IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

A Review on Load Balancing Algorithms in Cloud Computing Environment

Mala Yadav[1*], Jay Shankar Prasad[2]

[1,2]School of Computer and Information Science, MVN University Palwal, Haryana, India

**Corresponding author: 12ca9002@mvn.edu.in*

Available online at: www.ijcseonline.org

Accepted: 16/Aug/2018, Published: 31/Aug/2018

Abstract- Virtualized Cloud Computing environment having the ability to change the IT software industry. Various practices and technologies are influenced by this amazing platform of next stage of evolution of internet. Information technology industry modernized by facilitating flexible on-demand allocating of computing resources. These issues are driving forces for the formation of an effective load balancing algorithm. Load serves in various forms as network load, CPU load and memory. Load balancing mechanism improved resource utilization, job response time and minimize migration by distributing the load among various nodes in a distributed system. To achieve effective load balancing the situation like some of the nodes are heavily loaded and some are under-loaded must be avoided. Load balancing algorithm must ensure that every node in the network distributes equal amount of work to all the processors. Load Balancing plays a critical role in cloud computing environment. Efficient load balancing algorithm provides customer's on-demand basis in pay-as-you-use-manner resource. This paper presents several load balancing algorithms in diverse cloud computing environment.

Keywords**:** Load Balancing, Cloud Computing, Resource Allocating, Resource Scheduling.

I. Introduction

According to NIST [1], Cloud computing is defined as a "*model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and delivered with minimal managerial effort or service provider interaction*". In the field of technology cloud computing is a fusion of two terms i.e. Cloud and Computing. Cloud is a group of heterogeneous resources and Computing is performed as per condition mentioned in SLA (Service Level Agreement). Aim of Cloud computation to attain maximum resource utilization with higher accessibility at lessened cost. Resources of cloud computing are provided as services, and used according to well-established standards and best practices extended in the domain of SOA (Service Oriented Architecture). It facilitates global and easy access to cloud services in a standardized manner. Tasks/jobs are cloudlets in cloud environment. The load balancing of tasks decided by various parameters like task size, VM size, task priority, task arrival time and so on. Optimal load balancing policies are measured with metrics response time, resource utilization, execution time. The VM and cloudlets are available in different sizes. The provisioning set-ups use space-shared and time-shared policies for VMs and task units. Space-shared policy allows one VM or cloudlet to be executed at a given moment of time. The time-shared policy allows multi-task with in a VM. The objective of a scheduling policy is to pack maximum number of cloudlets in a given VM and accomplish efficient load balancing to minimize migration.

This paper is arranged in following sections. Section II describes the Load Balancing tasks in Cloud computing environment. Section III covers Load Balancing Algorithms Related Work. Section IV concludes this paper.

II. Load Balancing in Cloud Computing Environment

Load balancing delivers efficient solution of various issues of cloud computing environments. Resource allocation and Task scheduling are two major tasks handled by load balancing in distributed environment. Efficient Load balancing policies must ensure resources should be easily available on request. Resources are efficiently utilized in load variations. In low load circumstance energy must be saved and resource utilization cost should be reduced.

A superlative load balancing algorithm must ensure the following features:

- ✓ Maximum CPU utilization
- ✓ Maximum throughput
- ✓ Minimum response time
- ✓ Minimum waiting time

CloudSim [2]and CloudAnalyst [3] is the most efficient simulating tool for evaluating the efficiency and effectiveness of Load Balancing algorithms. In CloudSim VMs are managed by hosts and hosts are managed by datacenters.

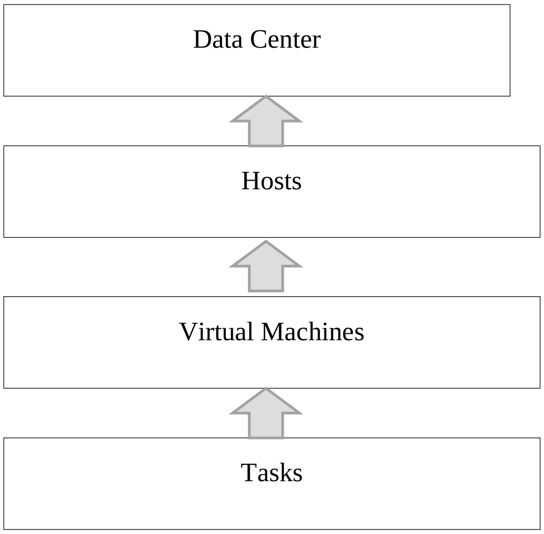

Figure 1. Cloud architecture design relationship of four basic entities

Cloudsim architecture have four basic entities Datacenters, Hosts, Virtual Machines and Application with System Software. Fundamental cloud computing environment set-up is developed by these entities which permit user to evaluate the efficiency and effectiveness of Load Balancing algorithms. Figure 1 shows a standard Cloud model with four basic entities of CloudSim.

Infrastructure Level Services are provided to the cloud users by datacenters entity. Datacenters serve as a home to several host entities. Physical Servers are hosts in clouds and have pre-configured processing capabilities. Responsible of Host is to provide Software level service to the Cloud Users. Each host have their own memory and storage. Host entity formed by combining numerous instances of virtual machine that’s why host works like a home to virtual machines. Mapping of Virtual machines to a host is performed to matches their essential features like processing, memory, software. According to availability, similar instance of virtual machine is mapped to matching instance of a host. Virtual machine executes system and application software on-demand. Therefore, to simulate cloud computing environment CloudSim object oriented methodology is preferred.

A. Resource Allocation

In resource allocation request based task of mapping of the resources to different entities of cloud is performed. Resource allocation must be done in a way that cloud do not experience any kind of wastage of bandwidth, processing capability and storage. Any node in the cloud should not be overburdened. There are two levels of resources mapping to cloud entities:

1) ***Mapping VM onto The Host:*** Host (i.e. physical servers) have virtual machines. A host maps several instances of VM subjected to its capabilities and availability. According to allocation policy host assign processing unit to VM on demand. Important features of Host and VM should be consistent must ensure by allocation policy.

2) ***Mapping Task onto VM:*** VM executes applications or tasks. Individually application needs some amount of processing power for their completion. Required processing power is provided by the VM’s to the tasks. Tasks must be mapped onto suitable VM according to its availability and configuration.

B. Task *Scheduling*

After the resource allocation task scheduling is performed on all cloud entities. Different entities are allocated as per manner which are defined by scheduling. Which resource will be available to customer according to requirements is defined by resource allocation, but in which manner the assigned resource is available to customer is defined by task scheduling. Space shared and time shared are two ways of Task scheduling. VM and Hosts can be assigned to customers in space shared way or time shared way as per algorithm design.

Resources are preempted continuously in time sharing way, but in space sharing way resources are allocated to task till end of execution (i.e. resources are not preempted). Figure 2 shows VM Scheduling in Cloud environment.
As per resource allocation and scheduling, four cases analyzed in various execution criteria so as to get proficient load balancing scheme.

- VMs and Hosts, assigned in time sharing way.
- VMs and Hosts, assigned in space sharing way.
- VMs assigned to tasks in time sharing way and Hosts assigned to VMs in space sharing way.
- VMs assigned to tasks in space sharing way and Hosts assigned to VMs in time sharing way.

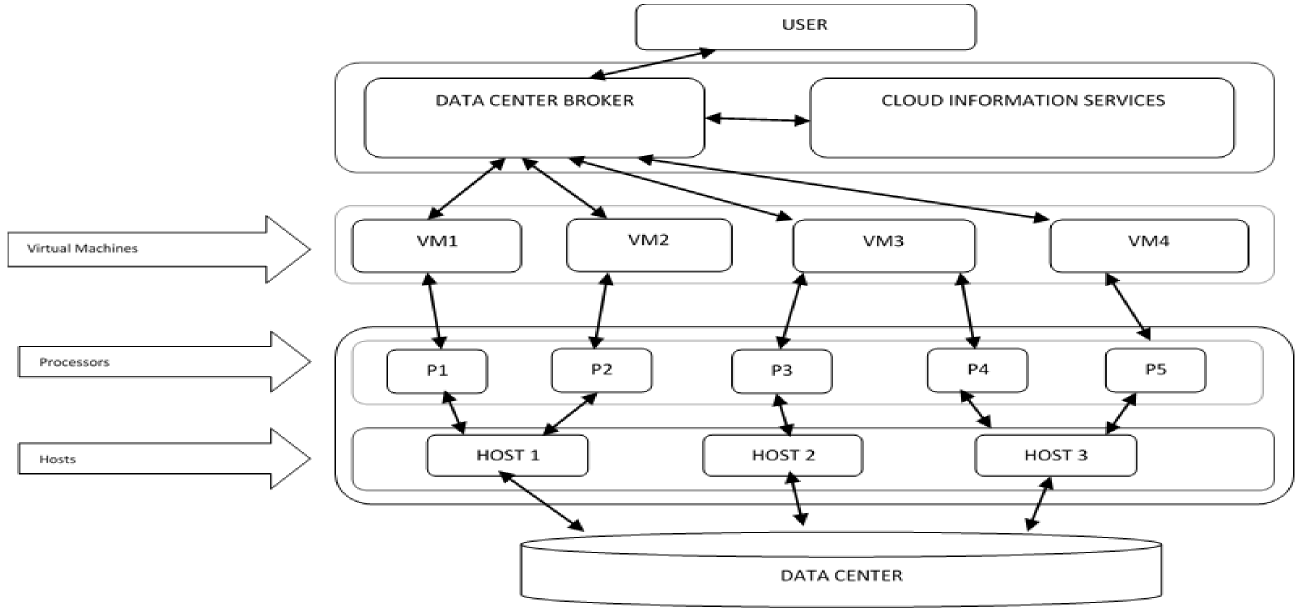

Figure 2. VM Scheduling in Cloud Computing Environment

III. Related Work of Load Balancing Algorithms

Load balancing mechanisms and approaches are of various types. Most of the investigations have been named two primary classifications [4, 5, 6]: static and dynamic. In static techniques [7-9], prior knowledge and some assumptions about the overall status of the system such as communication time, processing capability of system, job resource requirements, memory and storage devices capacity are required. A static approach is a type of assignment from a set of tasks to a set of resources which can take either a deterministic or a probabilistic form [5]. Generally, this approach is defined in the design or implementation of systems [10]. In deterministic task allocation, the extra workload of a certain node will be relocated to another particular node all of the time, but in probabilistic approaches each node sends its additional tasks with probability *P* to a node and with probability *1-P* to another one. Static load balancing algorithms do not consider the current state of the system at the time of making decisions that's why it is not a suitable approach in distributed systems as these systems change their state dynamically.

In Dynamic load balancing techniques present state of systems play important role in decisions making. The cloud provider installs heterogeneous and flexible resources in dynamic environment. Dynamic environment load balancing algorithm proposed to easily adapt to run time changes in load. The main advantage of dynamic load balancing algorithms is that tasks can move from an overloaded node to an under-loaded according to the current state of the system and state updated continuously. Dynamic load balancing algorithm design and implementation is more complicated and difficult but it provides better performance, efficient and more accurate solutions [4, 5].

Distributed and non-distributed approaches are used in design of dynamic load balancing algorithms. In distributed approaches e.g. [11-14], load balancing process can be performed by all the nodes of the system. In addition, in this approach all nodes can communicate with each other for achieving a global goal in the system which is called cooperative or every node can work independently for just achieving a local goal that is non cooperative form.

In non-distributed approach [15-18], load balancing mechanism is performed by a single node. In semi-distributed form, the system divided into some clusters or partitions and load balancing process performed by a single node in each partition.

Dynamic, distributed and adaptive load approaches are more suitable for large scale distributed systems like grid and cloud computing [10]. Figure 3 shows a general classification of load balancing algorithms and approaches [5, 10].

A. Static Load Balancing Algorithms

Cloud supplier installs homogeneous and not flexible resources in static environment. The cloud must require earlier information of processing power, nodes capacity, memory, performance and statistics of customer requirements in static environment. Customer requirements are not altered at run-time in static approach. Static environment is simple and easier to simulate but is not suitable for heterogeneous cloud computing environment.

1) Round Robin algorithm [19]

Round Robin algorithm provides load balancing in static environment. It selects the node randomly for allocating a job which is least loaded. First-cum-first-serve (FCFS) approach is applied to assign the resources to the task and organized in circular order in time sharing manner without using priority of the task. Queue of incoming requests is managed and in time scheduling manner requests are assigned to virtual machines in a datacenter. Specific time quantum is allocated to each request for execution (i.e. request is preempted) and request waits in queue for the next round. In this approach each request gets equal priority but resource utilization in less efficient.

2)Min-Min Load Balancing Algorithm [20] [21]

Min-Min load balancing algorithm is a simple static load balancing approach that perform task scheduling with excellent performance. The new task waits in a queue for execution. First of all, the cloud service manager finds the completion time of all unassigned tasks. From the unassigned tasks, minimum completion time task is selected. After that the machine(node) is selected which takes minimum time for executing all the tasks. At last, the selected tasks and the selected nodes are coordinated. Cloud service manager keep updates of node. This algorithm shows its performance with the tasks which has minimum execution time and lead the starvation problem if task has maximum execution time. Maximum execution time tasks have to wait for undefined time.

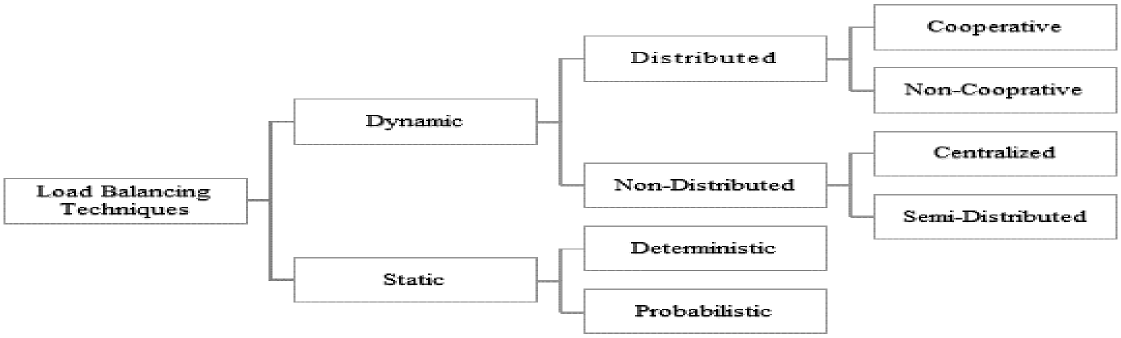

Figure 3. Classification of Load Balancing Techniques

Table 1. Comparative analysis of various static and dynamic load balancing algorithm.

Algorithms	Environment	Key Concept	Merits	Demerits	Implementation Complexity
Round Robin	Static	Follow FIFO fashion and works on dynamic time quantum	Each process get equal priority so no process will go under starvation	Less resource utilization.	LOW
Min-Min Load Balancing Algorithm	Static	Identifies resources with high makespan and then selects the task with minimum execution time.	Out performs when small tasks are greater in number.	Starvation for maximum execution time	LOW
Max-Min Load Balancing Algorithm	Static	Identifies resources with high makespan and then selects the task with maximum	Better make span and selection of resources	Starvation for minimum execution time	LOW

		execution time			
Equally Spread Current Execution	Dynamic	Task handled with priority. Size of the process specify the priority.	Take less time, and give maximum throughput	No fault tolerance	LOW
Throttled Load Balancing Algorithm	Dynamic	Allocate VM's according to indexing and matching config.	High Load Movement Factor	High communication	HIGH
Honeybee Foraging Algorithm	Dynamic	An optimization algorithm motivated by the natural seeking behavior of honey bees	Reducing energy consumption using VM migration, Considering cloud QoS	High complexity for selecting best overloaded host No prediction for future workload of hosts	HIGH
Active Clustering	Dynamic	Optimizes job assignment by linking similar services	Increase in throughput	Network more complex	HIGH
Stochastic Hill Climbing Algorithm	Dynamic	Load balancing mechanism uses soft computing approaches and a local optimization of stochastic hill climbing	Observed effective at certain problems due to simplicity, speed	Algorithm is completely centralized and executed by a single node it causes bottleneck and a single point of failure	LOW
Ant Colony Optimization (ACO) Algorithm	Dynamic	Avoid deadlock by and maximum resource utilization	Each process get equal priority so no process will go under starvation	Most of the time processor remains idle	LOW
Genetic algorithm(GA)	Dynamic	First Randomly Initialize the population and then encode them into binary strings	Efficient Load balance, and low migration	Genetic-based algorithm	HIGH
Particle Swarm Optimization (PSO) Algorithm	Dynamic	Iterative selection of particle delivering VM's to neighboring host.	Particle Move in multidimensional search space to find the best location in that space	High throughput, More refined finite element formulations	LOW

3) Max-Min Load Balancing Algorithm [22] [23]

Max-min load balancing algorithm works in a similar fashion as Min-Min algorithm works apart from that first it finds minimum completion time of tasks and then select the maximum value. Machine which execute all the tasks in minimum time is selected. At last the selected node and the selected task are coordinated. Cloud service manager keep updates of node. This algorithm avoids starvation problem of the tasks with maximum completion time but minimum completion time tasks still waits in queue till all the maximum completion time tasks finished.

B. Dynamic Load Balancing Algorithms

Cloud provider installs heterogeneous resources in dynamic environment. In dynamic environment resources are flexible. In this environment cloud must have run-time information. User requirements may change at run-time. Dynamic environment algorithm should be designed in such a way that they can adapt run time modifications in load to achieve efficient load balancing. Management and simulation of the dynamic environment is difficult but it is extremely adaptable with cloud computing.

1) Equally Spread Current Execution

Equally Spread current execution [24] [25] is a dynamic load balancing algorithm. This algorithm handles task according to priority. Size of the task decides the priority. The responsibility of load balancer is to distributes the load on to different nodes randomly. According to the size of task, load is transferred to the lightly loaded Virtual Machine which handles the task easily and take less time, and give maximum throughput.

2) Throttled *Load Balancing Algorithm [24]:*

Throttled load balancing algorithm is a dynamic approach. In this algorithm, client requests load balancer to find an appropriate virtual machine which execute the given tasks in efficient way. Virtual machine has multiple instances in cloud computing environment. So the virtual machines grouping can be performed according to the kind of requests which are handled properly by that group. When load balancer receives the clients request, the first job of load balancer is to find the appropriate group for that request. After that load balancer allocate the task to the lightly laden instance of the group.

3) Honeybee Foraging Algorithm [24]

Honeybee Foraging algorithm [25] is a dynamic approach of load balancing and it works according to nature of honeybees. Finders and reapers are the variety of bees. First of all, honey sources are searched by finder honeybees. When they find the source, they do waggle dance after returning back to the honey comb. Quality and quantity of honey available are signified by the waggle dance. After that reapers collect the honey from the sources. After collecting and returning back reapers also do dance to reflect how much honey left. It is a decentralized load balancing approach inspired by the behavior of honeybee. In this algorithm a separate queue is maintained by each node (virtual server). After executing the request from the queue, profit is calculated similarly the quality and quantity signifies by the bees through their dance. Server exist if it gives high profit, otherwise it forages. Each node profit calculation is an extra overhead. Due to the additional queue and the calculation overhead it gives low throughput.

4) Active Clustering

Active Clustering [26,27] is a clustering based dynamic algorithm. In cloud computing environment the idea of clustering is given by this approach. Cluster of nodes can improve the performance of algorithm. Each cluster work as a group. In this approach similar nodes are grouped together for processing. First of all, a node selects a neighbor node that node is called as a matchmaker node. this matchmaker node is of different type. Then a connection is made by the matchmaker node to its neighbor node. This neighbor node is similar to the initial node. Matchmaker node detached at the end. Iteratively this process is performed. In this approach availability of resources is high that increases the throughput. But performance degrades with an increase in system diversity [3].

5) Stochastic Hill Climbing Algorithm [28]

Stochastic Hill Climbing (SHC) is a variations of Hill Climbing algorithm and provide the solution for optimization problem. Complete and incomplete are two methods of this procedure. This algorithm applied incomplete method for optimization problem solving. This local optimization approach for increasing the value uninterruptedly shifts in the upward direction. It will automatically stop if not any neighbor has a higher value. The operation is repeated until either the solution found or non-neighbor has a high value. This algorithm has two main modules candidate generator and evaluation criteria. Candidate generator perform mapping process for one solution candidate to a set of probable inheritors. Evaluation criteria perform grading of each valid solution so that improving the evaluation is recorded to get to better solutions.

6) Ant Colony Optimization (ACO) Algorithm

In 2012, Ratan Mishra [29] presented "Ant colony optimization" (ACO) to avoid deadlock situation in cloud. The implementation performed using time shared and space shared scheduling strategies. Experimental results show it consume less memory and give high performance.

Kumar Nishant et al. [30] proposed a modified version of Ant Colony Optimization (ACO). ACO is applied for cloud computing load balancing so that system works efficiently in usage hours. Head node of this AI-based solution generates ants. Ants traverse the width and length of the cloud network in the way that they are familiar about the position of under-loaded or overloaded nodes. Pheromone table is updated along with ants movements for keeping the resource utilization information. Movement of ant are proposed in forward and backward as in original ACO. According to these and pheromone table, the loads transferred from overloaded nodes to under-loaded ones.

In 2014, EktaGupta [31] proposed another ACO algorithm, head node is arranged in such a way that maximum number of neighboring nodes gets helps to navigate in most possible directions of the network. The ants originate from head node continuously. Congestion of network avoided by limited number of ants and each ant have a suicide timer, the ant will halt itself when it reaches zero. Timer value depend on the size and number of nodes in the network. Ants traverse the width and length of the network in such a manner that they know about the location of under loaded or overloaded nodes in the network. Pheromone table is updated according to ants traversal. This algorithm contributes efficient use of resources.

7) Genetic algorithm(GA)

In 2013, Kousik Dasgupta [32] proposed "Genetic algorithm" (GA) to deliver an efficient utilization of resource in cloud environment and also guarantees QOS. A very simple approach of GA has been used and supposed that all the jobs are of the same priority. First, randomly population is initialized for processing unit and after that encoded into binary strings. Next each population is evaluated for fitness value in crossover step which is followed by mutation step. In this process small value is picked as mutation probability. This process is performed repeatedly till the fittest chromosome (or optimal solution) is searched or the ending criteria matched (i.e. maximum number of repetition). GA compared with three commonly used scheduling algorithms First come first serve (FCFS), Round robin (RR), Scholastic hill climbing (SHC). Result analysis indicates that the proposed algorithm for load balancing outperforms a few existing techniques.

8) Particle Swarm Optimization (PSO) Algorithm

In 2013, Elina Pacini [33] proposed a cloud VM scheduler based on "Particle swarm optimization" (PSO). This algorithm considers all hosts of cloud as swarm and each host in cloud considered as particle in the swarm. In search space each particle has a position. Host is searched to find the optimal solution by comparing velocity difference with neighboring host in each iteration. Lower load host is selected to find the optimal solution. At every iteration each particle moves towards one of its neighbor, having minimum load among all. If all neighbors overloaded, then no move performed. Each particle updates its position and velocity as per their own experience and as per their neighbor. Due to multidimensional search space this algorithm offers high throughput.

IV. Conclusion

Load Balancing is an essential task and main challenge in Cloud Computing environment to achieve maximum utilization of available resources. In this paper, we have presented a survey on various load balancing schemes, each having some advantages and disadvantages. Static load balancing scheme well performed in homogenous and stable environment but fails in heterogeneous cloud computing environment. Simulation and monitoring of static algorithm is easier. Dynamic load balancing algorithm are challenging to design, manage and simulate but are excellent suited in heterogeneous environment of cloud computing and provide better performance and more accurate and efficient solutions. Distributed, dynamic and adaptive load balancing approaches are more suitable for large scale distributed systems such as cloud computing.

References

[1] http://www.nist.gov/itl/cloud/

[2] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", in Software: Practice and Experience (SPE), Volume 41, Number 1, ISSN: 0038-0644, Wiley Press, New York, USA., pp. 23-50,2011.

[3] B. Wickremasinghe, R. N. Calheiros and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analyzing cloud computing environments and applications", in Proc. of Proceedings of the 24th International Conference on Advanced Information Networking and Applications (AINA2010), Perth, Australia, pp.446-452, 2010.

[4] K. A. Nuaimi, et al. A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms. in Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on. 2012. IEEE.

[5] A. M. Alakeel, A guide to dynamic load balancing in distributed computer systems. International Journal of Computer Science and Information Security, 2010. 10(6): p. 153-160.

[6] C. Zenon, M. Venkatesh, and A. Shahrzad, Availability and Load Balancing in Cloud Computing. 2011.

[7] Y. Alexeev, et al. Heuristic static load-balancing algorithm applied to the fragment molecular orbital method. in High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for. 2012. IEEE.

[8] K. Chonggun, and H. Kameda, Optimal static load balancing of multi-class jobs in a distributed computer system. IEICE TRANSACTIONS (1976-1990), 1990. 73(7): p. 1207-1214.

[9] S. Penmatsa, and A.T. Chronopoulos, Game-theoretic static load balancing for distributed systems. Journal of Parallel and Distributed Computing, 2011. 71(4): p. 537555.

[10] A. Khiyaita, et al. Load balancing cloud computing: state of art. in Network Security and Systems (JNS2), 2012 National Days of. 2012. IEEE.

[11] B. Cosenza, et al. Distributed load balancing for parallel agent-based simulations. in Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on. 2011. IEEE.

[12] J. Shi, C. Meng, and L. Ma. The Strategy of Distributed Load Balancing Based on Hybrid Scheduling. In Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on. 2011. IEEE.

[13] A. Gonzalez-Ruiz and Y. Mostofi, Distributed load balancing over directed network topologies. in American Control Conference, 2009. ACC'09. 2009. IEEE.

[14] I. Riakiotakis, et al., Distributed dynamic load balancing for pipelined computations on heterogeneous systems. Parallel Computing, 2011. 37(10): p. 713-729.

[15] I. Ahmad, and A. Ghafoor. A semi distributed load balancing scheme for large multicomputer systems. In Parallel and Distributed Processing, 1990. Proceedings of the Second IEEE Symposium on. 1990. IEEE.

[16] I. Ahmad, and A. Ghafoor, Semi-distributed load balancing for massively parallel multicomputer systems. Software Engineering, IEEE Transactions on, 1991. 17(10): p. 987-1004.

[17] W. Zhu, C. Sun, and C. Shieh. Comparing the performance differences between centralized load balancing methods. in Systems, Man, and Cybernetics, 1996., IEEE International Conference on. 1996. IEEE.

[18] S. Das, H. Viswanathan, and G. Rittenhouse. Dynamic load balancing through coordinated scheduling in packet data systems. in INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. 2003. IEEE.

[19] S. Ray and A. Sarkar "Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment" International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.5, October 2012.

[20] G Liu, J Li, J Xu "An improved min-min algorithm in cloud computing", Proceedings of the 2012 International Conference of Modern Computer Science and Applications; Berlin Heidelberg: Springer.2013. p.47-52.

[21] T. Kokilavani, DD Amalarethinam "Load balanced min-min algorithm for static meta-task scheduling in grid computing", International Journal of Computer Applications. 2011 Apr;20(2):43-9.

[22] U. Bhoi, PN Ramanuj "Enhanced max-min task scheduling algorithm in cloud computing", International Journal of Application or Innovation in Engineering and Management. 2013 Apr; 2(4):259-64.

[23] N. Balaji, A. Umamakeshwari "Load balancing in virtualized environment - A survey", IndianJournal of Science and Technology. 2015 May 1; 8(S9):230-4.

[24] S. Moharana, Rajadeepand and Digamber Power "ANALYSIS OF LOAD BALANCERS IN CLOUD COMPUTING" International Journal of Computer Science and Engineering ISSN 2278-9960 Vol. 2, Issue 2, May 2013.

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

[26] N. Sran, N. kaur "Comparative Analysis of Existing Load balancing techniques in cloud computing", International Journal of Engineering Science Invention, Vol-2 Issue-1 2013.

[27] N. J. Kansal, I. Chana "Existing Load Balancing Techniques in cloud computing: A systematic review" Journal of Information system and communication Vol-3 Issue-1 2012.

[28] B. Mondal, K. Dasgupta, P. Dutta "Load balancing in cloud computing using stochastic hill climbing - A soft computing approach". Procedia Technology. 2012 Dec 31; 4:783-9.

[29] R. Mishra and A. Jaiswal, "Ant colony Optimization: A Solution of Load balancing in Cloud", in International Journal of Web & Semantic Technology (IJWesT), Vol.3, No.2, pp. 33-50, 2012

[30] K. Nishant, et al. Load Balancing of Nodes in Cloud Using Ant Colony Optimization. in Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on. 2012. IEEE.

[31] E. Gupta, V. Deshpande, "A Load Balancing Technique for Servers of Datacenter of Cloud using Ant Colony Optimization", ISSN (Online): 2347 - 2812, Volume-2, Issue -6,7, 2014.

[32] K. Dasgupta, B. Mandal, P. Dutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing" in Proc. of Elsevier, Procedia Technology 2013.

[33] E. Pacini, C. Mateos and C. G. Garino, "Dynamic Scheduling based on Particle Swarm Optimization for Cloud-based ScientificExperiments" HPCLatAm 2013 VI Latin American Symposium on High Performance Computing.

AUTHORS PROFILE

MALA YADAV received her MCA in computer science from Guru Jambheshwar University Hisar, MPhil in computer Science from Chaudhary DeviLal University Sirsa. She has 10 years of Teaching experience. She is currently pursuing Ph.D. Her area of research interest includes Cloud Computing, Artificial Intelligence, Datacenter Networking and Optimization.

JAY SHANKAR PRASAD research interest is Applications of AI techniques to solve the issues of data centre, wireless networks path optimization, Computer Vision issues, to apply the soft computing techniques to control the Humanoid Robots, Gesture Recognition, ISL Recognition, Pattern mining etc. He has published Twenty-Two papers in International journals and International conferences. He has 17 years of teaching and 3 years of software industry experience. He also guided many postgraduate and undergraduate level projects.