ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

Analysis of hybrid load balancing strategy used in cloud computing based on Honey Bee and Min Max

Prof.HomeraDurani[1], Dr. Nirav Bhatt[2]

1 Research Scholar & Assistant Professor, MCA Department, RK University, Rajkot, Gujarat, India;

2 Associate Professor, MCA Department, RK University, Rajkot, Gujarat, India

Abstract - From many years, vast research is still going on load balancing in cloud computing due to tremendous storage of data in the server. But still there is the issue of load balancing in cloud computing. In this research paper Hybrid HoneyBee and MinMax algorithm has been proposed to perform better make span load balancing in cloud computing. The main goal is to distribute load in such a way to avoid underutilization and over utilization in the storage management. The proposed algorithm is prepared where all the data is evaluated in VM (Virtual Machine) and comparison is made whether it is high or low. If Load is found high, it is applied to the Modified HoneyBee algorithm. If Load is found, the Less Modified MinMax Algorithm is applied. While applying the above technique the make span results are found much better than other research work done previously.

IndexTerms: Cloud Computing, Load Balancing, HoneyBee, MinMax, Virtual Machine.

1. INTRODUCTION

Cloud computing has been much popular in the market after pandemic situation. Now days due to sudden increase in flow of data Load Balancing is in high demand [1]. Load Balancing plays a vital role in managing and distribute data simultaneously. But the crucial portion is when there is rise of data increasing load balancing does not provide better performance and services [2]. Many researchers have done tremendous work on better performance and services in load balancing. But still result varies in cloud environment when flow of data is high. So still better algorithms are needed in load balancing for task scheduling, performance and make span.

1.1 LOAD BALANCING

The main objective of load balancing is to make sure that no a single node is overloaded. Nor any node should be kept idle [3]. Load balancing make node in balanced manner and keep a track neither any hub is overloaded or under loaded. This is major issue now days when flow of data is increasing in heterogeneous environment of cloud due to tremendous boom in market for environment computing. Key role of load balancing is to prevent server or network by being overloaded or under loaded [4][5].

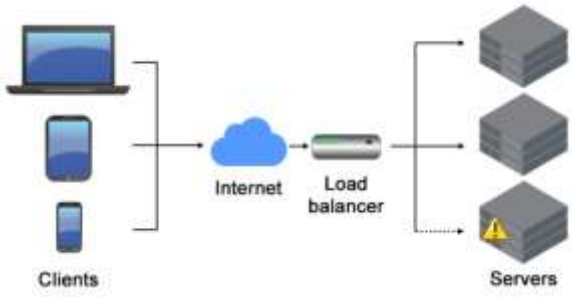

Figure 1 Load balancing

Load Balancing is mainly divided by two parts.

1. Static
2. Dynamic

Static: In Static load balancing working regards to previous data information. Data has been transferred according to previous status rather than considering current state. Further once the data is allocated it cannot transfer or reallocates. There cannot be any modification once allocation has been done [7]. Examples of Static load balancing are Min-Max, Max-Min and Round Robin etc.

Dynamic**:** In Dynamic load balancing it based on three main criteria circulated, non-circulated and semi-circulated approaches [6]. In Circulated methods all nodes are dispatched equally. In non-circulated method there is one hub in between which gathers all requests and where all nodes are divided. And here each cluster is main hub which then distributes data then transfer data to all servers [8]. In semi-circulate method there are group of cluster. Example of Dynamic load balancing is join Idle Queue, Honey Bee, and Ant Colony etc.

Hybrid: Hybrid load balancing is new approaching algorithm which is use to recover drawback of static and dynamic methods [12]. This algorithm is developed with combination of either two static methods and dynamic or wise versa. Even this algorithm is deriving which combination of static and dynamic method. This technique has results to more improve result in load balancing cloud computing environment.

2. RELATED WORK

Two Phase Scheduling Load Balancing Algorithm was proposed by Wang et al., Algorithm was combination of load balancing opportunistic and Min - Min load balancing algorithm [9]. This algorithm helps in keeping track of each node in work station to achieve the goal of load balancing. Another was MM scheduling algorithm which is use to minimize execution time for every task and complete work in particular time. These above algorithms were keen in resource utilization and efficiency of work.

Hybrid job scheduling algorithm using fuzzy theory and genetic algorithm was proposed by Javanmardi et al. The algorithm was proposed to allocate task to resources rather than checking their job length or capacity [10]. This algorithm was on basic of first come approach. Which was later modified and converted to fuzzy theory and achieved its goal. Here two parameters were being defined which leads to performance improved to 45% and time of execution reached to 50%. The implementation of this algorithm was performed on cloud sim and hybrid approach was outcome.

MOGA – Multi – Objective genetic algorithm based on scheduling was proposed by Liu, Luo, Zhang, Zhang and Li. This algorithm is combination of greedy and random methods. In this algorithm calculation is done on the basis of energy consumptions and service provider's profit [11]. Best result is generated when it is stored in node and after that selection is done on two parts: selection and gathering. In this scenario mutual alteration is done on individuals to get best energy consumptions and profit on service provider.

Dynamic Round Robin and First –Fit hybrid algorithm using energy efficient virtual machine in cloud system was proposed by Lin, Liu and Wu. This algorithm is mixture of Round Robin and FF in form of Hybrid [13]. This algorithm works by considering criteria probability distribution using normal scenario. Time constrain of each virtual machines are considered first. Then cost of each individual is considered of virtual machine. The algorithm first fit was used when there was heavy rush on machines and dynamic RR was implemented for energy consumption during that time duration when data is transferred.

3. PROPOSED WORK

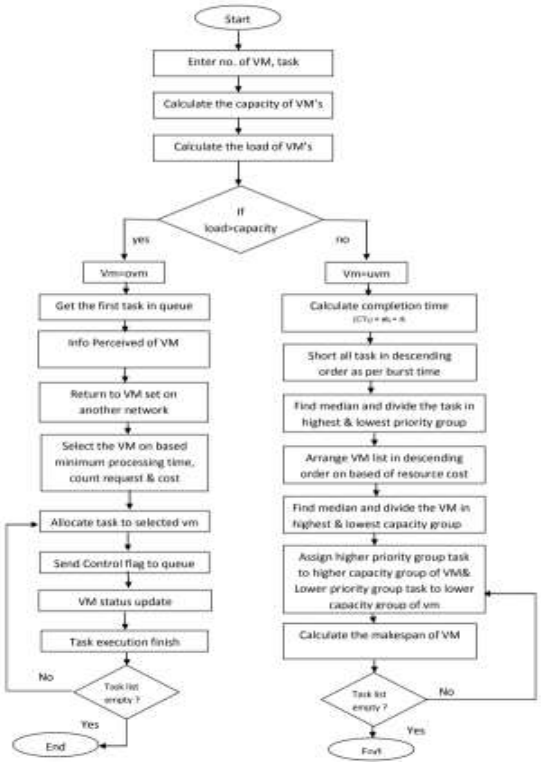

Figure 2 Flow Chart of Given Process

PROPOSED ALGORITHM

New Implementation

1. Start
2. Enter no VM i.e. VM1, VM2... Task i.e. T1, T2, ..
3. Calculate the capacity of VM
4. Calculate the load of VM

LVMi = Task Length I / Service Rate I

5. Check if (load > capacity)

VM= OVM

Else

VM= UVM

6. If VM is overload then (Apply modified Honey-Bee algorithm)

6.1 Get the first task in queue

6.2 Info perceived of VM

6.3 return to VM set on another network

6.4 Select the VM on based minimum processing time, count request & cost

6.5 Allocate task to selected VM

6.6 Send control flag to queue

6.7 VM status update

6.8 Task execution finish

6.6 If task list is not empty

Then go to 6.5

Else
End
7. If VM is under load then (apply modified Max-Min algorithm)
7.1 Calculate Completion time (CTij) =etij+rtj;
7.2 Short all task in descending order as per burst time.
7.3 Find median and divide the task in highest & lowest priority group
7.4 Consider three groups according to priority:
(1) Highest priority group
(3) Lowest priority group
7.5 Take highest priority of group & Arrange task in descending order of burst time.
7.6 Arrange Vm list in descending order on the basis of Resource Cost as
Resource cost = (RAM of Virtual machine * Cost/memory) +
(Size of Virtual machine*Cost/storage)
7.7 Find median and divide the VM in highest & lowest capacity group
7.8 Assign higher priority group task to higher capacity group of VM&Lower priority group task to lower capacity group of vm
7.9 Calculate the Make span of VM
7.10 If Task list is not empty
Then go to 7.8
Else
End
8. Stop

4. RESULTS AND ANALYSIS

SIMULATION ENVIRONMENT

Sr. No	Item Name	Value
1	Hardware	Core i9 3.00 GHz, 16 GB Ram
2	Operating System	Windows 10 (64 bit)
3	Simulator	CloudSim (8.2 Version)
4	Platform	Eclipse

Table 1 Simulation Environment

In this research paper above simulation environment is taken which is analyze with the parameter transmission time, make span and resource utilization.

Equation for Transmission Time:

$$Transmissiontime = \frac{Sizeofthetask}{BandwidthoftheVMs} \quad -(1)$$

Equation for Make Span:

$$Makespan = \max_{1 \le i \le m} \{Completiontime_i\} \quad -(2)$$

Equation for Resource Utilization:

$$Resourceutilization = \frac{\sum_{i=1}^{m} Completiontime_i}{Makespan \times m} \quad -(3)$$

SIMULATION RESULT OF DIFFERENT EXPERIMENTS

The algorithm which is proposed is implemented in java language to developed hybrid load balancing approach. In the proposed algorithm different experiments were conducted for finding results. Total three experiments were performed by using different Virtual Machine and 5 Data Centers. In experiment 1 virtual machine were taken 5. In experiment 2 virtual machine were taken 50. While experiment 3 150 virtual machines were taken into consideration and different results have been obtained. Below are different table and graph for different experiments.

Experiment-1:

No of VMs 5	Number of Task	Transmission Time(seconds)			Makespan (seconds)			Resource Utilization %		
		Honey Bee	Min-Max	HHM (Hyrbid Honeybee+MinMax)	Honey Bee	Min-Max	HHM	Honey Bee	Min-Max	HHM
	10	0.6	0.664	0.4	1.28	1.35	1.11	96	80	92
	20	0.9	1.114	0.45	2.76	2.78	2.15	97	82	93
	30	1.13	1.674	1	4.48	5.34	4.14	98	86	95
	40	1.19	1.927	1.1	5.13	5.79	4.48	99	87	97

Table 2: Performance analysis of proposed algorithm when virtual Machine is 5

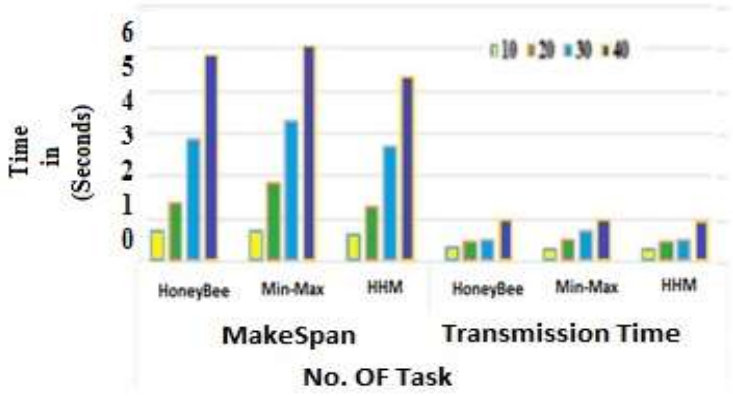

Figure 3**.** Experiment result of transmission time and make span when Virtual Machine is 5.

Experiment-2

No of VMs 50	Number of Task	Transmission			Makespan (seconds)			Resource Utilization %		
		Honey Bee	Min-Max	HHM (Hyrbid Honeybee+MinMax)	Honey Bee	Min-Max	HHM	Honey Bee	Min-Max	HHM
	10	0.53	0.553	0.51	0.77	0.62	0.57	84	93	92
	20	1.23	0.98	1.18	1.84	1.24	1.78	87	97	90
	30	1.58	2.09	1.42	2.38	2.45	2.34	92	98	91
	40	2.16	3.78	2.08	4.76	4.56	4.45	97	99	96

Table3**:** Performance analysis of proposed algorithm when virtual Machine is 50.

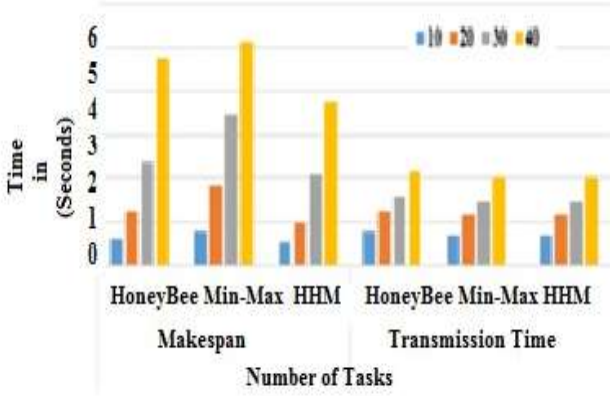

Figure 4 Experiment result of transmission time and make span when Virtual Machine is 50.

Experiment –3

No of VMs 150	Number of Task	Transmission			Makespan (seconds)			Resource Utilization %		
		Honey Bee	Min-Max	HHM (Hyrbid Honeybee+MinMax)	Honey Bee	Min-Max	HHM	Honey Bee	Min-Max	HHM
	1000	0.76	0.82	0.67	0.78	0.64	0.58	90	93	87
	2000	0.67	0.85	0.66	1.83	1.23	1.71	87	96	85
	4000	0.65	0.78	0.64	2.45	2.22	2.19	85	97	83
	6000	0.64	0.79	0.62	4.88	4.55	4.14	81	96	79

Table 4**:** Performance analysis of proposed algorithm when virtual Machine is 150.

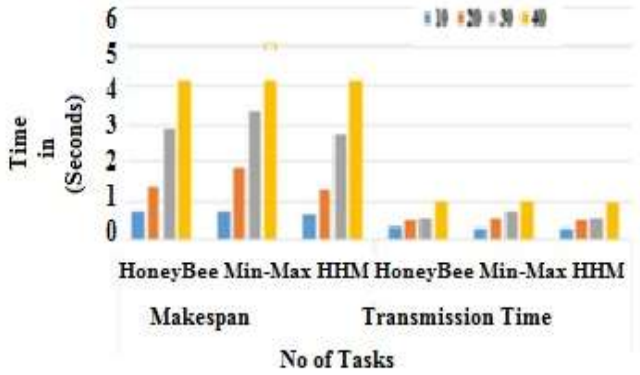

Figure-5 Experiment result of transmission time and make span when Virtual Machine is 150

.

In the above tables and figures, different task was performed on Honeybee, Min-Max and HHM (Hybrid Honeybee Minmax). Comparison were done using different virtual machine and results were generated of transmission time, make span and resource utilization.

5. CONCLUSION

In this current situation the major challenge in load balancing is cloud computing. To deliver proper resources, services, response time and make span for satisfaction of customer. To check the current status of node static load balancing algorithm is used. When to check previous data of

node and status dynamic load balancing is used. Many Researchers have work on static and dynamic but still efficient result has not been achieved. Which result of evolution of hybrid algorithm. This algorithm is combination of static and dynamic to overcome the lacking facilities in above two. This paper is proposing of algorithm in cloud computing for better result in hybrid environment.

REFERENCES

1. Pradeep Kumar Sharma, Prem Shankar Kaushik, PrernaAgarwal,Payaljain, Shivangi Agarwal, Kamlesh Dixit, "Issues and challenges of Data Security in a Cloud computing Environment",IEEE 2017.
2. SuyelNamasudra , Pinki Roy, BalamuruganBalusamy, "Cloud Computing: Fundamentals and Research Issues", IEEE 2017
3. Nitin Kumar , Shrawan Kumar Kushwaha and Asim Kumar, "Cloud Computing Services and its Application" Advance in Electronic and Electric Engineering., 2014
4. Prof. SatvikKhara, Prof.Umanag Thakkar, "A Novel Approach for Enhancing Selection of Load Balancing Algorithms Dynamically in Cloud Computing", IEEE 2019
5. EinollahJafararnejadGhomia, Amir MasoudRahmania, NooruldeenNasihQaderb," Load –Balancing algorithms in cloud computing: A Survey "Science Direct 2020.
6. Sambit Kumar Mishra, 1Bidhudatta Sahoo, PritiParamitaParida," Load Balancing in Cloud Computing: A big Place", Journal of King Saud University- Computer and Information Sciences 2018.
7. Mokhtar A Alworafi, AtyafDhari, AsmaA.Al-hashmi," An improved SJF Scheduling Algorithm in Cloud Computing Environment", IEEE 2016.
8. Pandaba Pradhan, PrafullaKu.Behera, B N R Ray," Modified Round Robin Algorithm for Resource Allocation in Cloud Computing, Science directs 2016.
9. Gaurangpatel, Rutvik Mehta, Upendrbhoy," Enhanced Load Balanced Min Min Algorithm for Static Meta task scheduling in Cloud Computing, Science direct 2014.
10. James kokKonjaang, Fahrul hakim Ayob and Abdullah Muhammed," Cost Effective Expa- Max-Min Scientific Workflow Allocation and Load Balancing Strategy in Cloud Computing", Research gate 2020.
11. S. Wang K . Yan , S. Wang and C. Chen , A Three-Phase Scheduling in a Hierarchical cloud computing network, 3rd International conference on communications and mobile computing IEEE Computer Society Press, 2018.
12. Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu H., and Abraham , A. Hybrid job scheduling algorithm for cloud computing environment. IEEE 2018
13. Zhao, C., Zhang S., Liu, Q., Xie, J., and Hu, J. Independent tasks Scheduling Based on Genetic Algorithm in Cloud Computing. 5th International Conference on Wireless Communications, Networking and Mobile Computing 2019.