2016 IEEE Eighth International Conference on Advanced Computing (ICoAC)

A Survey of Soft Computing Techniques Applied in Cloud Load Balancing

Sridevi S
Department of Information Technology
Anna University, Chennai

Dr. V. Rhymend Uthariaraj
Department of Information Technology
Anna University, Chennai

***Abstract*—Load Balancing (LB) of tasks in cloud is an NP hard problem. NP hard class of problems poses the challenge of proving that a solution's value is near optimum without knowing what the optimum value is. Hence applying heuristic techniques to solve the load balancing problem becomes imperative. Reaching a satisfactory solution to the task load balancing problem is a major research area in cloud environment. Numerous algorithms to solve load balancing problem are available. Bio-inspired algorithms are presented based on thecharacteristics of living beings. Various animals and birds inspire us to apply their behaviors to this problem. Such soft computing techniques available for load balancing in cloud are studied and analyzedin this paper. A summary of all the techniques and their variants are discussed. Taxonomy, algorithmic complexities, applications and contributions of these approaches are presented.The LB algorithms are classified and their performances are compared.**

Keywords—cloud; computing; load balancing; scheduling; soft computing; heuristics; ACO; BCO; GA; PSO;

I. INTRODUCTION

A number of researchers around the globe are now focusing on enhancing cloud performance. Various soft computing techniques are applied for cloud load balancing. These techniques draw interest as they are bio-inspired and are analogous to real life scenarios. The behavior of animals and birds inspire us to model them for solving various computing problems. Cloud environment puts forward a problem of balancing the load evenly and utilizing the resources efficiently. Load is the amount of computational work a system performs per unit time. The load on a node in the cloud depends on the computational load, network delay load and the amount of memory used. Load balancing mainly focuses on maximizing utilization and improving response time.

Real world entities attract our interests to apply in such optimization problems. Birds and animals like honey bees, ants, glowworms and fireflies showcase intelligent behavior in solving their day to day problems. Adopting such intelligent behavior to optimize load on computing nodes have proved that the aggregated performance of the cloud can be improved.

Customers are primarily expecting a reduction in the overall completion time of tasks running on the cloud.[20] Such high level of QoS requirements are to be supported by solving the NP hard problem of task load balancing. Heuristics are to be applied to solve such NP hard problems. Various soft computing techniques are proposed which aims to improve the performance of cloud by balancing the load evenly.

II. RELATED WORK

A. Soft Computing Techniques

Soft computing, in particular, fuzzy logic, mimics the ability of human mind [1] to model logical thinking and reasoning. These techniques are approximate rather than exact. Accuracy and precision are compromised as a tradeoff for the huge computational cost involved. Genetic algorithms, fuzzy logic, gradient programming and neural networks are the major areas under soft computing. Formalizing the neural model which is employed by humans and other animals is the main focus of soft computing techniques.

B. Issues in Task Load Balancing (TLB)

Massive computing nodes and expensive hardware resources are now available at the stroke of a button. It is a result of the huge revolution brought forth by cloud computing technology. Cloud users' major concerns include security, trustworthiness, availability and quick response. Incoming task requests are to be efficiently handled by the cloud. A balance in the amount of work done by the Virtual Machines (VM) is directly proportional to the overall throughput achieved by the cloud environment.Allocating a task to the most suitable resource available is a major research area. Task scheduling can be done aiming to improve efficiency or to balance the load or for improving the energy efficiency. TLB problem can be solved using various soft computing techniques available. Few issues surrounding TLB are computational cost, number of task migrations, delay due to load balancing, handling heterogeneous resources.

III. TAXONOMY OF LOAD BALANCING ALGORITHMS

A broad classification of LB algorithms is presented here. Its classification depends on the cloud environment, spatial distribution, task dependencies and initiation of LB process.

The type of the LB algorithm to be used is to be decided aprior based on various factors such as task arrival distribution, level of autonomy or centralization necessary, nature of incoming task requests and the components involved in the balancing process.

A. LB based on cloud environment

Various transfer policy strategies in cloud environment include static, dynamic and adaptive. They are outlined as follows:

978-1-5090-5888-4/16/$31.00@2016 IEEE

- Static: Selection and transfer policies are hard wired in the algorithm using aprior knowledge of the system. Static algorithms are best suited when the instances of the cloud are of homogeneous nature. But strict policies of scheduling the task and balancing the load may incur a lot of waiting time of certain tasks.
- Dynamic: The transfer policy used for load balancing dynamically changes as per incoming task requests, the available amount of resources and the current number of requests handled by the VMs.
- Adaptive: The system states and other dynamically changing parameters are considered on-the-go and an adaptive strategy of static and dynamic transfer policy is applied in few LB algorithms.

B. LB based on spatial distribution of nodes

- Centralized: Transfer of tasks is centralized. Used in master slave architectures. This kind of spatial distribution of node is prone to single point of failure.
- Distributed: The nodes act independently to reach a common goal. This mode of operation is best suited for peer-to-peer architecture.
- Hierarchical: Balancing is done in level wise approach. Global and local level of balancing is followed here. Computational overhead is high in this case.
- Semi-distributed: It is a mediator between centralized and distributed. It is best suited in large distributed environments.

C. LB based on task dependency

- Dependent: The tasks which execute after taking up results and resources released from other tasks.
- Independent: These are free standing tasks which execute with available data and resources without depending on other tasks.

D. LB based on initiation

- Sender initiated: Here the LB algorithm is initiated by an overloaded node and attempts to migrate the task to an underloaded VM. The overloaded node is the sender and underloaded n ode is the receiver. In this case, an already overloaded VM is burdened with the job of selecting underloaded VMs to migrate the task.
- Receiver initiated: An underloaded node sends its willingness to carry out tasks of the VMs which are overloaded. This approach relieves overloaded VMs from doing extra computation to select a receiver.
- Symmetric: Both senders and receivers initiate the LB process. They search through the VM list to find good candidates for sending and receiving tasks thereby the load balance is maintained.

TABLE I. TAXONOMY OF LOAD BALANCING ALGORITHMS

Sl. No.	*Based on*	*Type*	*Description*	*Usage*	*Areas Well Suited*	*Pain Areas*
1	Cloud Environment	Static	Hard wired selection and transfer policy	Use aprior knowledge to do hard wired task migrations	Homogeneous resource and tasks	Long Waiting times
		Dynamic	Progressive load estimated transfer policy	Useful in changing environments	Heterogeneous tasks and resources	Computation overhead
		Adaptive	Combination of static and dynamic load balancing	Useful for handling tasks with varied priorities	Applications requiring quick response	Culmination of pros in static and dynamic
2	Spatial Distribution	Centralized	Centralized transfer of tasks	Used in master slave scenarios	Highly adaptive and consistent	Single point of failure
		Distributed	Independent operation of hosts for a common goal	Used in peer-to-peer architecture	Highly scalable	Overhead on each host
		Hierarchical	Global and local level of balancing is done	Level wise approach is followed	Heterogeneous multi rack clouds	High computation overhead
		Semi Distributed	Mediator between centralized and distributed	Improving performance by adopting distributed and centralized methods	Large distributed environments	Complex design and implementation
3	Task Dependencies	Dependent	Contingent tasks which execute from results obtained from other tasks	Models real world tasks	For pipelined scheduling of tasks	Challenge is to reduce the waiting time of tasks
		Independent	Free-standing tasks which execute with available data	When tasks are self contained	For parallel scheduling of tasks	Challenge is to increase the throughput

Sl. No.	*Based on*	*Type*	*Description*	*Usage*	*Areas Well Suited*	*Pain Areas*
4	Initiation of LB process	Sender	Overloaded node initiates the LB process	On demand LB initiation	Applications which require reactive load balancing	Pressure on the overloaded node to initiate LB
		Receiver	Underloaded nodes sends willingness to share the load	Relieves overloaded node from extra computation	Proactive load balancing	Persistent wait by underloaded nodes to balance the load
		Symmetric	Both sender and receiver participates in LB initiation process	Combination of the two	Symmetric initiation of LB for critical applications	Quite an amount of time spent in LB initiation by both sender and receiver

IV. Soft Computing Approaches to LB

Various soft computing approaches are applied for cloud load balancing problem. These approaches are heuristic based and are bio-inspired. The classification of these algorithms, the methods, metrics of comparisons and major contributions identified are thoroughly studied and tabulated as follows:

A. Techniques based on ACO

Ant Colony Optimizations (ACO) based techniques are applied for load balancing of tasks in cloud environment. Few modifications are applied on the basic ACO algorithm to improve the performance of the load balancing strategy. The taxonomy, methodology and performance analysis is shown in tables II and III.

ACO suffers from random initialization, scalability degrades with increasing number of ants, complex parameter selection and pheromone update procedure.

B. Techniques based on BCO

Bees possess a great deal of intelligence in honey foraging act. They act in unison and achieve the goal collectively.

Few major works applying honey bee intelligence in load balancing are discussed in tables V and VI.

TABLE II. Techniques Based on ACO - Methodology

Sl. No.	**Method Overview**				
	Author(s)	**Method**	**Considerations**	**Methodology**	**Contributions**
1	Dorigo.M. [3]	Basic ACO (2006)	Amount of pheromone is proportional to n(ants). Pheromone evaporation is not considered.	Mimicking ant's ability to find optimal path to food	Minimize the makespan of tasks
2	Kun Li et. al. [4]	LB ACO (2011)	Tasks are mutually independent; non preemptive and computationally intensive	ACO with current status of VM and LB capabilities	Minimize makespan and balance load among nodes
3	Z. Zhang, X. Zhang [5]	Scaled ACO (2010)	Small-world and scale-free distributions	Evolution of complex networks to adapt to ant's pheromone intensity due to load	ACO is applied for LB in complex networks
4	S.Banerjee, I.Mukherjee P.K.Mahanti [6]	Modified ACO (2009)	Online mode and batch processing modes	Uses a modified version of pheromone updating factor and scheduling probability	Service load distribution to minimize makespan and increase probability of servicing the request

TABLE III. Techniques Based on ACO – Performance& Taxonomy

Sl. No.	**Performance Overview**			**Taxonomy Overview**			
	Method	***Parameters***	***Metrics***	***Cloud Environment***	***Spatial Distribution***	***Task Dependency***	***LB Initiation***
1	Basic ACO [3]	Task makespan	Completion time Time : $O(n \log n / \rho)$ [22] Space : $O(n^2)$ [3]	Dynamic	Distributed	Both	Sender
2	LB ACO [4]	Task execution time and load level	Balance degree	Dynamic	Distributed	Mutually independent and non preemptive	Sender

Sl. No.	Performance Overview			Taxonomy Overview			
	Method	*Parameters*	*Metrics*	*Cloud Environment*	*Spatial Distribution*	*Task Dependency*	*LB Initiation*
3	Scaled ACO [5]	Load distribution	Load levels	Dynamic	Distributed	Not mentioned	Symmetric
4	Modified ACO [6]	Makespan and probability of servicing request	Execution time and responsiveness	Static and Dynamic modes	Distributed	Both	Symmetric

TABLE IV. TECHNIQUES BASED ON BCO - METHODOLOGY

Sl. No.	Method Overview				
	Author(s)	*Method*	*Considerations*	*Methodology*	*Contributions*
1	L.D.,Krishna [7]	HBB-LB (2013)	Non preemptive and free standing tasks with priority	The tasks (HB) from overloaded VMs are assigned to under loaded priority wise ordered VMs	Mathematical evidences and correlations of HBBand HBB-LB algorithm
2	Kim et al. [8]	EBABC (2013)	Global exploration and local exploitation and search space	Exploration and Exploitation in spread searching based on BinaryABC	Extensive verification using convergence trends and testing of hypothesis
3	M.Rathore, S.Rai, N. Saluja [9]	HB Galvanizing (2015)	Population size (PN), Most Cycle range (MCN), Search Limit	Increased HB foraging technique with random stealing	Fitness function for quick convergence

TABLE V. TECHNIQUES BASED ON BCO - PERFORMANCE

Sl. No.	Performance Overview			Taxonomy			
	Method	*Parameters*	*Metrics*	*Cloud Environment*	*Spatial Distribution*	*Task Dependency*	*LB Initiation*
1	HBB-LB [7]	Level of balance and priorities of tasks	Imbalance degree, no. of tasks migrations	Dynamic	Distributed	Non preemptive and independent tasks	Sender
2	EBABC [8]	Control and Spread searching	Makespan and Convergence trend	Dynamic	Distributed	Dependent tasks with FRS(Flexible Ranking Strategy)	Symmetric
3	HB Galvanizing [9]	PN,MCN and limit to diversify search	Resource Util, Response time, Idle time	Dynamic	Distributed	Preemptive tasks	Symmetric

BCO follows random initialization procedure which may degrade the performance. There are several tunable parameters in consideration and complex probabilistic search is involved leading to decreased performance.

C. Techniques based on GA

Genetic Algorithm (GA) is a nature inspired algorithms which is based on 'survival of the fittest' concept. GA is applied in load balancing of tasks among VMs in cloud and has proven to be one of the efficient algorithms. Variants of GA, the methodology and major contributions, along with performance and taxonomy based tabulations are given in tables VI and VII

GA suffers from slow convergence rate and risk of premature convergence. There is no guarantee of finding global maxima and the method to arrive at fitness normalization / selection is completely based on trial and error.

D. Techniques based on SO

Swarm intelligence deals with algorithms based on methods followed by birds and fishes during migration, glowworms and particulate matter moving as a group. Such algorithms when applied to TLB problem, proves to outperform ACO based algorithms.

Few variants of Swarm Optimization (SO) algorithms are discussed in tables VIII and IX.SO methods are not applicable for non-coordinate systems and partial optimism is observed.

TABLE VI. TECHNIQUES BASED ON GA - METHODOLOGY

Sl. No.	Method Overview				
	Author(s)	***Method***	***Assumptions***	***Methodology***	***Contributions***
1	J. Page, J.Naughton [10]	GA Based Hetero System (2005)	Processors are assumed to have fixed execution rate	Batchedgenetic processing of tasks arriving as different distributions	A rebalancing heuristic is given to improve quality of results
2	Dasgupta et al. [11]	GA Based (2013)	Tasks are assumed to be of same priority	Mechanisms of natural selection and genetics	Performance - delay cost comparisons and QoS guarantees
3	J.Gu, J.Hu, T. Zhao, G. Sun [12]	New schedulin g based on GA (2012)	Task granularity is large and transferred data is large	Population initialization through spanning tree and terminates with the tree with heat restriction requirement	Average load distance measure based comparisons
4	Tingting et al. [13]	JLGA (2014)	Elite population survives many iterations	Double fitness adaptive job spanning time and load balancing GA	Greedy algorithm to initialize population, Elitism is adopted

TABLE VII. TECHNIQUES BASED ON GA – PERFORMANCE& TAXONOMY

Sl. No.	Performance Overview			Taxonomy			
	Method	***Parameters***	***Metrics***	***Cloud Environment***	***Spatial Distribution***	***Task Dependency***	***LB Initiation***
1	GA Based Hetero system [10]	Relative error based fitness function, re balancing heuristics	Makespan, response time Time complexity: $\Theta(H^2)$	Batched - Dynamic	Heterogeneous Distributed	Unknown no. of independent tasks	Symmetric
2	GA Based (2013) [11]	GA parameters	Balance degree, makespan Time complexity: $O(n_1 + ck + (n_2+1)(3m)$	Dynamic	Distributed	Both depended and independent	Symmetric
3	New scheduling based on GA (2012) [12]	Minimize cost and maximize load balance	Load vibration rate, Migration cost, Utilization rate	Dynamic	Centralized - Spanning tree structures	Both	Symmetric
4	JLGA (2014) [13]	SCALE and genetic operators	Total / Avg execution time, load on the VM	Dynamic	Centralized scheduling	Computational and independent	Symmetric

TABLE VIII. TECHNIQUES BASED ON SO - METHODOLOGY

Sl. No.	Method Overview				
	Author(s)	***Method***	***Assumptions***	***Methodology***	***Contributions***
1	N. Krishnanand, D. Ghose [14]	Glowworm SO (2009)	Apriori info about objective function is not available	Fitness is calculated using luciferin value corresponding to the luminescence quality	Simultaneous computation of multi modal objective functions
2	J. Kennedy, R. Eberhart [15]	PSO (1995)	Agents are considered to be collision-proof birds	Nearest neighbor velocity matching and craziness measure	Flying feasible solutions in hyperspace accelerating to reach optimum
3	Liu, Wang [16]	PSO Based VMs (2012)	Fitness function is assumed to be based on only two parameters	Swarm changes directions, scatters, re groups until they reach goal	PSO is proved to be better than ACO

TABLE IX. TECHNIQUES BASED ON SO – PERFORMANCE& TAXONOMY

Sl. No.	Performance Overview			Taxonomy			
	Method	***Parameters***	***Metrics***	***Cloud Environment***	***Spatial Distribution***	***Task Dependency***	***LB Initiation***
1	Glowworm SO (2009) [14]	Cognitive and social parameters	No. of runs, avg. no. of peaks, mean distance travelled by the agent, computation time, convergence iterations	Dynamic	Distributed glowworms	Dependent	Symmetric
2	PSO (1995) [15]	13 parameters resulting in 13-D space	Convergence of local optima to global one	Dynamic	Distributed flying agents	Dependent	Symmetric
3	PSO Based VMs (2012) [16]	Pbest and gbest	Performance measured using fitness functions	Dynamic	Distributed	Dependent	Symmetric

E. Other algorithms

Other algorithms based on hill climbing method, firefly method, VM capacity and current VM load based distribution methods exist. Few such algorithms which provide good performance results are tabulated below:

These algorithms rely on mathematical models to solve the given task scheduling and load balancing problem. They do not consider one important factor of data locality. All these algorithms consider only the CPU processing requirement whereas data locality is an important consideration to improve response time and to decrease data migration during request processing.

TABLE X. OTHER ALGORITHMS - METHODOLOGY

Ref	Method Overview				
	Author(s)	***Method***	***Assumptions***	***Methodology***	***Contributions***
1	B.Mondal,K. Dasgupta, P. Dutta [17]	SHC (2012)	Assumes 5%of registered users are online during simulation	Candidate generator maps to possible successors and ranks to move uphill towards optima	Outperforms RR and FCFS using uphill climbing strategy
2	A. Florence, V. Shanthi [18]	Firefly (2014)	Decision parameters are considered as load on the node	Load index table is used to calculate the optimal allocation	population gen, cal. of scheduling index, optimization of list
3	Kuo-Qin et al. [19]	OLB	Assigns tasks intuitively	Assign arbitrarily irrespective of expected task execution times	Simple and less LB overhead
4	Naghibzadeh, M [20]	Load Bal MinMin	Availability of aprior task completion times	Min execution time tasks assigned to VMs with min processing capacity	Lesser makespan
5	Naghibzadeh, M [20]	Load Bal MaxMin	Availability of aprior task completion times	Max execution time tasks assigned to corresponding VMs with given processing capacity	Minimize cost incurred from executing lengthy tasks
6	Sridevi. S, Chitra Devi D, V. Rhymend Uthariaraj [21]	WRR++ (2015)	Resource requirements of tasks are not considered	Processing capabilities of VMs, load on the VM, job lengths and job interdependencies are taken into account	Proved that time shared heterogeneous execution with task preemption outperforms basic WRR

TABLE XI. OTHER ALGORITHMS - PERFORMANCE& TAXONOMY

Ref	Performance Overview			Taxonomy			
	Method	***Parameters***	***Metrics***	***Cloud Environment***	***Spatial Distribution***	***Task Dependency***	***LB Initiation***
1	SHC [17]	Available VMs and Job requests	Response time and throughput	Dynamic	Centralized	Independent	Central node
2	Firefly [18]	Attraction coefficient between fireflies	Resource utilization and balanced load	Dynamic	Distributed	Dependent	Receiver
3	OLB [19]	Arbitrary assignment	Computational overhead O(mt)	Static	Centralized	Independent	Central node
4	Load Bal MinMin [20]	Execution times and processing capacity	Minimize makespan $O(mt^2)$	Static	Centralized	Independent	Central node
5	Load Bal MaxMin [20]	Execution times and processing capacity	Level of balance and makespan $O(mt^2)$	Static	Centralized	Independent	Central node
6	WRR++ [21]	Job length. Job interdependencies, capacity of VM and current load on VMs	Execution time of tasks, responsiveness	Dynamic	Distributed	Both Dependent and independent	Sender

V. CONCLUSION AND FUTURE WORK

Soft computing techniques follow fuzzy, incremental or evolutionary approaches to solve hard problems. Various algorithms based on such soft computing techniques available for load balancing in cloud are analyzed and surveyed in this paper. Extensive survey shows that taxonomy of algorithm is important while selecting an algorithm for a particular application. All the available algorithms have a complex search procedure without any guarantee to efficient execution of tasks within deadline. Overly concentrating on searching for optimal solution without considering data locality, leads to decreased efficiency and increased response time. On quantitative terms, response time of tasks depends on percentage of data available in the current VM rather than searching and bringing the data to the VM where task is allocated. The key point inferred from this study is that data locality is to be considered for load balancing. Tradeoffs are to be considered. Power, makespan, load level, responsiveness and system availability are to be applied as a multi objective optimization problem in order to reach an optimum. These techniques may be combined in an adaptive fashion and tested

against the varying conditions. By combining, the overhead should be maintained at a low rate whereas efficiency is to be maximized.

Acknowledgment

Gratefully acknowledge Vishveshvaraya PhD scheme for Electronics and IT, DeitY, Ministry of Communications and IT, Government of India's fellowship grantthrough Anna University, Chennai for their support throughout the working of this paper and for carrying out research in this domain.

References

[1] Lotfi A. Zadeh,"Fuzzy Logic, Neural Networks, and Soft Computing", Communications of the ACM, Vol 37, No 3, March 1993.

[2] Marei S. AI-Amri, Rana Ejaz Ahmed, "New Job Selection and Location Policies For Load Distributing Algorithms", Canadian Conference on Electrical and Computer Engineering, Vol 2,pp 1139-1144,2001.

[3] Marco Dorigo, Thomas Stutzle, Ant Colony Optimization, A Bradford book, MIT Press, 2004.

[4] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang, "Cloud Task scheduling based on Load Balancing Ant Colony Optimization",Sixth Annual ChinaGrid Conference, Pg 3-9, 2011.

[5] Z. Zhang, X. Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", 2nd International Conference on Industrial Mechatronics and Automation, 240-243, 2010.

[6] S.Banerjee, I.Mukherjee P.K.Mahanti, "Cloud Computing Initiative using Modified Ant Colony Framework",World Academy of Science, Engineering and Technology Vol:3 2009-08-27, 2009.

[7] L.Dinesh,Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied Soft Computing 13, 2292–2303, 2013.

[8] Kim et al., "Optimal Job Scheduling in Grid Computing Using Efficient Binary Artificial Bee Colony Optimization", Soft Computing, Volume 17, Issue 5, pp 867-882, 2013.

[9] M.Rathore, S.Rai, N. Saluja, "Load Balancing of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud", International Journal of Computer Science and Information Technologies, Vol. 6 (4), 4128-4132, 2015.

[10] J. Page, J.Naughton, "Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing", 19th IEEE International Parallel and Distributed Processing Symposium, 1530-2075/05, 2005.

[11] Dasgupta et al., "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing",International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), 2013.

[12] J.Gu, J.Hu, T. Zhao, G. Sun, "A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment", Journal Of Computers, Vol. 7, No. 1, Pg 42-52, January 2012.

[13] Tingting et al., "Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing", 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, 978-1-4799-5079-9/14, 2014.

[14] N. Krishnanand, D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions", Swarm Intelligence, Vol 3, Pg 87-124, 2009.

[15] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", IEEE, 0-7803-2768-3/95/$4.00, Pg 1942-1948, 1995.

[16] Liu, Wang, "A PSO-Based Algorithm for Load Balancing in Virtual Machines of Cloud Computing Environment",Advances in Swarm Intelligence, Third International Conference Part 1, pp 142-147, 2012.

[17] B.Mondal,K. Dasgupta, P. Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach", Procedia Technology 4, pp 783 – 789, C3IT-2012.

[18] A. Florence, V. Shanthi, "A load balancing model using Firefly algorithm in cloud computing", Journal of Computer Science 10 (7): 1156-1165, 2014.

[19] Kuo-Qin et al., "Towards a Load Balancing in a Three-level Cloud Computing Network", 978-1-4244-5540-9/10/$26.00 ©2010 IEEE.

[20] Naghibzadeh, M., "A min-min max-min selec- tive algorithm for grid task scheduling". 1-42440- 1007-X/07/$25.00, 2007 IEEE. Dept. of Computer Engineering Ferdowsi University of Mashad, 2007.

[21] Sridevi. S, Chitra Devi D, V. Rhymend Uthariaraj, "Efficient Load Balancing and Dynamic Resource Allocation in Cloud Environment", International Journal of Engineering Research & Technology (IJERT), Vol. 4 Issue 02, February-2015.

[22] Frank Neumann, Dirk Sudholt, Carsten Witt, "Computational Complexity of Ant Colony Optimization and Its Hybridization", Aspects of Computer Science, (STACS '05), Springer, LNCS, vol 3404, pp 44–56, 2005.