IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

LOAD BALANCING IN COMPUTING ALGORITHMS – AN ANALYTICAL SURVEY

[1]Mr.Bibin baby and[2]Dr.K.B Manikandan

[#1] Research Scholar, Department of Computer Science, Sri Krishna Adithya College of arts and Science, Coimbatore

[#2]Assot.Professor, Department of Computer Science, Sri Krishna Adithya College of arts and Science, Coimbatore

Abstract: *In distributed environment, load balancing is necessary for effective functioning. Load balancing for the cloud has developed into a very fascinating and significant study subject as a result of the rapid growth of cloud computing and the clients' increase need for more services and better outcomes. To create effective processes and algorithms for allocating the client's requests to accessible Cloud nodes, numerous techniques have been proposed. These methods seek to improve the Cloud's general performance while giving users better, more effective services. In this article, we examine the various strategies put out to address the load balancing and work scheduling issues in cloud computing. In order to give a comprehensive review of the most recent methods in the field, we discuss about and contrast various algorithms, particularly hybrid algorithms.*

Keywords: *Cloud Computing, Load Balancing, Task Scheduling, Cloud Storage, Replications.*

1. INTRODUCTION

In the last several years, cloud computing has gained a lot of popularity. It offers a versatile and simple way to store and access data and files as part of its services. Particularly for making huge data sets and files accessible to the expanding global user base. In order to improve and streamline processes and provide consumers with acceptable levels of speed, handling such big data sets necessitates a variety of strategies. To increase storage efficiency and download speed for consumers, it is crucial to do research in a few areas of the cloud. Platforms on the cloud are very scalable.

It can be either increased or decreased at any time based on user needs.The dynamic nature of cloud platforms necessitates efficient load balancing among all computers in order to minimize make span, task response time, energy consumption, and service interruption. If the various cloud resources are correctly load balanced, high availability of services is also provided in the event that one of the other resources is not operating as intended. In cloud computing, there are numerous task scheduling methods available.

The fundamental issue is that simple load balancing methods do not efficiently utilize resources. As a result, it lengthens the time it takes to process a request in the cloud .The virtualization ability of cloud computing hides the variety of resources, which maintains it distinct from other developments that have already been made. A set of rules known as load balancing assigns a particular task to a particular virtual machine. Depending on the virtual machine's capacity, each task requires a certain amount of time to complete. In order to extend the task's duration span and speed up response time, load balancing algorithms distribute the tasks among suitable and accessible virtual machines. In order to develop a novel approach for load balancing, this research will compare and contrast the many load balancing techniques that have been put out by researchers.

II. LITERATURE SURVEY

The most well-known contributions to load balancing in cloud computing are covered in this section. The load balancing algorithms are divided into three categories: static, dynamic, and dynamic algorithms.

A. *Static Load Balancing Algorithms*

The duties are distributed among the nodes by static load balancing methods only on the basis of each node's capacity to handle fresh requests. The only basis for the process is prior knowledge of the characteristics and capacities of the nodes. Processor speed, memory and storage size, as well as the node's most recent recorded communication performance, would all fall under this category. Static algorithms typically don't take into account dynamic changes in these properties at run-time, even though they might include knowledge about the previous prior performance. Additionally, these methods are unable to adjust as the load does throughout operation. An algorithm named CLBDM was proposed by Radojevic [1] (Central Load Balancing Decision Model). The Round Robin Algorithm, which relies on session flipping at the application layer, is improved by CLBDM. A well-known load balancing algorithm is called Round Robin [2]. Nevertheless, it routes the request to the node with the fewest connections. The enhancement to CLBDM is the calculation of the connection time between both the client and the cloud node; if this connection time exceeds a certain limit, a problem exists. If a problem is discovered, the connection will be cut off and the work will be passed on to a different node according to the standard Round Robin procedures. An automatic administrator is CLBDM. The viewpoint of a human administrator served as inspiration for the concept.

The algorithm put forth in [4] has been improved by Kumar's proposed algorithm [3]. Both methods gather data about the cloud nodes from the ant behaviour in order to determine which node should be given the task. Unfortunately, the algorithm in [4] has a synchronisation problem with the ants, and the author in [3] is attempting to fix this by giving the ants the ability to commit suicide.

The MapReduce method [6] is supplemented by the algorithm suggested in [5]. A model called MapReduce has two primary functions: mapping tasks and reducing task outcomes. Three different approaches are included in this model as well. The three techniques are group, comp, and part. Before starting to map tasks, MapReduce first runs the part method. The Map tasks are used to divide the request entity into sections at this point. The comp technique is used to compare the parts once each part's key has been saved in a hash key table. The group method then uses Reduce tasks to group the components of comparable entities.The Reduce tasks will be overburdened since many Map tasks can read and process things concurrently. Therefore, it is suggested in this study that in order to lessen the pressure on these jobs, there should be one more load balancing level added between the Map task and the Reduce task. Only the huge jobs are divided into smaller tasks by the intermediate load balancing, which then sends the smaller blocks to the Reduce tasks according to their availability.

A load balancing technique [7] utilising virtual machine to physical machine mapping was presented by Junjie for the private Cloud. The algorithm's architecture includes a resource monitor and a central scheduling controller. The scheduling controller undertakes all of the effort to determine which resource can handle the task before allocating it to that particular resource. The resource monitor, however, does its function of gathering information regarding the resources' availability. Accepting the request for a virtual machine, receiving resource details via the resource monitor, mapping tasks, and mapping tasks are the four primary stages of the procedure.For the private network, Junjie presented a load-balancing algorithm [7]. The resource with the best score will be the one selected to get the job once the controller has calculated the resources' capacity to handle tasks. The client will finally have access to the application.

B. Dynamic Load Balancing Algorithms

Algorithms for dynamic load balancing take into consideration the various node characteristics and network bandwidth. The majority of these methods depend on a combination of run-time attributes received when the chosen nodes complete the task's components and knowledge based on previously gathered data on the Cloud nodes. Based on the collected and estimated attributes, these algorithms assign the jobs to the nodes and may change their assignments dynamically. These algorithms are typically more difficult to build and necessitate continuous monitoring of the nodes and task progress. They could, however, lead to more effective load balancing because they are more accurate. Finding an algorithm to reduce data redundancy and duplication is the aim of [8].The suggested algorithm, INS (Index Name Server), combines access point selection optimization with reduplication. The process of determining the ideal selection point involves a lot of variables. The target server's position, the Hash code of the data block to be downloaded, the maximum bandwidth for downloading from the target server, the transition quality—which is determined by node weight and performance judgment chart—and the path parameter are a few of these parameters.

Based on an already-existing method named WLC [10], Ren [9] proposed a dynamic load balancing solution for cloud computing (weighted least connection). Based on the amount of connections that are available for that node, the WLC algorithm allocates tasks to that node. The assignment is then given to the node with the fewest connections after comparing the Summation of connections from each node in the cloud. The abilities of each node, such as processing power, storage space, and bandwidth, are not taken into account by WLC. The suggested algorithm is known as ESWLC (Exponential Smooth Forecast based on Weighted Least Connection). ESWLC enhances WLC by considering time series and trials.That is, ESWLC determines whether to allocate a particular task to a node after assigning that node a number of tasks and getting to know its capabilities. Based on the node's CPU power, RAM, connections, and amount of disc space currently in use, ESWLC constructs the decision. The chosen node is then predicted by ESWLC using exponential smoothing.

The dual direction downloading technique from FTP sites is what is suggested in [11]. (DDFTP). The proposed approach can also be used for load balancing in cloud computing. DDFTP divides a file of size m into m/2 segments in order to operate. The work allocated to each server node is then started processing based on a predetermined pattern. One server might begin downloading from block 0 and continuing doing so incrementally, while another might begin downloading from block m and keep doing so in decreasing order. As a result, despite operating separately, both servers will download the entire file to the client in the quickest amount of time possible given their respective capacities.As a result, the work is considered complete when the servers download two successive blocks, at which point the servers can be given new tasks. The algorithm lowers the amount of network overhead by reducing the amount of communication between the client and nodes that is required. Additionally, parameters like network load, node load, and network speed are taken into account automatically; no run-time monitoring of the nodes is necessary [12] [13].

C. Hybrid Load balancing algorithm

The dynamic computing over the internet causes cloud computing to suffer from a demand overflow. To make the best use of each virtual machine's capabilities, load balancing must be done in a way that evenly distributes the load across all VMs. Ebadifard and Babamir [14] developed a PSO-based task scheduling algorithm and altered it to incorporate load balancing. This combines load balancing with PSO. By enabling high resource usage and a high convergence, the makespan is intended to be decreased. However, their choice of environment was uniform, which was a drawback. The authors only took into account dynamic workloads with a set number of virtual machines, which is why. The algorithm's efficiency may have been increased by taking into account dynamic VMs in addition to the growing number of tasks. Makespan, Resource Utilization, and Response Time are the performance measures employed.
To reduce power usage and overall costs, Vanitha and Marikkannu [15] have suggested a dynamic, well-structured load balancing algorithm. It does not take into account other QoS measures and has a lesser fault tolerance capacity. Performance can be measured in terms of power usage.

By taking into account two primary restrictions, namely load and capacity, Polepally and Chatrapati [16] [17] have suggested a dragonfly optimization-based load balancing for cloud computing. The server-level VM load balancing was taken into account by the authors. To migrate the tasks, authors employed selection and deciding factors for the tasks and the VMs. To validate their algorithm, load, capacity, and the number of migrated jobs are used as QoS metrics. The best threshold value to gauge the level of load imbalance is found using the dragonfly algorithm. By taking into account a growing number of duties, this work could be improved.

Kruekaew and Kimpan [18][19] have presented a hybrid technique that combines the ABC algorithm with the heuristic approach. By considering the make span and load-balancing requirements, the effectiveness of the suggested technique is confirmed in both homogeneous and heterogeneous contexts. The above-mentioned QoS parameters were significantly decreased by this approach. By taking into account other QoS metrics like resource utilisation, reaction time, etc., this technique may have been applied to a real-world dataset.
A hybrid load balancing strategy based on the Q-learning algorithm and modified PSO has been proposed by Jena et al. [20]. A multi-objective fitness function has been taken into account by the authors with regard to load, energy consumption, makespan, and resource use. A homogenous environment, or one with a set number of activities and virtual machines, was used for their simulations. Real-world benchmark data and taking into account varied surroundings could help the algorithm.

Heuristic algorithms were proposed by Seema A. Alsaidy et al.[21][22][23] to support the PSO method for work scheduling. Using the longest job to fastest processor (LJFP) and minimal completion time (MCT) techniques, the PSO's particles are heuristically initialised. Starting the search using solutions based on LJFP and MCT can have a big impact on convergence speed and performance. The effectiveness of the proposed LJFP-PSO and MCT-PSO algorithms is shown by simulation results. The suggested heuristic initialised PSO outperforms contemporary task scheduling algorithms in terms of convergence and load balancing, according to comparison simulation results.It is important to note that the first particles produced by the suggested heuristic initialization of the PSO population all begin the search process from the same beginning position. A heuristic approach is used to generate this beginning position.In order to appropriately search the solution space, Hojjat Emami [24] suggested Enhanced Sun Flower Optimization (ESFO), which enhances the pollination operator of the traditional SFO method. The suggested algorithm locates a polynomial time-complexity optimal task scheduling problem.

The standard JAYA that was created to handle the continuous optimization problem was improved by K.Mishra et al.[25]. Additionally, this has been modified to address more 0s and 1s-only dynamic optimization issues, such as cloud computing task scheduling. The obtained continuous values should be converted into discrete values because our work is a dynamic problem and requires the discrete values to represent task-resource assignments. In binary JAYA, which is comparable to task-resource assignments in a task scheduling issue, particles are moved in a constrained direction that is only allowed to be either 0 or 1.

III. CONCLUSION

Adopting an effective task scheduling algorithm is essential in cloud computing for both consumers and providers. We have explored numerous static and dynamic algorithms in this study, along with their benefits and drawbacks. We only took into consideration a select few significant research articles on load balancing in cloud computing. In this work, we found that hybrid algorithms outperform static and dynamic algorithms in terms of quicker convergence, lower energy consumption, more resource utilization, and higher make span.

REFERENCES:

[1] Radojevic, B. and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." In proc.34th InternationalConvention on MIPRO, IEEE, 2011.
[2] Sotomayor, B., RS. Montero, IM.Llorente, and I. Foster, "Virtualinfrastructure management in private and hybrid clouds," in IEEEInternet Computing, Vol. 13, No. 5, pp: 14-22, 2009.
[3] Nishant, K. P. Sharma, V. Krishna, C. Gupta, KP. Singh, N. Nitin andR. Rastogi, "Load Balancing of Nodes in Cloud Using Ant ColonyOptimization." In proc. 14th International Conference on ComputerModelling and Simulation (UKSim), IEEE, pp: 3-8, March 2012.
[4] Zhang, Z. and X. Zhang, "A load balancing mechanism based on antcolony and complex network theory in open cloud computingfederation." In proc. 2nd International Conference on. IndustrialMechatronics and Automation (ICIMA), IEEE, Vol. 2, pp:240-243,May 2010.
[5] Kolb, L., A. Thor, and E. Rahm, E, "Load Balancing for MapReducebasedEntity Resolution," in proc. 28th International Conference on DataEngineering (ICDE), IEEE, pp: 618-629, 2012.
[6] Gunarathne, T., T-L. Wu, J. Qiu and G. Fox, "MapReduce in the Cloudsfor Science," in proc. 2nd International Conference on Cloud ComputingTechnology and Science (CloudCom), IEEE, pp:565-572,November/December 2010.
[7] Ni, J., Y. Huang, Z. Luan, J. Zhang and D. Qian, "Virtual machinemapping policy based on load balancing in private cloud environment,"in proc. International Conference on Cloud and Service Computing(CSC), IEEE, pp: 292-295, December 2011.
[8] , T-Y., W-T. Lee, Y-S.Lin, Y-S.Lin, H-L.Chan and J-S. Huang,"Dynamic load balancing mechanism based on cloud storage" in proc.Computing, Communications and Applications Conference(ComComAp), IEEE, pp:102-106, January 2012.
[9] Ren, X., R. Lin and H. Zou, "A dynamic load balancing strategy forcloud computing platform based on exponential smoothing forecast" inproc. International Conference on. Cloud Computing and IntelligentSystems (CCIS), IEEE, pp: 220-224, September 2011.
[10] Lee, R. and B. Jeng, "Load-balancing tactics in cloud," in proc.International Conference on Cyber-Enabled Distributed Computing andKnowledge Discovery (CyberC), IEEE, pp:447-454, October 2011.
[11] Al-Jaroodi, J. and N. Mohamed. "DDFTP: Dual-Direction FTP," inproc. 11th IEEE/ACM International Symposium on Cluster, Cloud andGrid Computing (CCGrid), IEEE, pp:504-503, May 2011.

[12] Wang, S-C., K-Q. Yan, W-P.Liao and S-S.Wang, "Towards a loadbalancing in a three-level cloud computing network," in proc. 3rdInternational Conference on. Computer Science and InformationTechnology (ICCSIT), IEEE, Vol. 1, pp:108-113, July 2010.
[13] Sang, A., X. Wang, M. Madihian and RD. Gitlin, "Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems," in Wireless Networks, Vol. 14, No. 1, pp: 103- 120, January 2008.
[14] F. Ebadifard, S.M. Babamir,**A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment,**Concurr.Comput. Practice Exp., 30 (12) (2018), Article e4368
[15] M. Vanitha, P. Marikkannu,**Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines,**Comput.Electr. Eng., 57 (2017), pp. 199-208
[16] S. Mohanty, P.K. Patra, M. Ray, S. Mohapatra,**An Approach for load balancing in cloud computing using JAYA algorithm,**Int. J. Inform. Technol. Web Eng., 14 (1) (2019), pp. 27-41
[17] V. Polepally, K.S. Chatrapati,**Dragonfly optimization and constraint measure-based load balancing in cloud computing,**Cluster Comput. (2019), pp. 1-13
[18] J.P. Mapetu, Z. Chen, L. Kong, **Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing,** Appl. Intell., 49 (9) (2019), pp. 3308-3330
[19] B. Kruekaew, W. Kimpan,**Enhancing of Artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing,**Int. J. Computat. Intell. Syst., 13 (1) (2020), pp. 496-510
[20] U.K. Jena, P.K. Das, M.R. Kabat,**Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment,**J. King Saud Univ. Comp. Inform. Sci. (2020), 10.1016/j.jksuci.2020.01.012
[21] E. Rafieyan, R. Khorsand, M. Ramezanpour,**An adaptive scheduling approach based on integrated best-worst and VIKOR for cloud computing,**Comput. Ind. Eng., 140 (2020), pp. 1062-1072
[22] A.S. Thakur, T. Biswas, P. Kuila,**Binary quantum–inspired gravitational search algorithm–based multi–criteria scheduling for multi–processor computing systems,**J. Supercomput. (2020), 10.1007/s11227-020-03292-0
[23] Seema A. Alsaidy, Amenah D. Abbood, Mouayad A. Sahib,Heuristic initialization of PSO task scheduling algorithm in cloud computing,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 6, Part A,2022,Pages 2370-2382, ISSN 1319-1578,https://doi.org/10.1016/j.jksuci.2020.11.002.
[24] HojjatEmami,Cloud task scheduling using enhanced sunflower optimization algorithm, ICT Express,Volume 8, Issue 1,2022,Pages 97-100,ISSN 2405-9595,https://doi.org/10.1016/j.icte.2021.08.001.
[25] Kaushik Mishra, JharashreePati, Santosh Kumar Majhi,A dynamic load scheduling in IaaS cloud using binary JAYA algorithm,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 8, Part A,2022,Pages 4914-4930,ISSN 1319-1578,https://doi.org/10.1016/j.jksuci.2020.12.001.