International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-03, June 2018

A Comparative Study of Post Migration Algorithms for Dynamic Load Balancing in Cloud Computing Environment

[1]Anmol Bhandari, [2]Kiranbir Kaur

[1]Student, [2]Assistant Professor, Guru Nanak Dev University Amritsar India.

[1]anmolbhandari60@gmail.com, [2]kiran.dcse@gndu.ac.in

Abstract-Cloud computing is trending as a new platform for the purpose of distributed computing in today's IT world. It enables the user or client to access the resources worldwide without any boundation to storage and processing units. The major issue prevailing in Cloud computing is load balancing; which is distributing the dynamic local workload among the nodes in the cloud in an even manner. It ensures that no specific node is overloaded while another node is fully idle. The basic idea of load balancing is to optimize the resource consumption. To achieve optimal results, many performance metrices are considered. This paper focuses on the different existing techniques in a different cloud environment and illustrates the comparison between them.

***Keywords* —Cloud Computing; Load Balancing; Energy Efficiency; Virtual Machine; Resource Utilization**

I. INTRODUCTION

Cloud computing has been constructed by amassing two terms in the department of innovation. To begin with, the term is Cloud and the following term is Computing[1]. Heterogeneous resources team up to form a cloud. Cloud has no significance with its name and is a combination of the tremendous framework. The applications that are conveyed to end clients as services over the Internet and also the hardware, as well as system programming in data centers, are summed up in Infrastructure that is in charge of giving those services. So as to make productive utilization of these assets and guarantee their accessibility to the end clients, "Computing" is done in view of specific guidelines determined in SLA. On the basis of the user's On-Demand in the pay-as-you-say-manner, Infrastructure and hardware are made usable to the clients. In Cloud Computing, major interest lies in order to achieve maximum resource utilization coupled with higher availability at minimum cost. It obliges popular and needful changes and helps any association in keeping away from the capital expenses of equipment and programming[2][3]. Thus, A system for empowering an appropriate, on-request network access to a mutual pool of processing resources that includes systems, servers, storage devices, networks, applications, services is known as Cloud Computing[4]. We can provision or deprovision these resources in no time with minimum service provider interaction or management efforts. This further aide in advancing accessibility [4]. It has been embraced in the business generally and also an extension in datacentres is quick, this is because the growth development that is exponential in cloud computing. This huge and useful development has caused the drastic and wide increment in the utilization of energy and its effect on the earth as far as carbon footprints are concerned.

To reduce the issues of energy consumption, Load Balancing comes to the rescue by distributing the load evenly and fairly and also by minimizing the consumption of resources. The major issue in Cloud Computing is Load Balancing[5]. It is defined as the mechanism which is responsible for administering the dynamic workload across the entire cloud evenly, such as to prevent the scenario in which some virtual machines are intensely loaded and others are totally free or doing very less work. It fosters the over resource utility and performance of the system by attaining higher resource utilization ratio and user satisfaction. Efficient and fair resource utilization is also assured[6]. The bottlenecks occurring due to an imbalance of load is also avoided by this mechanism.

Two main tasks are considered in load balancing, First one is resource provisioning or allocation and the second one is scheduling of tasks in a distributed environment. Following things are ensured for the two tasks stated above:

- Easy availability of resources on demand.
- Efficient utilization of resources whether the load is high or low
- Saving of energy when the load is low
- Reduction in cost of resource utilization.

The simulation environment is required for estimating the productivity and viability of Load Balancing Algorithms. The tool that can be utilized in Cloud Modelling is CloudSim[7] which is the most productive in its sphere. Amid the lifecycle of a Cloud, CloudSim enables VMs to be overseen by the hosts which thus are overseen by datacenters. Cloudsim gives the design to four

 DOI : 10.18231/2454-9150.2018.0346

fundamental substances. These substances enable the client to set-up an essential distributed computing condition and measure the adequacy of Load Balancing Algorithms.

The fundamental target of load balancing strategies is to accelerate the execution of uses on resources whose workload changes at runtime in an erratic way [8].

When a VM is heavily loaded with multiple tasks in Cloud Computing Environment, then the tasks are expelled from the heavily loaded VM and then are submitted to any other VM in the same datacentre which is less loaded. In the scenario, when we expel numerous tasks from the VM and there is the availability of more than one VM for their execution, then these tasks are distributed in the way that there will be a fair mix and match of priorities so that no task should wait for a long time for their execution. It is done at an intra-datacenter level that is virtual machine level.

A common Cloud demonstrated utilizing CloudSim comprises of following four substances[1] Datacenters, Hosts, Virtual Machines and Application as well as System Software. Infrastructure level Services are given to Cloud Users through the Datacenter Substance. To a few Host Entities or few hosts' substances, they enact as a Shelter that will sum up to frame a solitary Datacenter element. Physical Servers that have pre-designed executing capacities are the Host in Clouds.

The host is in charge of giving Software level support of the Cloud Users. Hosts have their own memory and storage. Hosts' ability of processing is communicated in MIPS (million instructions for every second). Mapping is done to a host that matches their basic qualities like memory, programming, storage, processing and accessibility prerequisites. Accordingly, mapping of comparable examples of Virtual Machine is done to the same instance of a Host in view of its accessibility. On Virtual Machine on-request, Application and system software are executed.

Following properties are needed to be satisfied as all the load balancing algorithms are made to be utilized[3] in large distributed systems:-

· **Distributed Execution-** Algorithms do not have any or less information regarding what other parts of the algorithms are working on, they are run simultaneously on the independent hosts. There is no central coordination by the server or single physical host which can act as leader or coordinator.

· **Local Information-** All the information needed in the execution of an algorithm is collected autonomously by the local physically host. The host uses the loads of the other physical hosts which are stored on shared devices, which is executing the algorithm.

· **Simplicity and Statelessness-** At every physical host, execution should not cost any numbers of local resources. Also, Physical hosts can add itself to the cloud due to the scalability and expel itself due to the reason of invalidation, so two necessary conditions for the algorithm are simplicity and statelessness.

II. METRICES IN LOAD BALANCING TECHNIQUES

Various parameters[4] like Scalability, Resource Utilization, Performance, Migration Time, Response Time, Throughput, Associated overhead and Fault Tolerance are considered in the existing load balancing algorithms. Moreover, in addition to this, energy consumption and carbon emission should also be considered for an energy-efficient load balancing.

· Overhead Associated- It regulates the overhead included at the time of implementation of a load-balancing algorithm. The overhead comprises overhead because of inter-process and inter-processor communication, movement of tasks. For the efficient working of a technique, it should be minimal.

· Throughput- It is used for the calculation of a number of tasks completed in the given period of time. For the improvement in performance, it should be high.

· Performance- This metric is utilized for the efficiency check of the system. Performance needs to advance at a cost that is affordable, for example, reduction in response time while having acceptable delays

· Resource Utilization- This metric is for the monitoring of utilization of resources. For an efficient load balancing, it should be optimized.

· Scalability- It is a skill of performing load balancing with any definite number of nodes for a system. This need to be improved and high.

· Response Time- This is the time taken by a load balancing algorithm to respond in a distributed system. For better performance, Response time needs to be minimum.

· Fault Tolerance- The skill of an algorithm to execute its designated work that is load balancing indeed a failure of any node or link is known as fault tolerance. The algorithm should have a great fault-tolerant mechanism.

· Migration Time- The time taken to move or migrate the job or task from one host to another is known as migration time. This time needs to be minimum for better performance of the system.

· Energy Consumption(EC)- This metric gives the calculation of all the amount of energy consumed by the resources in the system. Load balancing reduces the energy consumption as it evenly distributes the load among the nodes in the system, resulting in refraining overheating.

 DOI : 10.18231/2454-9150.2018.0346

Carbon Emission(CE)- It is used to compute the amount of carbon emission by all the resources of the system. Carbon emission and energy consumption are proportional to each other; more energy consumed, then more carbon emitted. It needs to be reduced for energy-efficient load balancing technique.

III. LITERATURE SURVEY

Cloud Computing can have two types of environment- either static or dynamic which depends on how software developer designs the cloud requested by the cloud provider.

1. STATIC ENVIRONMENT- In static condition[1] the cloud supplier introduces homogeneous resources. Likewise, the resources in the cloud are not adaptable when the condition is made static. In this situation, the cloud requires earlier learning of capacity of nodes, processing or computation power, performance, memory, and measurements of client prerequisites. These client necessities are not required to any change at the runtime. Algorithms proposed to accomplish load adjusting in static condition can't adjust to the runtime changes in load. Even though the static condition is up for easy simulation, however, isn't appropriate for heterogeneous cloud condition.

The choices identified with adjusting of load[9] will be set aside at compile time when asset prerequisites are assessed. The upside of this algorithm is the straightforwardness concerning both usage and overhead since there is no compelling reason to continually screen the hubs for execution measurements. They work legitimately just when variation in the load for the VMs is low; almost static. Accordingly, these calculations are not appropriate for grid and distributed computing conditions where the load differs at different purposes of time.

Example- Round Robin, Centralised Load Balancing Model

2. DYNAMIC ENVIRONMENT- In the unique condition of the dynamic environment[1], the cloud supplier introduces heterogeneous resources which are adaptable in prevailing dynamic condition. In this situation, cloud can't depend on the earlier information though it considers run-time measurements. The necessities of the clients are conceded adaptability (i.e. they may change at run-time). The algorithm proposed to accomplish load balancing in dynamic condition can without much of a stretch adjust to runtime changes in load. A dynamic condition is hard to be simulated yet is exceedingly versatile with distributed computing condition.

Dynamic load[9] adjusting calculations roll out improvements to the circulation of a load of among nodes at run-time; they utilize current load data when making decisions regarding distribution [16].

Example- ESWLC, LBMM(Load- Balancing Min-Min).

Following are the load balancing techniques which are currently prevailing in clouds:-

a) **ROUND ROBIN ALGORITHM-** The most straightforward calculation[10] that uses the idea of time quantum or slices. In this algorithm, the time is isolated into various slices and every hub is allotted a specific time quantum or time interim and in this time quantum or slice, the hub is needed to play out its activities. On the basis of this time quantum, the resources of the service provider are given to the users. In Round Robin Scheduling the time quantum assumes an imperative part to schedule, on the grounds that if time quantum is substantial at that point Round Robin Scheduling Algorithm is same as the FCFS Scheduling. In the event that the time quantum is to a great degree too little at that point, Round Robin Scheduling is called as Processor Sharing Algorithm and context switching is high in number. It chooses the load on arbitrary premise and prompts the condition where a few hubs are vigorously loaded while some are daintily loaded. 0Despite the fact that this algorithm is extremely straightforward however there is an extra load on the scheduler to choose the extent of quantum [11] and it has a longer average waiting time than fellow algorithms, higher context switches and turnaround time and low throughput.

b) **EQUALLY SPREAD CURRENT EXECUTION ALGORITHM (ESCE)-** A load balancer is needed in this Algorithm[12] which screens the tasks which are requested execution. The lining up of the tasks and submission of them to various virtual machine[13] is the job of the load balancer. The balancer investigates the queue often for new tasks and afterward designates them to the pool of free virtual server. The balancer additionally keeps up the pool of task dispensed to virtual servers, which causes them to recognize which virtual machines are free and should be designated with new tasks. Cloud Analyst simulation is used for the experimental work of this algorithm. The name recommends this algorithm that it chips away at similarly spreading the execution load on the various virtual machine.

c) ACTIVE MONITORING LOAD BALANCING ALGORITHM- The Load Balancer[12] of this algorithm keeps up data regarding each and every VMs and also the count of requests of users that are currently distributed to which VM. At the point when a demand to allot another VM arrives, it recognizes the minimum stacked VM [14]. In the event that there are numerous VM that can fulfill the request, it chose the primary one on the list. Then Active Monitoring Load Balancer restores the VM id to the Data Center Controller then the Data Center Controller forwards the demand to the VM distinguished by that id. At last, Data Center Controller tells the Load Balancer about the new distribution.

 DOI : 10.18231/2454-9150.2018.0346

ISSN : 2454-9150 Vol-04, Issue-03, June 2018

d) **THROTTLED LOAD BALANCING ALGORITHM-** This technique[12] is totally in light of virtual machine. Here, initially, the user demands the throttled load balancer to analyze the suitable host that is efficient to handle the load effectively and also can play out the activities that are provided by the user [15]. In this technique, the primary job is requested by the user to load balancer to locate an appropriate Virtual Machine that can perform the desired task.

e) **STOCHASTIC HILL CLIMBING LOAD BALANCING ALGORITHM-** Brototi Mondal et al. [16] have designed and developed the Stochastic Hill Climbing algorithm for load balancing. This algorithm is basically a circle that ceaselessly progresses in the bearing of incrementing value, that is tough. The algorithm terminates when it achieves the pinnacle value where no adjacent nodes have a greater value. This variation picks aimlessly from among the tough moves and the likelihood of determination can fluctuate depending upon the inclination of the tough move. Along these lines, it maps assignment of values to an arrangement of different values by rolling out just minor improvements to the first value. The best component of the set is made the following assignment. This essential task is rehashed until either an answer is found or a halting paradigm is come to. It is the incomplete approach to the optimized problems.

f) **DECENTRALIZED CONTENT AWARE LOAD BALANCING ALGORITHM** - H. Mehta et al. [17] projected another load balancing strategy named as workload and client aware policy (WCAP) that depends upon content awareness. It utilizes a one of uncommon and special property (USP) that indicates the interesting and exceptional property of the solicitations and in addition processing hubs too. USP causes the scheduler to choose the hub that is best reasonable for the handling the solicitations. This system is actualized in a decentralized way and with very low overhead. So,with utilizing the substance data to limit the hunt, the strategy enhances the looking execution and consequently overall execution of the system. It likewise helps in decreasing the idle time of the computing hubs henceforth enhancing their usage capacity.

g) **SERVER- BASED LOAD BALANCING FOR INTERNET DISTRIBUTED SERVICES-** A. M. Nakai et al. [18] projected another load balancing approach that is based on servers, which are conveyed everywhere throughout the globe. This strategy benefits in diminishing the response times by utilizing a convention which restrains the redirection of user requests to the nearest remote servers and also not over-burdening them. Proposal of a middleware is done to execute this convention. It likewise utilizes a heuristic for guiding web servers to continue over-burdens.

h) **JOIN-IDLE-QUEUE-** Y. Lua et al. [19] projected a load adjusting algorithms for the web administrations that are dynamically adaptable. The technique furnishes extensive scale stack adjusting with appropriated dispatchers by, first balance the load among inactive processors crosswise over dispatchers for the accessibility of processors that are idle at every dispatcher and at that point, delegating tasks to processors so that normal queue length at every processor is diminished. Also By expelling the load adjusting work from the basic way of demand preparing, it viably lessens the framework stack, brings about no correspondence overhead at work entries and also does not increment genuine response time.

i) **A LOCK-FREE MULTIPROCESSING SOLUTION FOR LOAD BALANCING** - X. Liu et al. [20] projected an arrangement that dodges the utilization of shared memory rather than other multiprocessing load adjusting arrangements that utilize shared memory and lock technique to keep up a client session. This is accomplished by changing Linux part. By running numerous load adjusting forms in a single load balancer, this helps in enhancing the general execution of load balancer in a multi-core condition.

j) **SCHEDULING STRATEGY ON LOAD BALANCING OF VIRTUAL MACHINE RESOURCES** - J. Hu et al. [21] projected a technique on load adjusting of VM assets that is based on scheduling and that utilizes verifiable information and current condition of the system. By utilizing a hereditary algorithm, it decreases dynamic migration and thus accomplishes the best load adjusting. It also helps in settling the issue of load-irregularity and the high cost of relocation resulting in accomplishing better utilization of resources.

k) **CENTRAL LOAD BALANCING FOR VIRTUAL MACHINES**- A.Bhadani et al. [22] projected Policy for Virtual Machines which is used to adjusts the LOAD equitably in a disseminated virtual machine or cloud computing condition. It does not consider the frameworks that are fault tolerant, nevertheless, this policy no doubt increments the general execution of the system.

l) **LOAD BALANCING STRATEGY FOR VIRTUAL STORAGE (LBVS)-** H. Liu et al. [23] projected a load balancing algorithm that gives Storage as a Service and also an extensive scale net information storage model demonstrate in light of Cloud Storage. Storage virtualization is accomplished utilizing three-layered design and by utilizing two modules of load adjusting, load balancing is accomplished. This technique helps to enhance the productivity of simultaneous access by utilizing reproduction adjusting and in response to, diminishing the reaction time and also improving the limit of disaster recovery. This technique likewise helps in

 DOI : 10.18231/2454-9150.2018.0346

adaptability, the strength of the framework and enhancing the utilization rate of the storage resource.

m) **A TASK SCHEDULING ALGORITHM BASED ON LOAD BALANCING-** Y. Fang et al. [24] examined a planning system that is two-level in light of load adjusting so that dynamic prerequisites of clients are accommodated and a high resource usage is achieved in response. Load balancing is accomplished by initial mapping of jobs is done to virtual machines and afterward it is done from virtual machines to resources at the host, subsequently enhancing the response time of tasks, general performance of the distributed computing condition and utilization of resources.

n) **HONEYBEE FORAGING BEHAVIOUR-** M. Randles et al.[25] explored a load adjusting system which is decentralized in nature and that has roused from nature algorithm for self-association. It accomplishes worldwide load adjusting through nearby server activities. Execution of the framework is improved with expanded framework assorted variety however throughput isn't expanded with an expansion in framework measure. This algorithm is most suitable for the scenarios where the assorted populace of service composes is required.

o) BIASED RANDOM SAMPLING- M. Randles et al. [25] examined an adaptable and dispersed load adjusting approach which utilizes irregular testing of the system space to accomplish self-association in the manner by adjusting the load over all hubs of the system. With the high and comparative populace of resources which results in an expanded throughput by successfully using the expanded framework resources, the performance of execution of the system is enhanced. This can also debase with an expansion in the populace of assorted variety.

IV. STATE-OF-THE-ART TABLE

ALGORITHM	ENVIRONMENT	DESCRIPTION	FINDINGS
Round Robin Algorithm [10]	It works in Static Cloud Computing	1. Uses the Time quantum or slice for the load balancing 2.Resources are given to each node for a specific time quantum	1. More average waiting time 2. High context switching and Turnaround time 3. Low throughput
Equally Spread Current Execution Algorithm [12]	Cloud Environment	1. Uses LB for queuing up the requested tasks. 2. The pool of free VMs are also maintained by LB	1.Additional overhead required for finding least active VM
Active Monitoring LB Algorithm [12]	It works in Distributed Environment	1. Uses active VM Load Balancer to keep the data about each VMs and quantity of user requests. 2.Uses VM id for the load balancing technique	1.High Response Time 2.The processing power of VM not considered.
Throttled Load Balancing Algorithm [12]	It works in Static Cloud Computing Environment	1. Uses job scheduler for maintaining VM allocation table. 2. VM has two states : idle(0) or occupied(1)	1. Large Waiting time 2. Size of the task and capacity of VM is not considered.
Stochastic Hill Climbing Load Balancing Algorithm [16]	It works in Cloud Computing Environment	1.Continuously moves in the direction of incrementing value. 2. Stops when the highest value is reached	1.Incomplete approach for an optimization Problem
Decentralized Content aware [17]	It works in Distributed Computing	1. Uses the property of user requests and computing nodes so that scheduler can decide on best nodes for processing of requests 2. To narrow down the research, content information is used.	1.Increased overall performance due to the enhanced search feature. 2.Decreased Idle time of the nodes
Server-Based Load Balancing [18]	It works on Distributed Web Servers	1. To avoid remote server overloading, a protocol is used to check on redirection rates. 2. A middleware is used to support this protocol. 3. For tolerance of abrupt load changes, a heuristics are used.	1. Service time is decreased by redirecting the requests to the nearest servers. 2.Less response time
Join-Idle-Queue	It works on Cloud Data	1. At first, assigns all the idle processors to the dispatchers so that available processors at each	1. Efficient in managing system load.

 DOI : 10.18231/2454-9150.2018.0346

ISSN : 2454-9150 Vol-04, Issue-03, June 2018

[19]	Centres	dispatcher is known. 2. Next, allots tasks to processors in order to decrease the average queue length at each processor	2. No overhead communication n at the arrival of jobs. 3. Actual response time is not increased.
Lock-free multiprocessing solution for LB [20]	It works in a Multi-Core environment	1 .In single load balancer, multiple loads balancing processes are executed.	1.The overall performance of load balancer is improved.
Scheduling Strategy on Load Balancing of VM [21]	It works in Cloud Computing	1.To achieve best load balancing, genetic algorithm, historical data and the current state of the system is used.	1.The problem of high migration cost and load imbalance is solved.
Central Load Balancing for VM [22]	It works in Cloud Computing	1.For making load balancing decisions, global state information is used.	1.To improve overall performance, the load is balanced evenly and efficiently. 2. Fault tolerance is not considered
Load Balancing Strategy for Virtual Storage (LBVS) [23]	It works in Cloud Storage	1. To achieve replica load balancing module; fair share replication strategy is used, that in return controls the access of load balancing. 2. To control data writing load balancing, Writing balancing algorithm is used	1.Flexibility and robustness are enhanced. 2. Storage as a service and large scale net data storage is provided.
Task scheduling algorithm [24]	It works in Cloud Computing	1. Tasks to virtual machines are mapped and then afterward the virtual machines to host resource.	1. Task response time is improved. 2. Resource utilization is improved.
Honeybee Foraging Behaviour [25] .	It works on Large-scale cloud systems	1. Through local server actions, global load balancing is achieved.	1. Increment in system diversity, so better performance 2. With the increase in system size, throughput does not increase
Biased Random Sampling [25]	It works on Large-scale cloud systems	1. To achieve load balancing across all system nodes, random sampling of the system domain is used.	1. With a same and high population of resources, it performs better. 2. As population diversity increases, performance degraded.

V. CONCLUSION

Despite many prevailing issues like virtual machine migration, server consolidation, load balancing, energy management etc. Cloud Computing has been largely adopted by the industry. These issues have not been completely addressed and solved. The major issue that needs some extra attention is load balancing; which is to allot the changing workload to all the hosts or machines in the cloud evenly and fairly, so that high user satisfaction and resource utilization ratio is attained. These technique also assures the even and fair distribution of resources. The metrices that are focussed by the existing load balancing techniques are reducing overheads, service response time and performance improvement etc. But the factors that are related to green computing i.e. carbon emissions and energy consumption are not considered by any of the prevailing algorithms. Also, the techniques do not consider the Availability Index (AI) of the virtual machines. So a demand is prevailing to build up a vitality productive load adjusting strategy that can result in enhancement of the execution of distributed computing by reducing response time with the introduction of Availability Index, that will result in a reduction in carbon emission and energy consumption which will further help in attaining Green Computing.

REFERENCES

[1] A. M. Mayanka Katyal, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment," Int. J. Distrib. Cloud Comput., vol. 1, no. 2, p. 14, 2013.

[2] "Cloud," no. May, p. 2009, 2009.

[3] Y. Zhao and W. Huang, "Adaptive Distributed Load Balancing Algorithm Based on Live Migration of Virtual Machines in Cloud," 2009 Fifth Int. Jt. Conf. INC, IMS IDC, pp. 170–175, 2009.

[4] N. J. Kansal and I. Chana, "Cloud load balancing techniques: A step towards green computing," Int. J. Comput. Sci. Issues, vol. 9, no. 1, pp. 238–246, 2012.

 DOI : 10.18231/2454-9150.2018.0346

[5] B. P. Rimal, E. Choi, and I. Lumb, “A taxonomy and survey of cloud computing systems,” NCM 2009 - 5th Int. Jt. Conf. INC, IMS, IDC, pp. 44–51, 2009.

[6] A. M. Alakeel, “A Guide to Dynamic Load Balancing in Distributed Computer Systems,” IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 10, no. 6, pp. 153–160, 2010.

[7] Y. H. Wang and I. C. Wu, “Achieving high and consistent rendering performance of java AWT/Swing on multiple platforms,” Softw. - Pract. Exp., vol. 39, no. 7, pp. 701–736, 2009.

[8] D. Eager, “Adaptive load sharing in homogeneous distributed systems,” … , IEEE Trans., no. September, 1986.

[9] L. D. Dhinesh Babu and P. Venkata Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” Appl. Soft Comput. J., vol. 13, no. 5, pp. 2292–2303, 2013.

[10] D. Patel and P. G. Scholar, “International Journal of Modern Trends in Engineering and Research Efficient Throttled Load Balancing Algorithm in Cloud Environment,” pp. 463–481.

[11] G. Pallis and A. Vakali, “Cloud Computing,” 2011.

[12] S. G. Domanal and G. R. M. Reddy, “Load balancing in cloud computing using modified throttled algorithm,” 2013 IEEE Int. Conf. Cloud Comput. Emerg. Mark. CCEM 2013, 2013.

[13] G. R. Nitika, Shaveta, “Comparative Analysis of Load Balancing Algorithms in Cloud Computing,” Int. J. Adv. Res. Comput. Eng. Technol., vol. 1, no. 3, pp. 120–124, 2012.

[14] H. S. Mahalle, “Load Balancing On Cloud Data Centres,” vol. 3, no. 1, pp. 1–4, 2013.

[15] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, “CloudAnalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications,” Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA, pp. 446–452, 2010.

[16] B. Mondal, K. Dasgupta, and P. Dutta, “Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach,” Procedia Technol., vol. 4, pp. 783–789, 2012.

[17] H. Mehta, P. Kanungo, and M. Chandwani, “Decentralized content aware load balancing algorithm for distributed computing environments,” Proc. Int. Conf. Work. Emerg. Trends Technol. - ICWET ’11, no. Icwet, p. 370, 2011.

[18] A. M. Nakai, E. Madeira, and L. E. Buzato, “Load balancing for internet distributed services using limited redirection rates,” Proc. - 2011 Latin-American Symp. Dependable Comput. LADC 2011, pp. 156–165, 2011.

[19] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg, “Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services,” Perform. Eval., vol. 68, no. 11, pp. 1056–1071, 2011.

[20] X. Liu, L. Pan, C. J. Wang, and J. Y. Xie, “A lock-free solution for load balancing in multi-core environment,” 2011 3rd Int. Work. Intell. Syst. Appl. ISA 2011 - Proc., pp. 4–7, 2011.

[21] J. Hu, J. Gu, G. Sun, and T. Zhao, “A scheduling strategy on load balancing of virtual machine resources in cloud computing environment,” Proc. - 3rd Int. Symp. Parallel Archit. Algorithms Program. PAAP 2010, pp. 89–96, 2010.

[22] A. Bhadani and S. Chaudhary, “Performance evaluation of web servers using central load balancing policy over virtual machines on cloud,” Proc. Third Annu. ACM Bangalore Conf. - Comput. ’10, pp. 1–4, 2010.

[23] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, “LBVS: A load balancing strategy for virtual storage,” Proc. - 2010 Int. Conf. Serv. Sci. ICSS 2010, pp. 257–262, 2010.

[24] Y. Fang, F. Wang, and J. Ge, “A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing,” Lncs, vol. 6318, pp. 271–277, 2010.

[25] M. Randles, D. Lamb, and A. Taleb-Bendiab, “A comparative study into distributed load balancing algorithms for cloud computing,” 24th IEEE Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2010, pp. 551–556, 2010.

 DOI : 10.18231/2454-9150.2018.0346