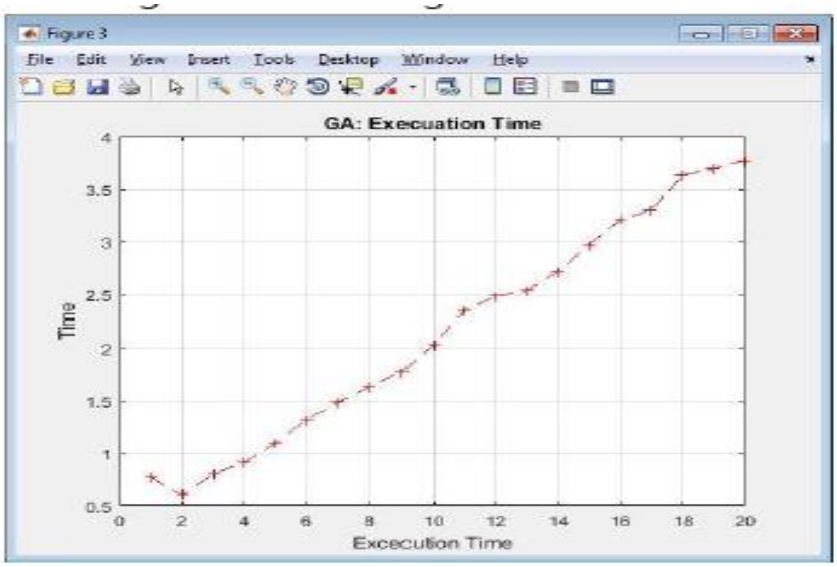

Fig 2 Genetic Algorithm Execution Time

The execution time of cloud scheduling is shown in fig 2. It indicates that execution time is decrease using genetic algorithm.

V. CONCLUSIONS

There are three parts in this report: Users, data centres, and task scheduling are all included. The cloud service task scheduling model (TSS) consists of these three components. In the user module, tasks are expected to follow a Poisson distribution. We have developed a GA-CACO algorithm, which combines GA and CACO in the scheduling module, due to the inadequacies of GA and ACO. The final thing we did was to compare GACACO, GA, and ACO to a variety of tasks. According to the test findings, the algorithm GA-CACO is the best one to use to optimize the objective function, as well as the fastest. An advantage of ACO over other techniques is its ability to provide better plans with higher statistics and mean access times, as well as more consistent job assignments. An important role in the ABC-GA model's job scheduling system is played by static data.

REFERENCES

[1] G. Singh Chabbra et al., "Qualitative Parametric Comparison of Load Balancing Algorithms in distributed Computing Environment", 14th International Conference on Advanced Computing and Communication, IEEE, Surathkal, pp. 58-61, 2006.

[2] A. Singh, P. Goyal, and S. Batra, "An optimized round robin scheduling algorithm for CPU scheduling," Int. J. Comput. Sci. Eng., vol. 02, no. 07, pp. 2383–2385, 2010

[3] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, and Shun-Sheng Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", Proc. 3rd International Conference on Computer Science and Information Technology (ICCSIT), pp.108- 113, 2010.

[4] A. Alnowiser et al., "Enhanced weighted round robin scheduling with DVFS technology in cloud," International Conference on Computational Science and Computational Intelligence (CSCI 2014), vol. 1, pp. 320–326, 2014.

[5] V. Mohammadian et al., "Fault-Tolerant Load Balancing in Cloud Computing: A Systematic Literature Review", IEEE Access, Vol. 10, pp. 12714-731, 2021

[6] G. Liu et al., “An Improved Min-Min Algorithm in Cloud Computing”, International Conference of Modern Computer Science and Applications. Advances in Intelligent Systems and Computing, vol 191. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33030-8_8.
[7] Li, X., Mao, Y., Xiao, X., and Zhuang, Y. ’An improved max-min task-schedulng algorithm for elastic cloud’, In IEEE International Symposium on Computer, Consumer and Control (IS3C), pp. 340-343, 2014.
[8] O. M. Elzeki et al., “Improved Max-Min Algorithm in Cloud Computing,” International Journal Computer Applications, vol. 50, no. 12, pp. 22-27, ISSN: 975–8887, 2012.
[9] Maheswaran, M., Ali, S., Siegal, H. J., Hensgen D. and Freund, R. F. ’Dynamic matching and scheduling of a class of independent tasks 415 onto heterogeneous computing systems’, Eighth Heterogeneous Computing Workshop, Proceedings, San Juan, pp. 30–44, 1999.
[10] S. C. Wang, K. Q. Yan, W. P. Liao, and S. S. Wang, “Towards a load balancing in a three-level cloud computing network,” 3rd IEEE International Conference on Computer Science and Information Technology, vol. 1, pp. 108–113, 2010.
[11] S. Penmatsa and A. T. Chronopoulos, "Cooperative load balancing for a network of heterogeneous computers," 20th IEEE International Parallel & Distributed Processing Symposium, 2006, doi: 10.1109/IPDPS.2006.1639393.
[12] O. M. Elzeki, M. Z. Rashad, and M. A. Elsoud, “Overview of Scheduling Tasks in Distributed Computing Systems,” International Journal of Soft Computing Engineering, vol. 2, no. 3, pp. 470–475, 2012.
[13] V. K. Prajapati, M. Jain and L. Chouhan, "Tabu Search Algorithm (TSA): A Comprehensive Survey," 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), 2020, pp. 1-8, doi: 10.1109/ICETCE48199.2020.9091743.

[14] X. Liu et al., "A comparative study of A-star algorithms for search and rescue in perfect maze," International Conference on Electric Information and Control Engineering, pp. 24-27, 2011, doi: 10.1109/ICEICE.2011.5777723.
[15] R. K. Naha , M. Othman, ”Cost-aware service brokering and performance sentient load balancing algorithms in the cloud”, Journal of Network and Computer Applications www.elsevier.com/locate/jnca Vol. 75, pp. 47–57, Aug 2016.
[16] A. M. Manasrah et al.,”A Variable Service Broker Routing Policy for data center selection in cloud analyst”, Journal of King Saud University - Computer and Information Sciences, Vol.29, No. 3, pp. 365-377, 2017, ISSN 1319-1578.
[17] B. Wickremasinghe, R. N. Calheiros and R. Buyya, "CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications," 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 446-452, 2010, doi: 10.1109/AINA.2010.32.
[18] Z. Benlalia et al., “A New service broker algorithm optimizing the cost and response time for the cloud computing”, International Symposium on Machine Learning and Big Data Analytics for Cyber Security and Privacy, pp. 992-997, Vol. 151, May, 2019.

[19] A. Khodar et al., “Evaluation and Analysis of Service Broker Algorithms in Cloud Analyst”, IEEE conference of Russian Young researchers in Electric and Electronics Engineering, St. Petersburg and Moscow, Russia, pp. 351-355, 2020, doi: 10.1109/EIConRus49466.2020.9039187.
[20] Y. Zhang et al., "Parameter Selection for a Centralized Thermostatically Controlled Appliances Load Controller Used for Intra-Hour Load Balancing," IEEE Transactions on Smart Grid, Vol. 4, issue 4, pp. 2100-2108, Dec. 2013, doi: 10.1109/TSG.2013.2258950.
[21] R. Shah et al., "On the Design of Adaptive and Decentralized Load Balancing Algorithms with Load Estimation for Computational Grid Environments," IEEE Transactions on Parallel and Distributed Systems, Vol. 18, issue 12, pp. 1675-1686, Dec. 2007, doi: 10.1109/TPDS.2007.1115.
[22] T. Deepa and D. Cheelu, "A comparative study of static and dynamic load balancing algorithms in cloud computing," International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 3375-3378, doi: 10.1109/ICECDS.2017.8390086.
[23] R. Khan et al., "Different technique of load balancing in distributed system: A review paper," Global Conference on Communication Technologies (GCCT), pp. 371-375, 2015, doi: 10.1109/GCCT.2015.7342686
[24] N. K. Rathore et al., “A sender initiate based hierarchical load balancing technique for grid using variable threshold value”, IEEE International Conference: Signal Processing, Computing and Control (ISPCC), 2013, DOI: 10.1109/ISPCC.2013.6663440
[25] M. Huang et al., “A Receiver-Initiated Approach with Fuzzy Logic Control in Load Balancing”, Journal of Computer and Communications, Vol. 8, issue 5, 2020, ISSN Print: 2327-5219
[26] G. Singh Chabbra et al., “Qualitative Parametric Comparison of Load Balancing Algorithms in distributed Computing Environment”, 14th International Conference on Advanced Computing and Communication, IEEE, Surathkal, pp. 58-61, 2006.