Weight Balancing Techniques in Cloud Computing

Kapil Thakkar, Prof. Roopesh Sharma, Upendra Singh

Patel College of Science and Technology, Indore

***Abstract*: Load Balancing is the standout amongst the most critical parts in circulated situations. As Cloud Computing is one of the best stage that gives stockpiling of information in exceptionally insignificant cost and open forever finished the web, stack adjusting for the distributed computing has transformed into an extremely fascinating and vital examination territory. Load adjusting backings to get a high client fulfillment and utilization of asset proportion by ensuring a capable and sensible allotment of each registering asset. There are various challenges in the heap adjusting strategies, for example, security, adaptation to internal failure and so on in distributed computing conditions. Numerous scientists have been proposed a few systems to improve the heap adjusting. This paper depicts a diagram on stack adjusting plans in cloud situations. We investigate the assorted sorts of calculations that is proposed by numerous scientists to take care of the issue of load adjusting in distributed computing.**

***Keywords*: Cloud Computing, Load Balancing.**

I. INTRODUCTION

Cloud is the bunch of conveyed PCs that gives on-request computational assets over a system. A Cloud registering is turning into a propelled innovation as of late. It is theoretically conveyed framework where processing assets appropriated through the system (Cloud) and administrations pooled together to give the clients on pay-as-required premise.
Distributed computing gives everything as an administration and are sent as open, private, group, and crossover mists. The three essential administration layers of distributed computing are: Software as a Service (SaaS), where the client does not have to deal with the establishment and setup of any equipment or programming, for example, Google Online office, Google Docs, Email cloud, and so on. Stage as a Service (PaaS), where an administration is a conveyance of a figuring stage over the web where clients can make and introduce their own particular application as they require. Design of figuring stage and server is overseen by the merchant or cloud supplier. Case of PaaS is Google App Engine. Foundation as a Service (IaaS), where servers, programming, and system gear is given as an on-request benefit by the cloud supplier [1].

The primary capacity of Sharing assets, programming, data through the web are the fundamental enthusiasm for distributed computing with a plan to decrease capital and operational cost, better execution as far as reaction time and information preparing time, keep up the framework consistency and to suit future adjustment of the framework. So there are different specialized difficulties that should be tended to like Virtual Machine (VM) migration, server solidification, adaptation to non-critical failure, high accessibility and versatility. Be that as it may, the focal issue is the heap adjusting [2], [3]. It is the component of spreading the heap among different hubs of a disseminated framework to enhance both asset arrangement and employment reaction time while likewise dodging a circumstance where a portion of the hubs are having an immense measure of load while different hubs are doing nothing or sit without moving with next to no work. It additionally guarantees that all the processor in the framework or every hub in the system does around the equivalent measure of work at any moment of time [4], [5].

The objective of load adjusting is to enhance the execution by adjusting the heap among the different assets (arrange joins, focal preparing units, plate drives, and so on.) to accomplish ideal asset use, greatest throughput, most extreme reaction time, and to keep away from over-burden. Underneath figure 1 demonstrates the piece graph of cloud design [10].

II. CLASSIFICATION OF LOAD BALANCING ALGORITHMS

Load adjusting calculations can be extensively characterized into two sorts: Static calculations and Dynamic calculations. In Static Scheduling the task of assignments to processors is done before program execution starts i.e. in gather time. Booking choice depends on data about undertaking execution times, preparing assets, and so forth., which are thought to be known at aggregate time [6]. Static planning techniques are non-preemptive. The objective of static planning strategies is to limit the general execution time. These calculations can't adjust to stack changes amid run-time [7]. Dynamic booking (frequently alluded to as powerful load adjusting) depends on the redistribution of procedures among the processors amid execution time. This redistribution is performed by exchanging errands from the vigorously stacked processors to the gently stacked processors with the intend to enhance the execution of the application. It is especially valuable when the prerequisite of process is not known from the earlier and the essential objective of the framework is to boost the use of assets. The real disadvantage of the dynamic load adjusting plan is the run-time overhead because of the exchange of load data among processors and basic leadership for the choice of procedures and processors for work exchanges and the correspondence delays related with the undertaking movement itself.

The dynamic load adjusting calculations can be incorporated or circulated relying upon whether the obligation regarding the errand of worldwide dynamic booking ought to physically dwell in a solitary processor (concentrated) or the work required in settling on choices ought to be physically appropriated among processors [8]. The most vital element of settling on choices halfway is straightforwardness [6]. Be that as it may, brought together calculations experience the ill effects of the issue of the bottleneck and single point disappointment. Disseminated stack adjusting calculations are free from these issues.

Again dispersed dynamic booking can be helpful or non-agreeable. The last one is basic where singular processors act alone as self-ruling substances and land at choices with respect to the utilization of their assets free of the impact of their choice on whatever is left of the framework. In the previous one every processor has the obligation to complete its own part of the planning errand to accomplish a typical framework wide objective [6], [8].
The heap adjusting issue can be separated into two sub issues:
1. Submission of new assignment for VM provisioning and situation of VMs on have.
2 Reallocation/movement of VMs. Distinctive load adjusting calculations are there in the writing which are talked about underneath.

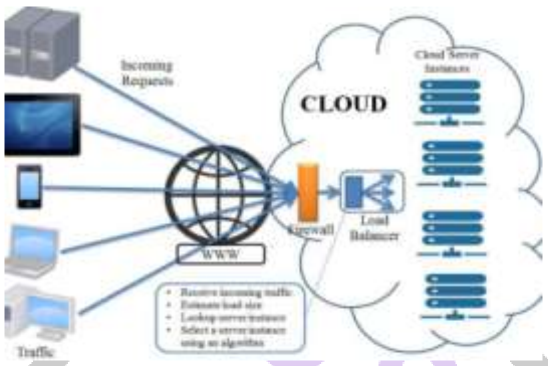

Fig. 1 Block Diagram of Cloud Architecture

III. GENERAL LOAD BALANCING ALGORITHMS FOR CLOUD COMPUTING

Round Robin Algorithm: The calculation deals with irregular determination of the virtual machines. The datacenter controller allots the solicitations to a rundown of VMs on a pivoting premise. The primary demand is distributed to a VM picked arbitrarily from the gathering and after that the Data Center controller appoints the solicitations in a round request. Once the VM relegates the demand, the VM is moved to the finish of the rundown [9]. The significant issue in this allotment is this that it doesn't consider the propelled stack adjusting necessities, for example, handling times for every individual solicitations and if the VM is not free then approaching occupation should hold up in the line.

Throttled Load Balancing Algorithm (TLB): In this calculation the heap balancer keeps up a file table of virtual machines and in addition their states (Available or Busy). The customer/server initially makes a demand to server farm to locate a reasonable virtual machine (VM) to play out the suggested work. The server farm questions the heap balancer for allotment of the VM. The heap balancer examines the record table from the top until the most readily accessible VM is found or the list table is filtered completely. On the off chance that the VM is discovered, the VM id is sent to the server farm. The server farm imparts the demand to the VM recognized by the id. Further, the server farm recognizes the heap balancer of the new allotment and the server farm modifies the list table as needs be. Amid preparing the demand of the customer, if proper VM is not discovered, the heap balancer returns - 1 to the server farm [10], [11].

Changed Throttled: Like the Throttled calculation, it likewise keeps up a list table containing a rundown of virtual machines and their states. The principal VM is chosen in an indistinguishable route from in Throttled. At the point when the following solicitation arrives, the VM at file by officially appointed VM is picked relying upon the condition of the VM and the standard strides are taken after, improbable of the Throttled calculation, where the file table is parsed from the main file each time the Data Center Queries Load Balancer for assignment of VM [12]. It gives better reaction time contrast with the past one. Be that as it may, in file table the condition of some VM may change amid the designation of next demand because of deallocation of a few undertakings. So it is not generally valuable to begin seeking from the by officially appointed VM.

Min-Min Scheduling Algorithm : It begins with an arrangement of assignments. At that point the asset which has the base fulfillment time for all assignments is found. Next, the undertaking with the base size is chosen and doled out to the comparing asset (thus the name Min-Min). At last, the undertaking is expelled from set and a similar strategy is rehashed by Min-Min until the point that all errands are allocated. The technique is straightforward, however it doesn't consider the current load on an asset before doling out an errand. So appropriate load adjust is not accomplished [13].

Load Balance Min-Min (LBMM) : This strategy utilizes Min-Min Scheduling calculation as its base. It utilizes a three level various leveled system. Demand director which is in the primary level of the design is in charge of getting the errand and relegating it to one administration chief in the second level of LBMM. Subsequent to getting the demand, benefit administrator separates it into subtasks to accelerate the handling. At that point the administration chief allocates the subtask to an administration hub for execution in view of various traits, for example, the rest of the CPU space (hub accessibility), remaining memory and the transmission rate. This calculation enhances the heap unbalance of Min-Min and limits the execution time of every hub, except does not determine how to choose a hub for a confounded undertaking requiring vast scale calculation [14].

Load Balance Improved Min-Min Scheduling Algorithm (LBIMM) : It begins by executing Min-Min calculation at the initial step. At the second step it picks the littlest size assignment from the heaviest stacked asset and figures the fulfillment time for that undertaking on every single other asset. At that point the base fruition time of that undertaking is contrasted and the make span created by Min-Min. On the off chance that it is not exactly makes pan then the errand is reassigned to the asset that deliver it, and the prepared time of both assets are refreshed. The procedure rehashes until the point that no different assets can create less fulfillment time for the littlest assignment on the intensely stacked asset than the makes pan. Hence the over-burden assets are liberated and the under stacked or sit out of gear assets are more used. This makes LBIMM to create a calendar which enhances stack adjusting and furthermore decreases the general finishing time. Yet at the same time it doesn't consider need of an occupation while booking [13].

Client Priority Awared Load Balance Improved Min-Min Scheduling Algorithm (PA-LBIMM) :
Client need is fused with the LBIMM calculation to create PA-LBIMM. This calculation will initially separate every one of the assignments into two gatherings G1 and G2. G1 is for the VIP clients' assignments having higher need prerequisite. G2 is for the common clients' assignments. The higher need errands in G1 are booked first utilizing the Min-Min calculation to dole out the undertakings to the VIP qualified assets set. At that point the errands with bring down need are booked to allocate them to every one of the assets by Min-Min calculation. Toward the end, the heap adjusting capacity is handled to streamline the heap of all assets to create the last calendar. The calculation is just worried about the makespan, stack adjusting and client need. It doesn't consider the due date of each assignment [13].

2-Phase Load Balancing Algorithm : This calculation is the mix of the OLB and LBMM to have a superior execution time and to adjust the heap all the more effectively. A line is utilized to store errands that should be done by supervisor. In the main stage OLB planning director is utilized to relegate employment to the administration administrator. In the second stage LBMM calculation is utilized to pick the reasonable administration hub to execute the subtask by the administration chief. The issue related with this approach is that it material just in static condition [14].

VMC Tune : This strategy utilizes the blend of nearby and worldwide planning. VMC Tune is conveyed to oversee hub and associates with every single physical machine which picks up stack status of all PMs and VMs occasionally and afterward reallocates asset by means of Hypervisors. VMC Tune incorporates neighborhood booking by reallocating asset of VMs inside same hub and worldwide planning by relocating VMs among hubs. The movement technique utilizes best-fit calculation to discover PM which fits the need of VM best. Show change of load on CPU and system however [15]

Load Balancing Strategy of Cloud Computing in light of Artificial Bee Algorithm: in this technique proposed a manufactured honey bee state calculation (ABC) in light of the qualities and prerequisites of distributed computing conditions. Countless synchronous solicitations with a similar sort lined in a similar server for the first ABC calculation. Thus, the nearby asset serious wonder raised and weakened load adjusting. Because of the disappointment of this instrument for the above case, an enhanced ABC is proposed. By supplanting different sorts of solicitations with the following served ask for, the kind of demand is changed. It finished the aggregation of demand and enhanced the framework throughput. Test comes about demonstrate that ABC calculation based load adjusting instrument is acclaim for its security and the enhanced ABC does well in the versatility.

Versatile disseminated Load adjusting calculation in light of Live movement of virtual machines in cloud : R. Achar et al [17] proposed Distributed intra cloud stack adjusting calculation to contrast and adjust in view of testing with achieve balance arrangement. The calculation was executed simultaneously on each host. Cost of running VM on each of the host was figured and it guarantees that VM dependably moves from pm with higher cost to those with bring down one and bigger was distinction of costs, higher was relocated likelihood of VM's.

Agreeable Scheduling Anti-stack adjusting Algorithm for Cloud: CSAAC : C. Thiam et al [18] displayed a decentralized dynamic planning approach entitled Cooperative Scheduling Anti-stack adjusting Algorithm for cloud (CSAAC). CASA proposed calculation receives the guaranteed work reaction time as the main model to assess the hubs capacities. CSAAC includes heap of hub as another standard to assess the hubs Capabilities. Every responder hub processes an expected consummation Time, asset status, essential vitality, and furthermore current load and conveys the data by methods for an ACCEPT message. The hub chose in view of a few parameters, for example, the guaranteed time to finish, vitality devoured, the hub stack between under-stack edge and over-stack edge, hub weight because of authentic communication records, and so forth. Choice strategies consider movement cost. The chose have was the one with the base vitality overwhelmed by best execution time, considered as relocation cost to lessen the heap of an over-burden have, it starts to move the slowest assignment. Determination approach will pick the undertaking that will remain the longest on the host. Strategy of limitation will then distinguish the host that will get the assignment without surpassing its abilities.

Load Balancing with Availability Checker and Load Reporters (LB-ACLRs) : P. B. Soundarabai et al. [19] proposed the idea of programming based load balancer alongside Availability-Checker and Load Reporters (LB-ACLRs) which decreases the overhead on server and the heap balancer to enhance execution in Distributed Systems. Roused by the idea of CSMA convention Load Reporter stubs were conveyed on each of the servers. This stub keeps running on the all server frameworks and gives current memory, CPU and system utilization subtle elements to the Availability Checker at each time interim. Load Reporters (LRs) refreshes the AC with all the heap subtle elements from the different accessible servers which get put away in a hash table or database. Utilizing this database, AC refreshes the fundamental that was available in LB. LB was in charge of choosing the accessible server in light of the database data and diverting the customer solicitations to the following the chose server. Just a single TCP association was required to refresh the servers' accessibility. So no multithreaded condition was utilized to gather the servers' heap and accessibility subtle elements each opportunity to time. decreases the overhead at LB have utilized Fault-Tolerant based LB-ACLR.

IV. SAVVY LOAD BALANCING ALGORITHMS FOR CLOUD COMPUTING

CLBC - savvy stack adjusted Resource assignment for parceled Cloud framework
M. R. Sumalatha et al [20] proposed DBPS (Deadline Based Pre-emptive Scheduling) and a TLBC (Throttled Load Balancing for Cloud) stack adjusting model in view of cloud dividing utilizing virtual machine. Once an undertaking was submitted to the cloud server, it was isolated into a few sub errands. The workload of the assignment was contrasted and the preparation set gathered from different virtual machines and furthermore the relative due date of that errand was anticipated utilizing tests.
These errands were appointed to the Task Manager and Deadline Based Priority Scheduling was connected on the assignment set. The planned assignment set was given to the primary controller which performs stack adjusting. The principle controller keeps up a status table where the status of the considerable number of hubs was put away. The undertaking with higher need in the planned rundown was submitted to the hub with the correct measure of assets accessible for the errand. The hub controller again checks the status of the Virtual Machine and presents the errand to the virtual machine with proper CPU and I/O measurements and afterward the assignment was executed. Lessening of Execution time and execution cost was illustrated. The heap adjusting in view of the assessed complete time of undertakings in distributed computing Y. Fahim et al [21] proposed another change of the heap adjusting by the calculation "assessed complete time stack balancer", that considers, the present heap of the virtual machine of a server farm and the estimation of the handling complete time of an assignment before any allotment, keeping in mind the end goal to beat the issues caused by the static calculations. The calculation permits cloud specialist co-ops, to enhance the execution, accessibility and amplify the utilization of virtual machines in their server farms.

Load Balancing with Optimal Cost Scheduling Algorithm : Amanpreet Chawla and Navtej Singh Ghumman [22] utilized Round Robbin calculation to plan approaching undertakings and advances the cost and timetables the assets in light of the cost. In the proposed calculation assets were assembled as bundles in each VM. At the point when the client demands for the asset the VM comprising of that bundle was executed. This procedure cuts down the execution cost of the specialist co-op.

Power Aware Load Balancing for Cloud Computing : J. M. Galloway et al [23] first accumulates the usage rate of every dynamic figure hub. For the situation that all figure hubs were over 75% use, PALB instantiates another virtual machine on the process hub with the most reduced use number. Something else, the new virtual machine (VM) was booted on the figure hub with the most astounding use (in the event that it can suit the extent of the VM).

Cloud Task planning in view of Load Balancing Ant Colony Optimization : K. Li et al [29] proposed a cloud errand booking arrangement in light of Load Balancing Ant Colony Optimization (LBACO) calculation. The primary commitment of their work was to adjust the whole framework stack while attempting to limiting the makespan of a given undertakings set. The new booking technique was recreated utilizing the CloudSim toolbox bundle. Analyses comes about demonstrated the proposed LBACO calculation outflanked FCFS (First Come First Serve) and the fundamental ACO (Ant Colony Optimization).

Load Balancing Strategy for Optimal Peak Hour Performance in Cloud Datacenters :A. K. Kulkarni and B. Annappa [24] proposed a VM stack adjusting calculation that guarantees uniform assignment of solicitations to Virtual machines notwithstanding amid top hours when recurrence of solicitations gotten in server farm was high to guarantee quicker reaction times to clients was proposed.
Honey bee MMT: A Load Balancing Method for Power Consumption Management in Cloud Computing S. M. Ghafari et al [25] proposed a heap adjusting technique called Bee-MMT (manufactured honey bee settlement calculation Minimal movement time), which utilizing honey bee state calculation (ABC) to location over used hosts. At that point with the MMT VM determination, chooses at least one VMs to relocate from them to lessen their usage. In the interim, it can discover underutilized has and in the event that it is conceivable, relocate all VMs which distributed to these hosts and afterward change them to the rest mode [25].

Errand Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization
F. Ramezani et al [31] proposed a Task based System Load Balancing strategy utilizing Particle Swarm Optimization (TBSLBPSO) to accomplish framework stack adjusting by just exchanging additional errands from an over-burden VM as opposed to relocating the whole over-burden VM. Molecule Swarm Optimization (PSO) streamlining model was utilized to move the additional errands to the new host VMs. It demonstrated that the TBSLB-PSO strategy fundamentally lessens the time taken for the heap adjusting process contrasted with conventional load adjusting approaches and the over-burden VMs was not be

stopped amid the movement procedure, and there was no compelling reason to utilize the VM pre-duplicate process. It wiped out VM downtime and the danger of losing the last action performed by a client, and expanded the Quality of Service experienced by cloud clients.

V. CLUSTER BASED LOAD BALANCING ALGORITHMS FOR CLOUD COMPUTING

A Cluster-Based Load Balancing Algorithm in Cloud Computing : S. K. Dhurandher [26] proposed group based load adjusting calculation for distributed computing. The system was partitioned into bunches. Each group had no less than one Inter Cluster Communication (ICC) hub. A Slave was the figuring component of the system. Each slave was associated with precisely one ace specifically The slave intermittently refreshes its lord with the latest estimations of the capacity, transfer speed, handling. The ace hub keeps up a table of load circulation among slaves. Every passage of the table portrays the heap on the particular slave [26]. Proposed stack adjusting calculation is partitioned into two sections: 1) Load conveyance among experts; and 2) Load appropriation from ace to slave.

1. Load dispersion among experts in light of

a. On getting the errand, the ace computes a parameter called execution factor, which shows the capacity of the ace to play out a particular assignment.

b. Based on execution factor assignment is executed inside the group or communicated to be executed by reasonable bunch.

2. Load Distribution from Master to Slave is finished with RR planning Group Based Load Balancing in Cloud Computing : S. Kapoor, and C. Dabas [32] proposed a Cluster based load adjusting calculation which considered asset particular requests of the assignments and decreased examining overhead by separating the machines into groups. Promote K-implies bunching approach was utilized to isolate VMs into group. The calculation contrasted and the current throttled and altered throttled calculations and demonstrated that the calculation gives better outcomes as far as holding up time, execution time, turnaround time and throughput.

VI. CONCLUSION

In this paper, we have studied various calculations and talked about the diverse calculations exist for Load adjusting in distributed computing and measurements for stack adjusting in cloud. In distributed computing fundamental issue is stack adjusting. Load adjusting is fundamental to disseminate the additional dynamic neighborhood workload reliably to the whole hub in the entire cloud to achieve a high client fulfillment and asset use proportion. It likewise ensures that each processing asset is circulated effectively and reasonably. Countless and distinctive sorts of delicate registering systems can be incorporated into the future for the better use and needs of the client. The diverse load adjusting strategies are additionally being looked at here.

REFERENCES

[1] A. Beloglazov, and R. Buyya, Energy efficient resource management in virtualized cloud data centers, Proc. 10th IEEE/ACM international conference on cluster, cloud and grid computing, 2010, 826-831.

[2] Qi Zhang, Lu Cheng, and Raouf Boutaba, Cloud computing: state-of-the-art and research challenges, Journal of Internet Services and Applications, 1(1), 2010, 7-18.

[3] Amandeep Kaur Sidhu, and Supriya Kinger, Analysis of Load Balancing Techniques in Cloud Computing, International Journal of Computers & Technology, 4(2), 2013, 737-741.

[4] R. Mishra, and A. Jaiswal, Ant colony optimization: A solution of load balancing in cloud, International Journal of Web & Semantic Technology, 3(2), 2012, 33.

[5] E. Caron, L. Rodero-Merino, F. Desprez, and A. Muresan, Auto-scaling, load balancing and monitoring in commercial and open-source clouds, 2012.

[6] X. Evers, W. H. CSG, CR. B. SG, I. S. Herschberg, D. H. J. Epema, and J. F. C. M. de Jongh, A literature study on scheduling in distributed systems, Delft University of Technology, 1992.

[7] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. Al-Jaroodi, A survey of load balancing in cloud computing: Challenges and algorithms, Proc. 2012 Second Symposium on Network Cloud Computing and Applications (NCCA), 2012, 137-142.

[8] Thomas L. Casavant, and Jon G. Kuhl, A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems, IEEE Trans, on Software Eng., 14(2), 1988, 141-154.

[9] M. M. D. Shah, M. A. A. Kariyani, and M. D. L. Agrawal, Allocation of Virtual Machines In Cloud Computing Using Load Balancing Algorithm, IJCSITS, 2013, 2249-9555.

[10] B. Wickremasinghe, CloudAnalyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments, MEDC project report, 22(6), 2009, 433-659.

[11] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications, Proc. 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010, 446-452.

[12] S. G. Domanal, and G. R. M. Reddy, Load Balancing in Cloud Computing using Modified Throttled Algorithm, Proc. IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2013, 1-5.
[13] H. Chen, F. Wang, N. Helian, and G. Akanmu, User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing, Proc. National Conference on Parallel Computing Technologies (PARCOMPTECH), 2013, 1-8.
[14] 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), 2010, 108-113.
[15] W. Zhou, S. Yang, J. Fang, X. Niu, and H. Song, Vmctune: A load balancing scheme for virtual machine cluster using dynamic resource allocation, Proc. 9th International Conference on Grid and Cooperative Computing (GCC), 2010, 81-86.
[16] J. Yao, and J. H. He, Load balancing strategy of cloud computing based on artificial bee algorithm. Proc. 8th International Conference on Computing Technology and Information Management (ICCM), 2012, 185-189.
[17] R. Achar, P. S. Thilagam, N. Soans, P. V. Vikyath, S. Rao, and A. M. Vijeth, Load balancing in cloud based on live migration of virtual machines, Proc. Annual IEEEI India Conference (INDICON), 2013, 1-5.
[18] C. Thiam, G. da Costa, and J. M. Pierson, Cooperative Scheduling Anti-load Balancing Algorithm for Cloud: CSAAC, Proc. 5th International Conference on Cloud Computing Technology and Science (CloudCom), 2013, 433-438.
[19] P. B. Soundarabai, A. S. Rani, R. K. Sahai, J. Thriveni, K. R. Venugopal, and L. M. Patnalk, Load balancing with availability checker and load reporters (LB-ACLRs) for improved performance in distributed systems, Proc. 2nd International Conference on Devices, Circuits and Systems (ICDCS), 2014, 1-5.
[20] M. R. Sumalatha, C. Selvakumar, T. Priya, R. T. Azariah, and P. M. Manohar, CLBC-Cost effective load balanced resource allocation for partitioned cloud system, Proc. International Conference on Recent Trends in Information Technology (ICRTIT), 2014, 1-5.
[21] Y. Fahim, E. Ben Lahmar, E. H. Labriji, and A. Eddaoui, The load balancing based on the estimated finish time of tasks in cloud computing, Proc. Second World Conference on Complex Systems (WCCS), 2014, 594-598.
[22] Amanpreet Chawla, and Navtej Singh Ghumman, Efficient Cost Scheduling algorithm with Load Balancing in a Cloud Computing Environment, International Journal of Innovative Research in Computer and Communication Engineering, 3(6), 2015.
[23] J. M. Galloway, K. L. Smith, and S. S. Vrbsky, Power aware load balancing for cloud computing, Proc. the World Congress on Engineering and Computer Science, 2011, 19-21.
[24] A. K. Kulkarni, and B. Annappa, Load balancing strategy for optimal peak hour performance in cloud datacenters, Proc. International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015, 1-5.
[25] S. M. Ghafari, M. Fazeli, A. Patooghy, and L. Rikhtechi, Bee-MMT: A load balancing method for power consumption management in cloud computing, Proc. Sixth International Conference on Contemporary Computing (IC3), 2013, 76-80.
[26] S. K. Dhurandher, M. S. Obaidat, I. Woungang, P. Agarwal, A. Gupta, A. and P. Gupta, A cluster-based load balancing algorithm in cloud computing, Proc. IEEE International Conference on Communications (ICC), 2014, 2921-2925.
[27] Jasmin James and Dr. BhupendraVerma, Efficient VM Load Balancing Algorithm for a Cloud Computing Environment, International Journal on Computer Science and Engineering (IJCSE), 4(09), 2012, 1658-1663.
[28] 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, Performance Evaluation, 68(11), 2011, 1056-1071.
[29] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, August. Cloud task scheduling based on load balancing ant colony optimization, Proc. Sixth Annual Chinagrid Conference (ChinaGrid), 2011, 3-9.
[30] Nusrat Pasha, Dr. Amit Agarwal and Dr. Ravi Rastogi, Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment, International Journal of Advanced Research in Computer Science and Software Engineering, 4(5), 2014.
[31] F. Ramezani, J. Lu, and F. K. Hussain, Task-based system load balancing in cloud computing using particle swarm optimization, International Journal of Parallel Programming, 42(5), 2014, 739-754.
[32] S. Kapoor, and C. Dabas, Cluster based load balancing in cloud computing, Proc. Eighth International Conference in Contemporary Computing (IC3), 2015, 76-8