An Enhanced Method For Load Balancing Algorithm Based on Honey Bee

Nishant Awasthi[1], Anshul Khurana[2]
[1]Dept of CSE
[2]Asstt. Prof, Dept of CSE
[1, 2] Shri Ram Institute of Technology, Jabalpur, Madhya Pradesh, India.

Abstract- *In a cloud environment, the aim of using optimal resources can be achieved using a load balancing technique. The load-balancing technique assigns a set of requests into a set of resources for distributing the load. It is one of the significant issues in cloud computing and known as an NP-hard problem. Therefore, many nature-inspired meta-heuristic techniques are proposed to provide high efficiency. However, despite the importance of the nature-inspired meta-heuristic techniques for solving the problem of the load balancing in the cloud environment, there is not a complete and detailed paper about reviewing and studying the main important issues in this domain. Therefore, this paper presents comprehensive coverage of the nature-inspired meta-heuristic techniques applied in the area of the cloud load-balancing.*

The main goal of this paper is to highlight the emphasis on load balancing algorithms and the benefits that they provide to overcome the cloud load-balancing challenges. In addition, to solve the load-balancing problem in the cloud environments, the advantages of the Honey Bee algorithms have been proposed and their significant challenges are considered for proposing the techniques that are more effective in the cloud computing environment.

Keywords- Load balancing, Cloud computing, Nature Inspired, Honey Bee, Response Time, Cloud Analyst.

I. INTRODUCTION

Ss Aim of scheduling is to find the best cloud resources for upcoming end users' applications (tasks) to improve the key performance parameters (QoS parameters) and improve the resource utilization ratio [1]. There are various performance indicator parameters like response time, execution cost, makespan time, reliability, energy consumption etc. in cloud computing. We must analyze and improve them using efficient resource scheduling algorithm to fulfill the requirement of end users as well as service provider without affecting the service level agreement (SLA) violation. Resource scheduling becomes a prominent issue in the field of cloud computing due to heterogeneity, dynamism and dispersion of resources that are not resolved by existing scheduling algorithms. Therefore, we need a scheduling algorithm that distribute the heterogeneous workload among the cloud resources (VMs) based upon the capacity of the resources and overcomes the problem of overload and underloaded.

1.3 Working of Load Balancer in Cloud Server

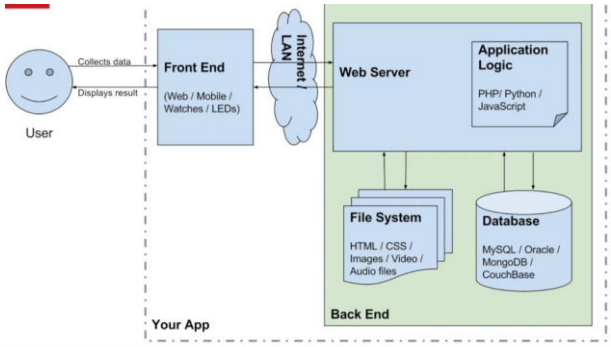

Figure 1.2: Cloud server with One Web server.

The above diagram is a single server setup where everything is residing on single server. The database, file system and one web server is running on single server. The server is having some limited ram (Say 1 GB RAMS), limited processor (Say 2.5 GHz).Now after sometimes the number of request gets increasing rapidly. After some time, there will be a scalability issues due to limitations of the Server Configuration. And the problem of Load Balancing will arise in the existing configuration. To overcome the above issues, the load from one server, has to be transferred to multiple server. The multiple server architecture is shown below in figure. This is the example how load balancer works in the cloud server environment. The user request is sent to the load balancer through front end. The load balancer helps in balancing the load. The same application is running on three web-servers, instead of one web-server in previous module.

All the servers are having individual RAM and Processing Elements. The load balancer is responsible for sending the user request to proper web server.

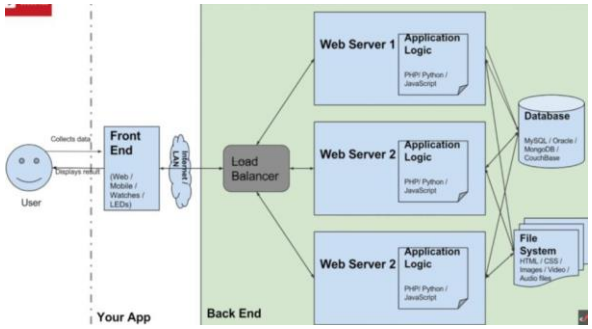

Figure 1.3: Load Balancer in Cloud server.

II. RELATED WORK

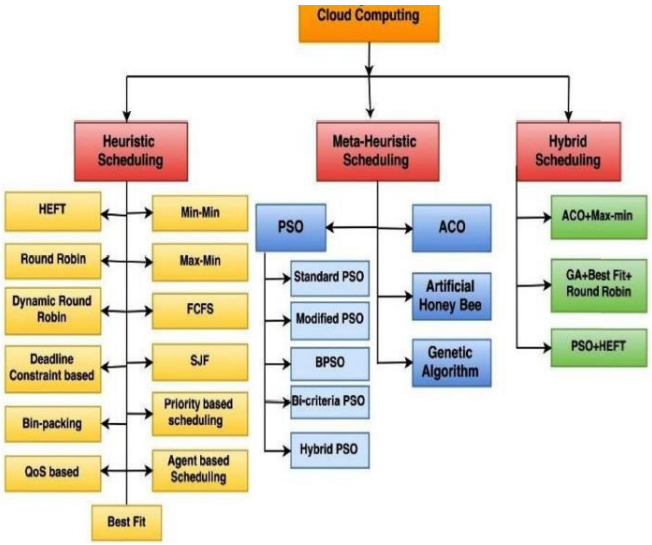

Figure 2.1: Classification of Load Balancing algorithms.

The load balancing algorithm in cloud computing has been categorized based on different parameters like heuristic based, Meta-heuristics based and hybrid scheduling based.

2.1 FCFS, SJF and RR algorithm: First, arrival request of tasks is fulfilled in first come first serve (FCFS) [2] algorithm. Mishra used the round robin and weighted round robin techniques [3] to solve the load balancing and improve the response time and makespan [4]. Devi et al. developed a resource allocation strategy to improve the time, overhead and other QoS parameters based on the concept of the enhanced weighted round robin (WRR) scheduling algorithm [5], but the algorithm failed to balance the load at cloud datacenter and unable to utilize the resource efficiently.

Deadline based scheduling: Deadline constraint has become an important factor in field of scheduling, it affects the SLA (service level agreement) if large number of application unable to meet deadline. E. Coninck et al. [6], proposed deadline based scheduling algorithm that dynamically provision the resource for scalability as per demand and improve the QoS parameters such as task meet with deadline, time etc. A new workflow scheduling algorithm is proposed [7] for SaaS using the concept of Partial Critical Paths (PCP) to improve the QoS parameters while satisfying the deadline constraint. One phase and two phase scheduling algorithm has been proposed for IaaS to overcome the above mentioned limitation [8].

However, some more deadline based scheduling algorithms have been proposed using the different novel approach such as Aneka cloud platform [9] that efficiently provision the resources (obtained from private/public cloud) to scientific applications and improve QoS parameters, deadline and budget constraint based algorithm to fulfill the user's demand. To overcome the limitation of backfilling algorithm, Analytic Hierarchy Process (AHP) work as a decision maker that select optimum lease from best effort queue and helpful for backfilling algorithm in scheduling of lease considering Haizea as lease manager. Further, AHP evaluate the ranks of similar lease and schedule more lease, hence reduce the wastage of slots and improve the performance of algorithm. Three scheduling algorithms have been proposed by K. [10] to solve the problem of workflow scheduling considering multiple deadline constraint and improve makespan time as well as execution time. Efficient task scheduling algorithm is proposed by A. Visheratin, M. Melnik to satisfy the user QoS considering hard deadline as constraint [11]. Adaptive two stage deadline based scheduling algorithm has been proposed by R. Khorsand [12]. First stage dynamically fragments the workflow while second stage allocates the workflow fragments to VMs based upon the capacity of resource considering deadline as constraint.

Adaptive task allocation technique has been proposed by L. Wang and E. Gelenbe to improve online QoS parameters. Proposed algorithm uses three schemes: reinforcement learning technique is used for measurement; “sensible” allocation scheme is used to reduce the response time for allocating the tasks to sub-systems in efficient way, third scheme splits the task arrival stream into sub-streams at rates computed from the hosts’ processing capabilities. Apart from above mentioned technique, some other scheduling algorithms such as instance and value (IVH) algorithm that avoid over-provisioning problem and reduce infrastructure cost. A new VM placement algorithm has been proposed by S. Sotiriadis et al. [13] that are based upon past VMs usage experience using machine learning approach. Further proposed algorithm allow self managed VM placement strategy (best mapping between physical host and deployed virtual machines) and improve QoS parameters at OpenStack cloud environment. A new algorithm (Hybrid Cloud Optimized Cost scheduling) based upon elasticity concept has been proposed

by L. Bittencourt and E. Madeira that determines which resources should be leased from the public cloud so that private cloud can execute their scientific workflow application with in time and improve execution cost . Another scheduling algorithm has been proposed by A. AbouElseoud et al. named as Online Potential Finish Time (OPFT) to optimize execution cost and time in cloud computing. OPFT algorithm inspired the concept of flow of electric current, as more tasks are allocated to powerful virtual machines that have the capability to process tasks in minimum time delay. Proposed algorithm is simulated at CloudSim tool and performs better than other baseline algorithms like round robin, FCFS, min-min [14] and MCT algorithm [15].

III. PROPOSED SYSTEM

Dynamic algorithms provide better results in heterogeneous environment. These algorithms are more stretchable. Dynamic algorithms can take in charge of the dynamic changes to the attributes. Main advantage of this algorithm is that, the selection of task is based on present state and this will help to improve the performance of the system. Proposed system performs dynamic load balancing using little modification in honey bee algorithm.

Cloud user base generates requests in the form of cloudlet and then cloudlets are received by data center controller. Load balancer component uses load balancing algorithm for balancing load among various virtual machines by instructing virtual machine manager. System architecture uses nature inspired honey bee based algorithm for efficient load balancing of resources to maximize utilization of resources.

3.1 Proposed Method

Honey Bee algorithm is based on the foraging behavior of honey bees. The artificial bee colony contains three groups of bees: scout bees, Forager bees and onlooker bees [14]. Scout bees are sent for search of suitable food source, when they found the source, they return to the hive and advertise it in the form of “Waggle Dance”. The suitability of the food source and its distance from the hive is communicated through the waggle dance display. Now, the forager bees follow the scout bees back to the discovered food source and begin to harvest it. After collecting the food they return to the hive and the remaining quality of the food available at the source is shown in the form of Waggle dance to decide that the remaining bees should sent to the same source or to search the new suitable source of food. Table below represents Honey bee system with cloud environment:

Table3.1 : Honey bee and cloud environment parameters.

Honey Bee Hive	Cloud Environment
Honey bee	Task (Cloudlet)
Food source	Virtual Machine
Foraging a food source	Loading of a task to VM
Reduction at food source	Overloaded VM
Finding new food source	Removed task scheduling to under loaded VM

At present, load is send to a randomly selected under-loaded VM after calculating threshold value. By randomly selecting under loaded VM we will not be able to determine the VM having the high performance. First we will be calculating the throughput and based on the highest throughput value we will be selecting that particular VM.

3.3 Proposed Algorithm

We will be calculating throughput at the time of creating the VM.

Step 1: Start
Step 2: Set number of tasks
Step 3: Set number of VM
Step 4: Calculate throughput.
Step 5: Initially, set the load on each VM to null
Step 6: Send first task to the VM having high throughput.
Step 7: Check current virtual machine load.
If current VM load > threshold value.
Step 8: If yes, select the VM having next high throughput.
Step 9: If no, send the task to the current VM
Step 10: If all task are send to the VM go to step 12
Step 11: Else go to step 6
Step 12: Stop.

The Throughput is the measure of performance of the task by a computing device over a specific time. It is a measure of amount of completed task against time consumed. It is used to measure the performance of the processor or VMs. It is calculated in terms of task executed per second of time unit.

To overcome the existing challenges of static load balancing, dynamic threshold based load balancing algorithm is being proposed and implemented. Every time when the application or node has been increased or decreased, the value of threshold will be calculated. This is easy to implement and use. The proposed load balancing algorithm will work at machine level, where the load will be calculated for each VM. The threshold value is the total current load of the VMs. The total value of threshold will be calculated by following formula:

Total_Load = Current_Load + (CurrentFile_Size).

The value of current load on any VM will be calculated by using the following formula mentioned below:

Current_Load = [α (Total_CPU_Usage) +β (Memory_Usage) +γ (Bandwith_Usage)].

IV. RESULT

4.1 Response Time Evaluation for Cloud Users

We have used different size virtual machines for this calculation. Table 5.1 below shows the performance comparison of the proposed algorithm along with Existing algorithm and one of the most common and well known Round Robin algorithms. The evaluations are done on the basis of makespan of each virtual machines using famous cloud simulator called Cloud Analyst. For evaluations, we have used 5 cloud service providers and 50 cloud users. On other set we have taken 3 Cloud service provider and 20 cloud users. Results show that the performance of proposed algorithm is better.

Table 5.1: Overall Response Time Comparison of load balancing Algorithms.

Number of CU and CSP	RR	Existing	Proposed
50/5	91.08	91.58	90.62
20/3	149.30	150.35	147.92

CASE I: Result Chart for 50 Cloud users and five Cloud Service providers.

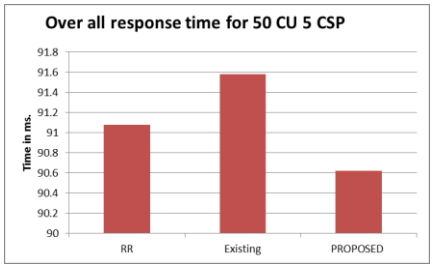

Figure 5.9: Graph showing result of Proposed Algorithm.

CASE II: Result Chart for 20 Cloud users and 3 Cloud Service providers.

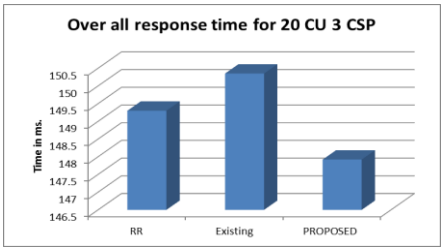

Figure 5.10: Graph showing result of Proposed Algorithm.

Results of the above calculation shows that the proposed method completes the process with lower response time as compared to existing Load Balancing algorithms. The Performance of the proposed model is better than existing ones in terms of makespan. Results shows that proposed algorithm behaves better in terms of response time after testing it with Cloud Analyst Simulator on different parameters.\psychological problem. Untreated depression increases the chance of dangerous behaviors. The significant challenge of detecting depression is the recognition that depressive symptoms may differ from patients' behavior and personality [1]. For clinic depression, doctors may evaluate the patient via the depression test taken by patients. Apparently, these clinical records are restricted due to many factors, such as age, sex; moreover, they are private and expensive. To overcome such limitations of clinical data, it would be beneficial to use text mining tools to extract and analyze depression symptoms from social media, such as Twitter. Social media generates countless data every day because of millions of active users share and communicate in entire community, it changes human interaction [2,3]. Other than the traditional data, such as literatures, social media data is richer and more accessible [4]. However, investigating this new fast-growth of data requires advanced development tool to discover useful insights. These advanced technologies include Natural Language Processing (NLP) [5], data mining, machine learning, social media analysis and so on.

In our research, the goal is to extract and summarize the uncommon but potentially helpful factors that depressive symptom performed from the social media data. Finally, the extracted depression symptoms w

IV. CONCLUSION

This paper focuses on the problem of load balancing in cloud computing environment. In this thesis we have proposed a method for proper balancing of loads in cloud computing environments. The cloud consists of data centres that have more number of Virtual machines and servers connected to each other for serving the user requests. The proposed algorithms are based upon optimization method and

it reduces the makespan significantly. Proposed algorithm is also compared with some existing load balancing algorithms and it is observed that both response time and processing time are improved in the proposed strategy.

REFERENCES

[1] B. Singh, S., Chana, I., 2016. Cloud resource provisioning: survey, status and future research directions. Knowl. Inf. Syst. 49 (3), 1005–1069.

[2] Li, W., Shi, H., Dec. 2009. Dynamic Load Balancing Algorithm Based on FCFS. In: 4th International Conference on Innovative Computing, Information and Control (ICICIC). Kaohsiung, Taiwan.

[3] Mondal, R.K., Nandi, E., Sarddar, D., Oct. 2015. Load balancing scheduling with shortest load first. International Journal of Grid Distribution Computing 8 (4), 171–178.

[4] Samal, P., Mishra, P., 2013. Analysis of variants in round robin algorithms for load balancing in cloud computing. Int. J. Comput. Sci. Inf. Technol. 4 (3), 416–419.

[5] Devi, D.C., Uthariaraj, V.R., 2016. Load balancing in cloud computing environment using improved weighted round robin algorithm for non-preemptive dependent tasks. Sci. World J.

[6] Coninck, E.D., et al., 2016. Dynamic auto-scaling and scheduling of deadline constrained service workloads on IaaS cloud. J. Syst. Softw. 118, 101–114.

[7] Kumar, M., Sharma, S.C., Nov. 2017. "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment" in. journal of Computers & Electrical Engineering (CAEE) 69, 395–411.

[8] Abrishami, S., Naghibzadeh, M., 2012. Deadline-constrained workflow scheduling in software as a service cloud. Sci. Iran. 19 (3), 680–689.

[9] Vecchiola, C., et al., 2012. Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka. Future Gener. Comput. Syst. 28 (1), 58–65.

[10] Bochenina, K., 2014. A comparative study of scheduling algorithms for the multiple deadline-constrained workflows in heterogeneous computing systems with time windows. Procedia computer science 29, 509–522.

[11] Visheratin, A., et al., 2016. Workflow scheduling algorithms for hard-deadline constrained cloud environments. Procedia Computer Science 80, 2098–2106.

[12] Khorsand, R., et al., 2017. ATSDS: adaptive two-stage deadline-constrained workflow scheduling considering run-time circumstances in cloud computing environments. J. Supercomput. 73 (6), 2430–2455.

[13] Babu, D., Krishna, P., 2013. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl. Soft Comput. 13 (5), 2292–2303.

[14] Rastkhadiv, F., Zamanifar, K., 2016. Task scheduling based on load balancing using artificial bee colony in cloud computing environment. Int. J. Adv. Biotechnol. Res. 7 (5), 1058–1069.

[15] Jena, R.K., 2017. Task scheduling in cloud environment: a multi-objective ABC framework. J. Inf. Optim. Sci. 38 (1), 1–19.