Survey of Load Balancing Algorithms in Cloud Computing

[1]Arun Pratap Singh, [2]Prof. Pritesh Jain, [3]Upendra Singh

[1,2]Patel College of Science and Technology, Indore

***Abstract*: Distributed computing has turned out to be prevalent because of its alluring components. The heap on the cloud is expanding enormously with the improvement of new applications. Load adjusting is an essential piece of distributed computing condition which guarantees that all gadgets or processors perform same measure of work in parallel measure of time. Distinctive models and calculations for stack adjusting in distributed computing has been produced with the mean to make cloud assets open to the end clients effortlessly and comfort. In this paper, we mean to give an organized and complete review of the exploration on stack adjusting calculations in distributed computing. This paper studies the best in class stack adjusting apparatuses and procedures over the time of 2004-2015. We aggregate existing methodologies gone for giving burden adjusting in a reasonable way. With this order we give a simple and succinct perspective of the fundamental model embraced by each approach.**

***Keywords*—distributed computing; stack adjusting; calculations, stack balancers;**

I. INTRODUCTION

Distributed computing gives adaptable approach to hold information and documents which includes virtualization, appropriated processing, and web administrations. It additionally has a few components like customer and conveyed servers. The point of distributed computing is to give most extreme administrations least cost whenever. These days, there are more than hundred a huge number of PC gadgets associated with the Internet. These gadgets present their demand and get the reaction immediately. Figure 1, indicates distinctive gadgets (tablet, PCs, portable workstations) interface and get to the information from a cloud at any given time. The principle targets of cloud are to diminish fetched, upgrade reaction time, give better execution, consequently Cloud is likewise called a pool of administrations [1]. Load has different sorts like, CPU stack, arrange stack, memory limit issue and so on. With regards to distributed computing, stack adjusting is to share heap of virtual machines over all hubs (end client gadgets) to enhance assets, benefit use and gives high fulfillment to clients. Because of load sharing, each hub can work proficiently, information can be gotten and sent immediately [2]. The dynamic load adjusting [3] calculation utilizes framework data while disseminating the heap. A dynamic plan is more adaptable and blame tolerant. Load adjusting empowers propel organize offices and assets for better reaction and execution. A few calculations are utilized to adjust cloud information among hubs. All the client stack is taken care of by cloud supplier for smooth provisioning of administrations. Along these lines, the proposed calculation will be utilized by cloud specialist co-op (CSP).

Load adjusting is typically connected on colossal measure of information movement and servers to circulate work. Propelled structures in cloud are embraced to accomplish speed and effectiveness. There are a few qualities of load adjusting, for example, break even with division of work over every one of the hubs, help in accomplishing client fulfillment, enhance general execution of framework, decrease reaction time, and give administrations to accomplish finish asset usage [4]. Figure 2 demonstrates the heap adjusting in distributed computing. For instance, on the off chance that we make one application on cloud and many clients are required to get to it at any one time. In this manner, reaction time to hundred individuals will be moderate and servers will wind up noticeably bustling rapidly, bringing about moderate reaction and inadmissible clients. On the off chance that we apply stack adjusting on our application, at that point work will be disseminated at different hubs and we can show signs of improvement reaction [5]. The current review does not basically talk about the accessible instruments and methods that are utilized as a part of distributed computing.

In this paper, we give an extensive outline of intelligent load adjusting calculations in distributed computing. Every calculation addresses diverse issues from various viewpoints and gives distinctive arrangements. A few constraints of existing calculations are execution issue, bigger preparing time, starvation and restricted to the earth where stack varieties are few and so on. A decent load adjusting calculation ought to evade the over stacking of one hub. The point is to assess the execution of the distributed computing load adjusting calculations that have been created over the time of 2004-2015. Whatever is left of the paper is sorted out as takes after. In area II, we look at audit changed load adjusting calculations. In Section III, the execution assessment of various distributed computing calculations have been talked about and assessed with the assistance of numerous tables. Our dialog and discoveries are condensed and the paper is finished up in segment IV.

Figure 1. A Cloud Computing Scenario [7]

II. RELATED WORK

The procedure in which the heap is partitioned among a few hubs of appropriated framework is called stack adjusting in distributed computing [6]. Load adjusting helps the distributed computing through calculations [7]. Heaps of work has been done to adjust the heap with a specific end goal to enhance execution and maintain a strategic distance from over usage of assets. Different load adjusting calculations have been talked about including round robin (RR), Min-Min, Max-Min and so on. Load adjusting calculation are isolated in two principle classes, to be specific static and dynamic [33]. Figure 3 demonstrates the order of load adjusting calculations. In this segment, we give a definite exchange on the current load adjusting calculations for cloud.

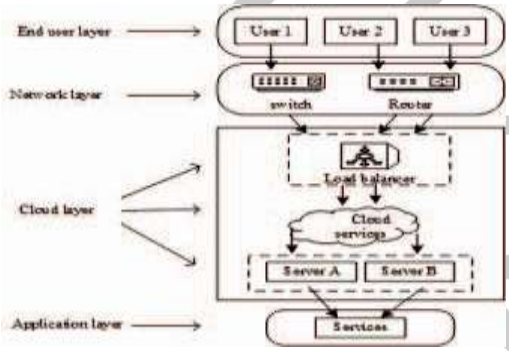

Figure 2. Load balancing in cloud computing

A. Static Algorithm : These calculations depend on finishing time of an undertaking [8].In static calculations choice about load adjusting is set aside a few minutes. These are constrained to the earth where stack varieties are few. These calculations are not needy upon the current state of framework. A static load balancer calculation partitions the activity similarly among the servers. It doesn't utilize the framework data while circulating the heap and is less intricate [9].A particular weight is doled out to the server. Server having most elevated weight gets more associations nearly. Occupation is alloted as per the capacity of the hub. In static calculation, dynamic changes at run-time are generally not thought. Alongside this, static calculation don't be able to deal with stack changes all through run-time [10].Radojevic proposed CLBDM (focal load adjusting choice model) and this is a propel type of Round Robin calculation [11]. This calculation works legitimately in a framework with low variety of load [12]. In CLBDM association time amongst customer and hub is computed [13].This calculation can be inconsistent in light of unexpected circles. Static calculation exchanges just fix measure of information [14][15]. It has no capacity to adaptation to non-critical failure [16].

1) Round Robin Load Balancer: In round robin settled quantum time is given to the occupation. Fundamental accentuation in round robin is on reasonableness and time constraint. It utilizes the ring to arrange the gathered errands. It uses rise to period to finish each errand [17] [18] [19]. If there should be an occurrence of overwhelming burden, round robin takes a long haul to complete all the given assignments. If there should be an occurrence of bigger errands it requires longer investment for finish [20]. In round robin loads are similarly dispersed to all VMs. Hardly any impediments of this calculation propose that, to accomplish superior, more than one customer associations ought not begin in the meantime [21].As the name round robin appears, it works in a roundabout example. Every hub is settled with a period cut and plays out an assignment at assigned time on its turn. It is less unpredictable [22][23][24]. Subsequently, at any minute some hub may have overwhelming burden and others may have no demand. In this way, it is not helpful for distributed computing [25][26][27][28]. This issue was handled by weighted round robin [29] where every hub is permitted to get particular number of solicitations as indicated by the appointed weight [30][31][32].

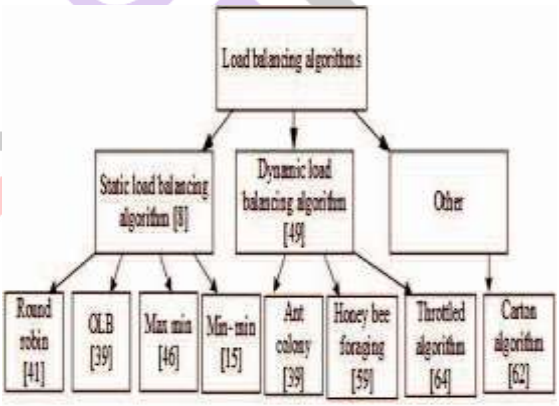

Figure 3. Classification of load balancing algorithms.

2) OLB + LBMM : Wang et al [39], recommended a mix of crafty load adjusting (OLB) and load adjusting min-min (LBMM) calculations to enhance the execution of errands. By this calculation, assets can be utilized all the more viably and it expands the assignment capability. All assignments are given to the hubs in a particular way. Its outcomes are superior to every other calculation [40] and it is utilized as a part of LBMM. LBMM works in three levels: in the principal level it acts like a demand director. It figures out how to get and apportion the errand to benefit director. At the point when the demand is gotten by benefit chief, the errand is isolated into lumps to accelerate the procedure. After that it allots these lumps to the hub. The doling out of undertakings depends on accessible hubs, cpu limit and remaining memory [38].Task finish and running time of the hub is not considered in OLB. That is the reason the undertakings set aside much time for consummation.

Regularly asks for are found in holding up list, till hubs turn out to be allowed to take different undertakings [41][42].

3) Max – Min : Max-min [46] is same as min-min calculation. However, max-min picks the undertaking with most extreme esteem and provides for the particular machine. After allocate the assignment, machine works as indicated by refreshes. These allocated assignments expel from the rundown [47]. The picked hub and undertakings orchestrate in a particular example refreshes about the prepared time are given by joining the running time of the employment [48].

Table 1. Merits And Demerits Of Load Balancing Algorithms.

Scheduling Algorithms	Merits	Demerits
Static load balancing	Decision about load balancing is made at compile time. Divides the traffic equally among the server. Fewer complexes.	Limited to the environment where load variations are few. Do not have ability to handle load changes throughput runtime.
Round Robin	Fixed time quantum.; Easy to understand; Fairness. Performs better for short CPU burst. Also used priority (running time and arrival time).	Larger tasks take long time. Can occur more context switches due to short quantum time. Job should be same to achieve high performance
Min- Min	Smallest completion time value. In presence of more small tasks, it shows best result.	Starvation Machine and tasks variation can't be predicted.
Max – Min	Requirements are prior known. So works better.	It takes long time to complete the task.
Dynamic load balancing	Distribute work at run time; Fault tolerance Only current state of system is required.	Need constant check of the nodes. Considered more complicated.
Honey Bee	Increases throughput; Minimize response time.	High priority tasks can't work without VM machine.
Ant - Colony	Faster information can be collected by the ants.; Minimizes make span.; Independent tasks; Computationally intensive	Network is over headed so search takes long time. No clarity about the number of ants.
Carton	Fairness; Good performance; Equal distribution of responses. Low communication is required.	It depends upon lower costs.
Throttled load balancing	Good performance; List is used to manage the tasks.	Tasks need to be waited.

4) Min – Min : Among every one of the assignments slightest tedious errand is look in the initial step [43]. The errand is organizing, as per that littlest time an incentive on the machine. The running time for different errands is likewise refreshed [44]. A couple of terms identified with static load adjusting are as per the following, ETC (Expected Time of Computer), OLB and MET (Minimum Execution Time) [34][35]. Min-min demonstrates best outcomes when there are little errands more in number. Starvation is a noteworthy inconvenience. [36][37]. Variety in machine and undertakings can't be anticipated through this calculation [45].

B. Dynamic Load Balancer : Dynamic calculation depends on the distinctive properties of the hubs, for example, capacities and system transmission capacity. This need consistent check of the hub and are generally hard to actualize [49][50]. Dynamic calculations are appropriate in distributed computing condition since they convey work at run time and dole out reasonable weights to the servers. A lightest weight server is seek in organize and favored by this calculation [51]. Dynamic calculations are consider more convoluted [52]. Ran [53] recommended WLC [53] (weight slightest associations), a dynamic load adjusting calculation for distributed computing. The WLC allot assignments on the premise of number of associations for existing hub. In unique load adjusting the heap circulates among the hubs all through run-time. In the event that heap balancer discovers high use of CPU the demand is send to the following hub [54]. To deal with the heap, current condition of the framework is utilized [55]. In powerful load adjusting, records and information can be downloaded with at any limitation of specific memory [56].Its advantage approaches, when any hub is fizzled. In such circumstance, it doesn't stop the framework, just its execution is influenced [57].

1) **Ant colony algorithm :** Diverse subterranean insect province calculations additionally acquaint with adjust the heap applying insect conduct for looking nourishment.

Bigger weight implies that asset has high calculation control [61]. Load adjusting subterranean insect state advancement (LBACO) adjust the heap as well as limits make traverse. All errands are thought to be commonly free and computationally serious.

2) Honey bee foraging algorithm : Dhinesh et al proposed a calculation after detail examination of searching conduct of bumble bees [58]. At the point when an under stacked VM doles out an assignment, it refreshes number of need undertakings and heap of VM to different errands in holding up list. This approach causes different procedures to pick their VM [59]. On the off chance that an assignment has high need, at that point it chooses a VM having least number of need undertakings. It doesn't contemplate just load adjusting additionally monitors needs of errands which right now expelled from substantial stacked machines [60]. It builds throughput and limits reaction time.

3) Throttled load balancing : This calculation relies on the hypothesis of appropriate inquiry of virtual machine. The undertaking supervisor makes a rundown of virtual machines. By utilizing the rundown, customer ask for apportioned to the pertinent machine. On the off chance that the size and ability of the machine is appropriate for ask for, at that point the occupation is given to that machine. This calculation is superior to round robin calculation [59][64].

4) Carton : Container [62] is a procedure that is mix of load adjusting (LB) and disseminated rate restricting (DRL). Through LB, occupations are decently allot to the servers. While DRL guarantees the equivalent dissemination of assets. Work stack is progressively dole out to enhance the execution and spread the heap similarly to every one of the servers. This calculation can undoubtedly be actualized as low correspondence required [63].

Table 2. Load Balancing Algorithms Criteria

LB algorithms	Fairness	Response Time	Throughput	Overhead	Fault tolerance	Performance	Resource utilization	Speed	Complexity
Static [25][51]	Yes	Fast	High	N/A	No	Fast	High	Fast	Low
Round Robin [41]	Yes	Fast	High	High	No	Fast	High	N/A	Low
Min-Min [15][9]	No	Fast	High	High	No	Fast	High	Fast	Low
Max-Min [44]	No	Fast	High	High	No	Fast	High	Slow	Low
Dynamic [23][11]	No	Slow	High	High	Yes	Slow	High	Fast	High
Honey Bee [59]	No	Slow	High	Low	No	Slow	High	Fast	Low
Ant colony [27][1]	No	Slow	High	High	N/A	Slow	High	Fast	No
Carton [62]	Yes	Fast	High	N/A	N/A	Fast	High	Fast	High
Throttle [32]	No	Fast	Low	Low	Yes	Fast	High	Fast	Low
OLB+ LBMM [39]	No	Slow	High	Low	No	Fast	High	Slow	High

III. PERFORMANCE EVALUATION

Table 1 depicts the assessment of the talked about Load Balancing (LB) calculations through various parameters like decency, throughput, holding up Time. In table 2, the correlation of these calculations demonstrate positive and negative outcomes and we portray this as high and low term. As examined pervious diverse calculations demonstrate distinctive outcomes. To such an extent that, Static calculation consider reasonable for convey the heap. Be that as it may, it is less mind boggling and not blame tolerant. Min-Min calculation is not reasonable and blame tolerant. If there should arise an occurrence of little errands, it indicates best outcome. In Max-Min, prerequisites are earlier known. So it works better and gives high throughput.

Alongside this, dynamic load adjusting requires just current condition of the framework and has all the more overhead and adaptation to internal failure. Bumble bee has high throughput and low reaction time. It has low overhead and execution since high need errands can't work without VM machine. Insect province is basic calculation and less perplexing. Container calculation requires low

correspondence and its working is reasonable. Table 2 gives a definite examination of various calculations over various parameters like decency, execution, speed, multifaceted nature. We suggest that, Round Robin is more proficient as per following truths, Round Robin consider reasonable for circulate the heap, it has high throughput, great reaction time and less mind boggling than different calculations. The significant favorable position of RR is time constraint and utilize measure up to period to finish each undertaking.

IV. CONCLUSION

In this paper, we have exhibited examination of various load adjusting calculations for distributed computing, for example, round robin (RR), Min-Min, Max-Min, Ant state, Carton, honey bee and so forth. We depicted points of interest and constraints for these calculations indicating brings about various conditions. The fundamental piece of this paper is examination of various calculations considering the attributes like reasonableness, throughput, adaptation to internal failure, overhead, execution, and reaction time and asset usage. The constraint of existing work is that each distributed computing calculation does not address the related issues like decency, high throughput and equity. Future work is to alleviate the above issue, and utilize the crossover way to deal with achieve better execution and secure the framework.

REFERENCES

[1] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in *Sixth Annu. Chinagrid Conf*, pp. 3–9, Aug. 2011.

[2] A. Abraham, "Genetic algorithm based schedulers for grid computing systems Javier Carretero, Fatos Xhafa," in *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 6, pp. 1–19, 2007.

[3] M. Katyal and A. Mishra, "A comparative study of load balancing algorithms in cloud computing environment." in *International Journal of Distributed and Cloud Computing*, Volume 1 Issue 2 December 2013.

[4] R. G. Rajan and V. Jeyakrishnan, "A survey on load balancing in cloud computing environments," in *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 12, pp. 4726–4728, 2013.

[5] X. Xu, H. Yu, and X. Cong, "A qos-constrained resource allocation game in federated cloud," *Seventh Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput.*, pp. 268–275, Jul. 2013.

[6] B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, I. Stoica, "Load balancing in dynamic structured P2P systems", in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, IEEE, 2004.

[7] K. Bala, S. Vashist, R. Singh, and G. Singh, "A review of the load balancing techniques," *International Journal of Advances in Computer Science and Communication Engineering (IJACSCE)*, vol. 2, no. I, march 2014.

[8] H. Chang and X. Rong, "Legendre transform-dual solution for a class of investment and consumption problems with hara utility," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[9] S. S. Chauhan and R. C. Joshi, "A weighted mean time min-min max-min selective scheduling strategy for independent tasks on grid," *Advance Computing patiala Conference (IACC), 2010 IEEE 2nd International*, pp. 4–9, 2010.

[10] I. Foster, "Virtual infrastructure management in private and hybrid clouds."in *Internet Computing, IEEE* , Volume: 13, Issue: 5, 2009.

[11] N. Haryani and D. Jagli, "Dynamic method for load balancing in cloud computing," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 16, no. 4, pp. 23–28, 2014.

[12] H. Izakian, A. Abraham, and V. Snasel, "Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments," *Int. Jt. Conf. Comput. Sci. Optim.*, pp. 8–12, Apr. 2009.

[13] B. R. Kandukuri, R. P. V., and A. Rakshit, "Cloud security issues," *IEEE Int. Conf. Serv. Comput.*, pp. 517–520, 2009.

[14] N. J. Kansal and I. Chana, "Cloud load balancing techniques : a step towards green computing," *IJCSI International Journal of Computer Science issue*, vol. 9, no. 1, pp. 238–246, 2012.

[15] H. Chen and F. Wang, "User-priority guided min-min scheduling algorithm for load balancing in cloud computing."*Parallel Computing Technologies (PARCOMPTECH), National Conference IEEE*, 2013.

[16] M. T. Cse, "Comparison of load balancing algorithms in a Cloud Jaspreet kaur," *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, vol. 2, no. 3, pp. 1169–1173, 2012.

[17] N. Sran and N. Kaur, "Comparative analysis of existing load balancing techniques in cloud computing," *International Journal of Engineering Science Invention*, vol. 2, no. 1, pp. 60–63, 2013.

[18] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of network and computer applications*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[19] N. Swarnkar, "A survey of load balancing techniques in cloud computing," *International Journal of Engineering Research & Technology*, vol. 2, issue 8, pp. 800–804, 2013.

[20] N. J. Kansal and I. Chana, "existing load balancing techniques in cloud computing : a systematic re- view," *Journal of Information Systems and Communication*, vol. 3, no. 1, pp. 87–91, 2012.

[21] R. Kaur and P. Luthra, "Load balancing in cloud computing," *Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC*, pp. 1–8, 2014.

[22] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, "Availability and load balancing in cloud computing," --*International Proceedings of Computer Science and Information Technology*, vol. 14, pp. 134–140, 2011.

[23] A. Roy, "Dynamic load balancing : improve efficiency in cloud computing," *International Journal of Emerging Research in Management &Technology*, vol. 9359, no. 4, pp. 78–82, 2013.

[24] A. A. P. Kazem, A. M. Rahmani, and H. H. Aghdam, "A modified simulated annealing algorithm for static task scheduling in grid computing," *Int. Conf. Comput. Sci. Inf. Technol.*, pp. 623–627, Aug. 2008.

[25] R. Lee and B. Jeng, "Load-balancing tactics in cloud," *Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov.*, pp. 447–454, Oct. 2011.

[26] S. Lorpunmanee, M. N. Sap, A. H. Abdullah, and C. Chompoo-inwai, "An ant colony optimization for dynamic job scheduling in grid environment," *proceedings of world academy of science, engineering and technology*, volume *23* pp. 314–321, 2007.

[27] J. Kaur and S. Kinger, "A survey on load balancing techniques in cloud computing," *International Journal of Science and Research (IJSR)*, vol. 3, no. 6, pp. 2662–2665, 2014.

[28] M. Zhou, R. Zhang, D. Zeng, and W. Qian, "Services in the cloud computing era : A Survey," *Universal Communication Symposium (IUCS), 4th International* ,2010.

[29] N. S. Raghava and D. Singh, "Comparative study on load balancing techniques in cloud computing," *open journal of mobile computing and cloud computing*, vol. 1, no. 1, 2014.

[30] B. Radojeviü and M. Žagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." *MIPRO, 2011 Proceedings of the 34th International Convention*, May 2011

[31] B. S. Rajeshwari, "Comprehensive study on load balancing," *An international journal of advanced computer technology*, vol. 3, no. Vi, pp. 900–907, 2014.

[32] M. Randles, D. Lamb, and a. Taleb-Bendiab, "A Comparative study into distributed load balancing algorithms for cloud computing," *IEEE 24th Int. Conf. Adv. Inf. Netw. Appl. Work.*, pp. 551–556, 2010.

[33] D. Rastogi and F. U. Khan, "effective fault handling algorithm for load balancing using ant colony optimization in cloud computing," *International Journal of Advances in Engineering & Technology*, no. April, 2014.

[34] P. M. Shameem and R. S. Shaji, "A methodological survey on load balancing techniques in cloud computing," *International Journal of Engineering and Technology (IJET)* vol. 5, no. 5, pp. 3801–3812, 2013.

[35] M. Abdullah and M. Othman, "Cost-based multi-qos job scheduling using divisible load theory in cloud computing," *Procedia Comput. Sci.*, vol. 18, pp. 928–935, 2013.

[36] C. T. Lin, "Comparative based analysis of scheduling algorithms for resource management in cloud computing environment," *International Journal of Computer Science International Journal of Computer Science and Engin and Engineering*, " issue. 1,volume 1, pp. 17–23, 2013.

[37] R. P. Lingawar, M. V Srode, and M. M. Ghonge, "Survey on load-balancing techniques in cloud computing," *International Journal of Advent Research in Computer & Electronics*, vol. 1, no. 3, pp. 18–21, 2014.

[38] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, and R. Rastogi, "Load balancing of nodes in cloud using ant colony optimization," *UKSim 14th Int. Conf. Comput. Model. Simul.*, pp. 3–8, Mar. 2012.

[39] J. Uma, V. Ramasamy, A. Kaleeswaran "Load Balancing Algorithms in Cloud Computing Environment - A Methodical Comparison " *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Volume 3, Issue 2, February 2014.

[40] P. Samal and P. Mishra, "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing,"*(IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 4, no. 3, pp. 416–419, 2013.

[41] D.Powar, S. Swaroop Moharana, R. D. Ramesh "analysis of load balancers in cloud computing," *International Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 2, pp. 101–108, 2013.

[42] Yatendra sahu, M. K. Pateriya "Cloud Computing Overview and load balancing algorithms", *Internal Journal of Computer Application*, Vol-65 No.24, 2013.

[43] A. Agarwal and R. N. Milind, "Computing environment," *Parallel, Distributed and Grid Computing (PDGC), 2014 International Conference* , pp. 9–13, 2014.

[44] U. Bhoi, P. N. Ramanuj, and W. S. Email, "Enhanced max-min task scheduling algorithm in cloud computing," *International Journal of Computer and Information Technology*, vol. 2, no. 4, pp. 259–264, 2013.

[45] T. Desai and J. Prajapati, "A survey of various load balancing techniques and challenges in cloud computing,"*International Journal Of Scientific & Technology Research* ,vol. 2, no. 11, 2013.

[46] X. Ren, R. Lin, and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast."

[47] S. Ray and A. De Sarkar, "E xecution a nalysis o f l oad b alancing a lgorithms in cloud computing," *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, vol. 2, no. 5, pp. 1–13, 2012.

[48] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," *Fifth Int. Jt. Conf. INC, IMS IDC*, pp. 44–51, 2009.

[49] V. Sakthivelmurugan, A. Saraswathi, and R. Shahana, "Enhancedload balancing technique in public cloud," *IJREAT International Journal of Research in Engineering & Advanced Technology* vol. 2, no. 2, pp. 1–4, 2014.

[50] R. Somani and J. Ojha, "A hybrid approach for vm load balancing in cloud using cloudsim," *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 3, no. 6, pp. 1734–1739, 2014.

[51] R. Tong, "A load balancing strategy based on the combination of static and dynamic," *Database Technology and Applications (DBTA), 2010 2nd International Workshop*, no. 2, pp. 1–4, 2010.

[52] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters," *Futur. Gener. Comput. Syst.*, vol. 37, pp. 141–147, Jul. 2014.

[53] Y. A. Younis, M. Merabti, and K. Kifayat, "Secure cloud computing for critical infrastructure : A Survey," 2013.

[54] M. Randles, D. Lamb, and A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing,'' *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops* , pp. 551–556. 2010

[55] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: a survey," *in sixth int. conf. semant. knowl. grids*, pp. 105–112, Nov. 2010.

[56] K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: challenges and algorithms," *Second Symp. Netw. Cloud Comput. Appl.*, pp. 137–142, Dec. 2012.

[57] I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas, "A Flexible Accounting Model for Cloud Computing," in *Annual SRII Global Conference*, pp. 277– 284, 2011.

[58] Hung-Chang Hsiao, et al., "Load Rebalancing for Distributed File Systems in Clouds", *IEEE transactions on parallel and distributed systems*, VOL. 24, NO. 5, MAY 2015.

[59] D. B. L.D. and P. Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2292–2303, May 2013.

[60] H. Zhong, K. Tao, and X. Zhang, "An approach to optimized resource scheduling algorithm for open-source cloud systems," *Fifth Annu. ChinaGrid Conf.*, pp. 124–129, Jul. 2010.

[61] R.-S. Chang, J.-S. Chang, and P.-S. Lin, "An ant algorithm for balanced job scheduling in grids," *Futur. Gener. Comput. Syst.*, vol. 25, no. 1, pp. 20–27, Jan. 2009.

[62] J. Hu, J. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud computing Environment", *Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, 2010.

[63] A. Daryapurkar and V. M. Deshmukh, "Efficient load balancing algorithm in cloud environment," *International Journal Of Computer Science And Applications*, vol. 6, no. 2, pp. 308–312, 2013.

[64] M. Kawser, "Performance comparison between round robin andproportional fair scheduling methods for LTE," *Int. J. Inf. Electron. Eng.*, vol. 2, no. 5, 2012.