International Journal of Engineering & Technology, 7 (4.31) (2018) 386-390

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Performance Evaluation of Load Balancing Algorithm for Virtual Machine in Data Centre in Cloud Computing

Yuganes A/P Parmesivan[1], Sazlinah Hasan[2]*, Abdullah Muhammed[3]

[1]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia.
**Corresponding author E-mail: sazlinah@upm.edu.my*

Abstract

Cloud computing has become biggest buzz in the computer era these days. It runs entire operating systems on the cloud and do everything on cloud to store data off-site. Cloud computing is primarily based on grid computing, but it's a new computational model. Cloud computing has emerged into a new opportunity to further enhance way of hosting data centre and provide services. The primary substance of cloud computing is to deal the computing power, storage, different sort of stages and services which assigned to the external users on demand through the internet. Task scheduling in cloud computing is vital role optimisation and effective dynamic resource allocation for load balancing. In cloud, the issue focused is under utilisation and over utilisation of the resources to distribute workload of multiple network links for example, when cloud clients try to access and send request to the same cloud server while the other cloud server remain idle at that moment, leads to the unbalanced of workload on cloud data centers. Thus, load balancing is to assign tasks to the individual cloud data centers of the shared system so that no single cloud data centers is overloaded or under loaded. A Hybrid approach of Honey Bee (HB) and Particle Swarm Optimisation (PSO) load balancing algorithm is combined in order to get effective response time. The proposed hybrid algorithm has been experimented by using CloudSim simulator. The result shows that the hybrid load balancing algorithm improves the cloud system performance by reducing the response time compared to the Honey Bee (HB) and Particle Swarm Optimisation (PSO) load balancing algorithm.

Keywords: *Cloud computing; Honey Bee (HB), Particle Swarm Optimization (PSO); Hybrid, Response time*

1. Introduction

Cloud computing is moving towards new era, has emerged into a new opportunity to further enhance way of hosting data center and provide services. In the world of cloud and data mobility, Cloud computing is a paradigm shift from grid computing plays vital role in mold upcoming data centers environment. Increasing of ability to manage resources in much more efficient way with the extensive use of cloud. Thus, the cloud computing model is expected to become the next-generation computing infrastructure owing to its performance efficiency. In cloud computing, virtualisation is one of the essential tools for example hardware resources as virtual machines (VM). Virtualisation in cloud computing gives the illusion of real thing but it is not real environment as it provides all the features of the real thing. For instance, a web server has the dispatcher to balance the incoming request to the servers.

The main goal of dispatcher in load balancing is able to send the request to the server that is active and under loaded at the time. The server and the front end will exchange the information on the load with each other to make each other effective decisions with the help of load balancing algorithms. Distribution of workload across nodes and available resources is one of the load balancing method. The main aim is to achieve efficient resource utilisation, reduce response time, increase in throughput and minimise the overhead in order to fairly distribute the workload. Load balancing in data center is achieved by the relevant hardware or software, for example multilayer switch, such as F5. Thus, in cloud system various load balancing algorithms to cope up with load and to serve the efficient load balancing in order to solve load balancing issue of cloud computing as there are situation in which some nodes are extremely busy and some idle.

Practically, this increases the overall response time with the available load balancing mechanism that ensure that the workload across the nodes is uniformly distributed. Architecture of the system is also one of the factor that affects performance of load balancing. There are few architectures of load balancing algorithm like centralised, decentralised, and hierarchical load balancer. In this context, the overhead in the distributed load balancing is less as compared to the centralised load balancing.

Therefore, we have use Honey Bee foraging algorithm, one of the examples of the distributed load balancing [1]. Ideally, many hybridisation of algorithms [2-8] help the performance of cloud computing that intergrate the two existing Load Balancing algorithms. This paper presents the new hybridisation of Honey Bee and Particle Swarm Optimisation.

2. Related Works

Cloud computing plays an important role in today's organisation workspace by providing preferred services based on the user requirements. Multiple algorithms have been developed and implemented on the cloud environment to enhance its performance and resources utilisation. Furthermore, Cloud computing is classified as an extensive scaled distributed model that relies on the financial range of the cloud services that works as abstract, virtual and dynamic. The main substance in cloud computing is to

manage the computing power, storage, multiple sort of stages and services that is assigned to the corporate users that demands through the internet. [3-4] [10].

Load balancing is categorised as one of the important optimisation process for task scheduling in cloud environment as it works mainly in utilising the dynamic resources allocation. Poor policy scheduling may result in overloading certain virtual machine and cause the other virtual machines to be in idle or under loaded state. Therefore, this load balancing algorithm works by re-assigning the task to the individual nodes of the shared systems and build resource utilisation efficiency to acquire better response time of the job. Randles et al. [11] researched a decentralized honey bee-based load balancing algorithm that is a nature inspired algorithm for self-association. It accomplishes global load balancing through local server actions. Honey bee works just like the behavior of bumble bees to find a sustainable food source likewise in cloud tasks is to find the fittest virtual machine(VM) for effective performance. The Swarm-based algorithm is a search algorithm capable of locating good solutions efficiently. It is related to nature's method to derive a search towards the optimal solution and uses a population of solutions for every iteration of single solution [12]. Particle Swarm optimisation (PSO) is classified as an optimisation procedure based on the social behavior of groups of organisations, for instance the flocking of birds. Individual solutions in a population as 'particles' change their positions according to time. Thus, each particle modifies its position in search space based on its own experience and neighbouring particle by recall the best position to be visited by itself and its neighbors. There are many research work has been carried out in resource allocation and task scheduling in order to improve efficiency in cloud computing. Thus, most of these researches improve the waiting time, make span, execution time, and resource utilisation. Mirjalili et al [23] introduced one of the algorithm for solving engineering problems Grey Wolf (GW). It is a new optimisation algorithm inspired by grey wolf hunting and their role. Abdullah & Othman [24], introduced simulated anneling for cloud environment targets on termination time among resources and load balancing.

2.1 Load Balancing in Cloud Computing

In cloud computing, load balancing is the strategy of balancing the heap or load on the cloud servers [12]. It is way of reassigning the entire load of the system to build resource utilisation efficient and get the better response time of the job run concurrently. For instance, working on all incoming requests while others servers remain free. This issue of over-burdening at a specific server may lead to the increase in reaction time and even the requests may get timeout. To diminish this inconvenience of the cloud clients' heap is adjusted equally among every available accessible server. By this plan we can't discover any state with the end goal that a server is excessively occupied while other server is free. In fact, there are numerous effective way of issues residing load balancing in cloud computing. Basically, two conditions on load balancing being considered, one of it is resource allocation and the other one is task scheduling in cloud environment. Brief explanation on efficient resource allocation and task scheduling will ensure [13]:

- Resources availability
- Resources are effectively used under several constraints of load or all situations.
- Threshold plays vital role as energy is reserved in case of low load, mainly when usage of cloud resources when it is below certain threshold
- Save energy in case of less load

Cloud computing environment can have either static or dynamic environment based on configurations demanded by the cloud provider.

- Static environment: In static environment, flexibility in cloud is not in good state. In this case, the cloud requires prior and basic knowledge of nodes capacity, CPU, processing power, memory, performance and statistics of user requirements before scheduling and they are statically assigned to available resources. They are easy to implement with less runtime overhead, but we need to ensure and pre-planned all the tasks or process before execution period. In conjunction, algorithms proposed to achieve load balancing in static environment is easier to simulate but is not at its best suited for heterogeneous cloud environment [14].
- Dynamic environment: It allows for a better management and well-organised of tasks execution, as tasks are scheduled instantly as they arrive in the system. The cloud provider installs heterogeneous resources for dynamic environment. In this case, cloud uses run-time statistics but not relying on prior knowledge. It is highly adaptable with cloud computing environment but it is not easy to simulate. Main goal of dynamic scheduling algorithm is the ability to adapt to changes and sustain without having any issues to the processes in the queue [14].

2. 2 Load Balancing Algorithm

Cloud is made up of enormous resources. Therefore, management of these resource requires efficient planning and proper structure with hand before construct an algorithm for resource provisioning on cloud server. Cloud task scheduling problem relates to scheduling of tasks and allows optimal allocation of resources among given tasks in a finite time to fulfil desire quality of service (QoS) in cloud. Loading balancing in cloud is a mechanism that distributes the excess dynamic local workload ideally balanced across all the nodes [2]. In order to achieve system overall performance and user satisfaction and higher resource utilisation to ensure that no single node is overloaded. According to the papers, there are different performance metrics to evaluate the performance of load balancing algorithms.

There are numbers of related work on load balancing and performance metrics adopted in our surveyed load balancing algorithms with their optimisation behaviour [4].

- Response time is the total time between a request for service and the completion of that request. Generally, total time taken to process the request.
- Time load variance and standard deviation of utilisation: These 2 metrics is mean utilisation are quite popular in some articles, as they are easy to be measured.
- Number of overloaded hosts: It is to evaluate load balancing effect; it calculates the number of hosts in cloud are overloaded. The value is dependent on the pre-configuration of overloaded threshold.
- Percent of all VMs located: It is basically about VM load balancing in multiple data centres and specifies the VM distribution percentage of different datacentre with a minimum and maximum percentages of all VMs that can be located at cloud.
- Throughput: Higher throughput comes along better system performance, as it measures how fast the hosts can handle with requests, as imbalance loads may reduce system performance.

2.1.1 Honey Bee (HB)

Bees algorithm basically formed and mimicked for several honey bee-based optimisation algorithms. In fact, naturally honey bees have several complicated behaviours such as mating, breeding, and foraging [15]. It accomplishes global load balancing through local server actions. Execution of the framework is upgraded with expanded framework decent variety yet throughput not expanded with an expansion in framework estimate. It is most appropriate for the conditions where the different populace of organisation forms is required.

This algorithm is about the behaviour of bumble bees for finding a sustainable food source. There is a class of bumble bees called the

forager bumble bees which rummage for sustenance sources, in the wake of finding one, they re-swing to the honey bee haven to broadcast this using a move called waggle move. The show of this move, gives the likelihood of the quality or measure of sustenance and moreover its expel from the collection of beehives. Scout bumble bees by then take after the foragers to the zone of sustenance and a short time later began to secure it. They by then return to the honey bee haven and finish a waggle move, which gives an idea of how much sustenance, is left and thusly realizes more abuse or surrender of the sustenance source.

The considered honey bee based load balancing framework uses a gathering of servers composed into virtual servers, each serving a virtual organisation line of demand. Each server setting up a demand from its line figures a reward, which is like the quality that the bumble bees show up in their waggle move. One measure of this preferred standpoint can be the measure of time that the CPU spends on the treatment of a demand. The move floor if there ought to be an occurrence of honey bees is nearly looking like an advert board here. This board is moreover used to promote the advantage of the entire state.

In load balancing activity [6], every server plays a specific honey bee part with probabilities. An executing server will finish the demand and compute the benefit of the just-overhauled virtual server. There are for the most part three sorts of honey bees present, i.e. spectator, honey bees, scouts and utilized honey bees in this algorithm. The utilized honey bees settle down to sustain resources and hold its surroundings in the memory; while spectators take this data from the utilized honey bees and pick the nourishment resources appropriately. Then again, the scouts are in charge of finding the new sustenance resources. The primary constituent of the collection of beehive is the moving territory where data is being shared among the honey bees. The whole correspondence between honey bees happen in the moving district. This data is identified with the area and nature of nourishment resources. This move is known as "the waggle move". As data in regards to all the ideal nourishment assets are accessible with the spectators, which exist on the move floor, it can pick the most gainful asset [16]. Given underneath are the essential strides of the algorithm of honey bee settlement advancement.

2.2.2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization(PSO) uses the social conduct of the gatherings of populace in nature, for example, creature groups or flying creature rushing, or tutoring of fish. PSO comprises of a populace called swarm and every individual from the swarm is known as a particle. The particles search through the global ideal with a set velocity. Since the particles alter and refresh the situation regarding itself and its neighbourhood, it has the ability to do both nearby and global search. PSO evaluates a problem by having a number of candidate solutions, which are known as particles, and moving these particles around in the search-space fitting to simple mathematical formulae over the particle's position and velocity [17].

Local best-known position and is also guided as to the best-known positions in the search-space, these are updated as better positions and then they are found by other particles. This is expected to move the best particle termed as swarm toward the best solutions [7]. In 1995, James Kennedy and Russell Eberhart introduced method for optimisation of continuous non-linear function. Authors presented their work details using concepts from different areas. Particle swarm optimization (PSO) for neural network using this method for optimisation has been in the center of attraction. PSO excels at global search which is a very common gradient based method for training connection weights of ANNs, PSO shows faster convergence [18]. The particle swarm optimiser was able to train the network so as to achieve 92 percent correct [8].

2.1.3 Ant Colony Optimisation (ACO)

ACO algorithms basically works like real and similar to ants in which they try to find shortest path between nest and food source. It is computational models inspired by the collective foraging behaviour of ants. There are several characteristics of ACO as unique as it is consecutive, population-based metaheuristic [21]. M.Dorigo proposed a scheme based on Ant Colony, which is utilized for different optimisation issues, where the ants of different settlements make solutions for an issue by sharing the quality data. ACO algorithm working is like the genuine ants in which they endeavor to locate a most limited way amongst home and sustenance source. In ACO, various artificial ants make solution for the optimization issue and offer quality data that is, pheromone fixation on the way crossed by ants. ACO takes the genuine subterranean insect conduct as a premise. In the cloud context, pheromone is used to move forward and backward as from the underloaded node to overloaded node at the same time vice versa for shifting the load. Mousavi & Fazekas [22], researched ACO reduce the longest termination time among resources and reduce the workflow time in order to balance the load.

3. Methodology

This research is focused on how resource alloction can be utilized efficiently for the purpose of load balanced in a specific environment. Cloudsim provides provisioning capabilities for tasks execution (cloudlets) that run on VMs that run on servers that form datacentre.There are three (3) phases involved in order to do this research project, which are:

1. This phase would require extensive amount of time and experiments as it involves simulating the existing work. Identification of the performance of the algorithm such as waiting time, response time and virtual machine and host specifications that needs to be used during the simulation and processes in the current Load Balancing Honey Bee and Particle Swarm Optimisation. Algorithm.
2. Simulate the Honey Bee, Particle Swarm Optimisation and proposed hybrid algorithm with same set of behaviours and attributes.
3. Determine the outcome of the experiment and compare results' of HB, PSO and proposed hybrid algorithm in terms of response time.

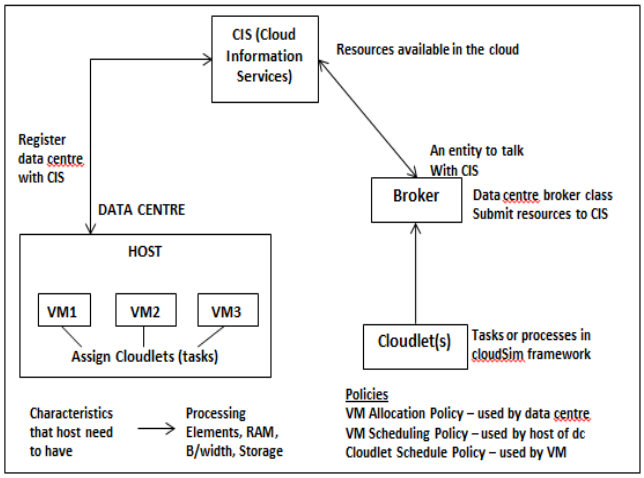

Fig. 1: Cloudsim work with load balancing algorithm [20]

The experiment performed in this study was carried out on CloudSim simulator. CloudSim is a simulation toolkit that enables for modelling and simulation of cloud computing system. Cloudsim supports the Clouds components such as data centers, virtual machines (VMs), load balancing algorithms and resource provisioning policies. On the other hand, the Cloudsim describes Cloud-based data center environment framework such as dedicated management interfaces for VMs, memory, storage, and

bandwidth provide simulated virtualisation as shown at **Fig. 1**. It exposes the configuration of related functionalities for hosts, applications especially number of tasks and their requirements, VMs, number of clients and their applications types, and broker scheduling policies. Simulations of cloud, the core framework administrations are displayed by a Datacenter part to handle benefit demands [20]. In this paper, Cloudsim is used to simulate task scheduling in order to minimise response time and maximise resource utilisation a hybrid approach of Honey Bee and Particle Swarm Optimisation algorithm is implemented inside of Cloudsim to enhance the work of the already implemented simple broker. This broker class model a representative, which is in charge interceding amongst clients and requirements upon clients QoS necessities and allocate resource efficiently. The broker following up for the benefit of clients recognizes reasonable Cloud service providers through the Cloud Information Service (CIS) and consults with them for an allocation of resources that addresses QoS issues of clients. The researcher and framework designers must broaden this class for leading examinations with their uniquely created application allocation approaches. Refer to proposed Hybrid Algorithm section on how the hybridisation of algorithm works to solve the load balancing issue.

3.2 The Proposed Hybrid Algorithm

The proposed algorithm is completely inspired by natural foraging behaviours of honey bees. Basically, the allocated task will update the remaining tasks about the VM status in a way similar to the bees finding the food sources updating the other bees in the bee hive through waggle dance. In this context, this task updates the status of the VM availability and the load of the VMs. In this model, we optimise the processing time of both task running time and system resource utilisation. A simple mutation mechanism and inertia weight method by classifying the fitness value. Then, global search is performed. Algorithm developed based on Honey Bee and Particle Swarm Optimisation algorithm to find a proper resource allocation to jobs.

The hybrid results will compare to PSO and Honey Bee accordingly. Load on resources are also balanced by the proposed algorithm. Initially, load balancing algorithm. Honey bee portrays system of load balancing among over-stacked virtual machines and under stacked virtual machines. The algorithm enlivened by Honey Bee conduct to balance the load in cloud computing. The center work is to present the assignments to the virtual machine till it gets over-burden. In this situation, stack on virtual machine turn out to be more than threshold which is set 90. By random search method, we send the remaining task to under loaded virtual machine and will never send tasks to overloaded virtual machine. Hybridisation work is combined with Honey bee-based behaviour and Particle Swarm Optimization (PSO) to improve the efficiency and help to reduce overhead, migration time, and enhance the performance with response time. The input parameters for the bee's swarm optimization are initialized

The approach of this proposed algorithm is focus on load of virtual machine where we have to identify which node has high workload, which node has low load and which has no task. The following are steps of the existing LB algorithm HB and PSO :

Honey Bee algorithm [18]

1. Start
2. Determine which Vm is loaded or underloaded.
3. Identify the under loaded VMs and overloaded VMs
4. Sort it
5. Sort the tasks in overloaded VMs based on priority
6. For each task in each overloaded VM find a suitable under loaded VM.
7. Update the overloaded and under loaded VM sets and go to step 2.
8. Stop

PSO algorithm [17]

1. Start
2. Initialize the position and velocity of n particles randomly.
3. For each particle calculate fitness and put it as pbestj.
3.1 Put the best pbestj as the gbest.
3.2 While (if not meet requirement)
3.2.1 Update the inertia weight W(t)
3.2.2 Update the velocity and position for every particle
3.2.3 Calculate its fitness.
3.2.4 IF position fitness is better than pbest
Update pbestj
3.3 End while
4. End while
5. Return gbest

Hybrid algorithm

1. Start
2. Determine which Vm is loaded or underloaded.
3. Identify the under loaded VMs and overloaded VMs
   //Reschedule cloudlets with PSO algo
4. Initialize the position and velocity of n particles
5. For each particle, calculate its fitness value
6. If the fitness value is better than the previous pbest
7. Set the current fitness value as new pbest
8. Repeat steps 5 & 6 for all particles
9. Sort the best particles as gbest
10. For all particles, calculate velocity and update their positions
11. End if max iteration
12. Else Repeat step 5 & 6
13. End

4. Simulation Configuration

CloudSim has been used to apply code of the HB, PSO and Hybrid algorithm. Parameter values for the three (3) algorithm are setting based on [18] as ahown in Table 1.

Table 1: Simulation Parameters (HB, PSO and HYBRID)

SIMULATION PARAMETERS	VALUE
Task (Cloudlet)	
Length of task (executable instruction length in bytes)	1000-20000
Total number of tasks	100-500
VM	
Number of VMs	50
Processor speed	500-2000MIPS
Available memory space in a single VM	256-2048 Mb
Bandwidth	
Cloudlet Scheduler	500-1000
Number of Processor Elements (PEs) requirement	Time Shared 1-4

5. Simulation Results and Discussions

Table 2 shows the results of average response time based on scenario for 50 virtual machine with different number of tasks tested using the hybrid algorithm, Honey Bee(HB) and Particle Swarm Optimization (PSO) algorithm.

Table 2: Average Response Time Results

LB Algorithm / Tasks	Average Response Time (ms)		
	Hybrid	PSO	Honey Bee
100	267.66	267.66	273.3
200	259.47	260.91	267.33
300	258.74	260.18	266.63
400	251.22	252.65	259.16
500	254.81	256.24	262.73

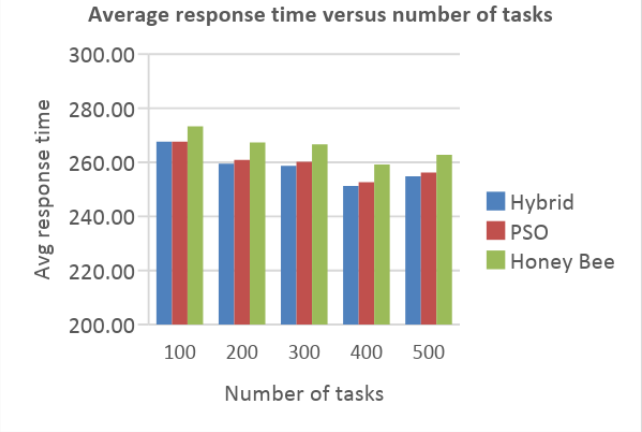

Fig. 2: Comparison Average Response Time of load balancing algorithm

Fig 2 shows that the hybrid algorithm of HB and PSO is more excellent and effective than the Honey Bee (HB) and Particle Swarm Optimization (PSO) with the lowest average response time because it is very simple and does not overhead computation. We defined 50 virtual machines in data centre and the cloudlets tested is twice the number of VM as this is part of enhancement where it can actually reschedule the cloudlets to be executed on specific VMs. PSO used to randomly update the inertia weight for better performance after the fitness value updated from first round of scheduling on VM fitness based. In this context, hybridisation work better than existing single population-based scheduling and no single node is responsible for entire scheduling decision. As the aim is to divide workload evenly across the virtual machines available to improve the response time with number of tasks proportionally.

6. Conclusion

This project introduced hybrid load balancing algorithm with the goal of dynamically re-allocating virtual machines when resource utilisation reaches high threshold size and also during utilisation is below minimal threshold size. The purpose of having virtual machines load balancing mechanism is to have the ability to consolidate virtual machines to allow resources to be utilised to a maximum capacity and to have the ability to reschedule the cloudlets and improve processing time.

The scheduler takes into account the utilisation of RAM, Bandwidth and CPU during interval processing time and chooses the highest parameter ratio. Instead of focusing on CPU load only, this technique gives the ability to determine which parameter contributed to high utilisation. With the following techniques in place, the proposed algorithm was able to achieve lesser load usage overall and indirectly proving that the performance analysis method is correct and can be a better alternative method.

Acknowledgement

This research was financially supported by the Universiti Putra Malaysia through PUTRA Grant with Project Number GP/2018/9621400.

References

[1] Al-Rayis, E., & Kurdi, H. (2013). Performance Analysis of load balancing Architectures in Cloud computing. In Modelling Symposium (EMS), European (pp. 520-524). IEEE.

[2] Issawi, S. F., Al Halees, A., & Radi, M. (2015). An Efficient Adaptive Load Balancing Algorithm for Cloud Computing Under Bursty Workloads. Engineering, Technology & Applied Science Research, 5(3), 795-800.

[3] Ayesta, U., Erausquin, M., Ferreira, E., & Jacko, P. (2016). Optimal dynamic resource allocation to prevent defaults. Operations Research Letters, 44(4), 451-456.

[4] Alamin, M.A., K.Elbashir, M., & A. Osman, A. (2017). A Load Balancing Algorithm to Enhance the Response Time in Cloud Computing. Red Sea University Journal of Basic and Applied Science, 473-490.

[5] You, L., Peng, J., Chen, M., & Qiu, M. (2017). A Strategy to Improve the Efficiency of I/O Intensive Application in Cloud Computing Environment. Journal of Signal Processing Systems, 86(2-3), 149-156.

[6] Xu, M., Tian, W., & Buyya, R. (2017). A survey on load balancing algorithms for virtual machines placement in cloud computing. Concurrency and Computation: Practice and Experience, 29(12).

[7] Singh, H., & Gangwar, R. C. (2014). Comparative Study of Load Balancing Algorithms in Cloud Environment. International Journal on Recent and Innovation Trends in Computing and Communication, Vol: 2, 3195- 3199.

[8] Nema, L., Sharma, A., & Jain, S. (2016). Load Balancing Algorithm in Cloud Computing: An Entensive Survey. International Journal of Engineering Science and Computing, 7463-7468.

[9] Rajeshkannan, R., & Aramudhan, M. (2016). Comparative study of Load Balancing Algorithms in cloud computing environment. Indian Journal of Science and Technology, 9(20).

[10] Thakur, P., & Mahajan, M. (2017). Different Scheduling Algorithm in Cloud Computing: A Survey. International Journal of Modern Computer Science (IJMCS), 44-50.

[11] Randles, M., Lamb, D., & Taleb-Bendiab, A. (2010). A comparative study into distributed load balancing algorithms for cloud computing. In Advanced Information Networking and Applications Workshops (WAINA), IEEE 24th International Conference on (pp. 551-556). IEEE.

[12] Dave, A., Patel, B., & Bhatt, G. (2016, October). Load balancing in cloud computing using optimization techniques: A study. In Communication and Electronics Systems (ICCES), International Conference on (pp. 1-6). IEEE.

[13] Rajoriya, S. (2014). Load Balancing Techniques in Cloud Computing: An Overview. International Journal of Science and Research, 3.

[14] Nema, L., Sharma, A., & Jain, S. (2016). Load balancing algorithms in cloud computing: An extensive survey. International Journal of Engineering Science and Computing, 6(6).

[15] Kaur, A., & Kaur, B. (2016). Load balancing in tasks using honey bee behavior algorithm in cloud computing. In Wireless Networks and Embedded Systems (WECON), 5th International Conference on (pp. 1-5). IEEE.

[16] Mousavi, S., Mosavi, A., Varkonyi-Koczy, A. R., & Fazekas, G. (2017). Dynamic resource allocation in cloud computing. Acta Polytechnica Hungarica, 14(4).

[17] Gudise, V. G., and Venayagamoorthy, G. K. (2003), Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks, in Swarm Intelligence Symposium, Proceedings of the IEEE, pp. 110-117.

[18] Hashem, W., Nashaat, H., & Rizk, R. (2017). Honey Bee Based Load Balancing in Cloud Computing. KSII Transactions on Internet and Information Systems (TIIS), 11(12), 5694-5711.

[19] Awad, A. I., El-Hefnawy, N. A., & Abdel_kader, H. M. (2015). Enhanced particle swarm optimization for task scheduling in cloud computing environments. Procedia Computer Science, 65, 920-929.

[20] Pakize, S. R., Khademi, S. M. & Gandomi , A. (2014). Comparison of CloudSim, CloudAnalyst And CloudReports Simulator in Cloud Computing. International journal of Computer Science & Network Solutions, 2(5), 19-27.

[21] Ramos, V., Muge, F., & Pina, P. (2002). Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. *HIS*, *87*, 500-512.

[22] Mousavi, S. M., & Gábor, F. (2016). A novel algorithm for Load Balancing using HBA and ACO in Cloud Computing environment. International Journal of Computer Science and Information Security, 14(6), 48.

[23] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

[24] Abdullah, M., & Othman, M. (2014). Simulated annealing approach to cost-based multi-quality of service job scheduling in cloud computing enviroment. American Journal of Applied Sciences, 11(6), 872.