Journal of Education for Pure Science- University of Thi-Qar
Vol.12, No.1 (March, 2022)
Website: jceps.utq.edu.iq Email: jceps@eps.utq.edu.iq

DOI: http://doi.org/10.32792/utq.jceps.12.01.08

Task Scheduling In Cloud Computing Based Deadline constraint

zahraa khadim [1] , Khaldun Ibraheem Arif[2]

[1,2] Department of Computer Science, College of Education for Pure Science, University of ThiQar, Nassiriya, Iraq.

Received 22/2/2022 Accepted 22/3/2022 Published 30/3/2022

Abstract:

One of the newest technologies is cloud computing most aspects in distributed systems that are most appealing . On-demand services are available on a pay-as-you-go system basis. In cloud computing, Major research subjects include work scheduling and genetic algorithms. Task scheduling refers to the process of allocating tasks to resources (virtual computers), while genetic algorithm is the process of creating a community between resources in order to find optimal solutions to issues using the theory of natural selection. We present a genetic approach to job scheduling with deadline constraints in this paper. Each time the virtual machine load plus completion time is tested smaller and equal to the capacity of the virtual machine and smaller and equal to the task deadline, two loops are formed, one for tasks and one for virtual machines The suggested algorithm is compared to various algorithms Existing in use, including as" An Effective Load Balancing Algorithm Based on Deadline Constraint(ELBAD) "and" Elastic Load Balancer (ELB)" and the experimental results show that the proposed algorithm is superior to others in terms of reducing rejected tasks and maximizing accepted tasks.

Keyword: Cloud Computing, task priority, scheduling algorithm, Virtual Machine, Genetic Algorithm (GA).

1. Introduction:

Cloud Computing affair Indicat to the resources and services that are submitted through the internet, these services are submitted in a cloud environment to adapt to user needs. The growing use of cloud resources or services by people from all walks of life necessitates a focus on cloud user management and their needs. Clients access each or part of these resources depending on the cloud system utilized with cloud computing. The use of developed equipment and virtual simulation technology, as well as the usage of distributed systems, has accelerated the development of cloud computing recently Because of the rapid growth of cloud computing services and the growing number of users, it is necessary to administer each of those cloud users and their requests. Job Scheduling is one of the most significant aspects of preserving user requests in the cloud environment; it is used to improve system performance and reduce task performance time. A task scheduler technique is regarded as a prerequisite for providing efficient cloud computing services. It works by assigning function needs to users for available resources and reducing total response time. [1]. Cloud computing prevalence in Our life as Email.

2. Related Work:

The challenge of work scheduling in a distributed context has piqued scholars' interest in recent years. The key concern is the execution time, which should be kept as short as possible. Task scheduling, on the other hand, is a crucial issue in the Cloud computing environment since it takes into account several parameters such as completion time, total cost of executing all users' activities, resource utilization, power consumption, and fault tolerance.

GE Junwei [2] has proposed a static evolutionary algorithm that takes total job completion time, average task completion time, and cost constraints into account.

Allocating the appropriate resource to the arriving jobs is one of the scheduling issues. If numerous jobs come at the same time, the dynamic scheduling process is termed difficult., S. Ravichandran and D. E. Naganathan [3] have created a solution to avoid this problem by letting arriving tasks to wait in a queue while scheduling recalculates and sorts them. As a result, scheduling is done by selecting the first task from the queue and allocating it to the resource with the best fit using GA. The goal of this system is to maximize resource consumption while also reducing execution time..

R. Kaur and S. Kinger [4] Enhancement GA based on a job scheduling method has been proposed. A new fitness function based on mean and grand mean values is used. They believe that their technique could be used to schedule tasks and resources.

On the Cloud computing environment, a comparison of three task scheduling algorithms - round-robin, pre-emptive priority, and shortest remaining time first algorithms - was conducted [5].

V. V. Kumar and S. Palaniswami [6] have published a study aimed at improving the job scheduling algorithm's efficiency for real-time Cloud computing services. They've also developed an algorithm for maximizing turnaround time by giving high priority to tasks with a short completion time and low priority to real-time abortion issues. Z. Zheng, et al. [7] have developed a GA-based technique for dealing with the scheduling problem in the Cloud computing context. called Parallel Genetic Algorithm (PGA) to use mathematics to accomplish optimization or sub-optimization for Cloud scheduling difficulties

Furthermore, from the standpoint of a Cloud provider, one of the key purposes of task scheduling is to maximize profit by employing resources efficiently. Therefore, K. Thyagaarajan, et al. [8] have developed a task scheduling model for use in the Cloud computing environment in order to maximize profitability for the Cloud computing service provider. In [9], S. By introducing numerous variations for job scheduling in the Cloud computing environment, Singh has provided an in-depth understanding of GA. By refining GA, he created an algorithm to tackle task scheduling problems in which the starting population is formed using the Max-Min strategy to provide more optimal results in terms of "makespan."

Khaldun Ibraheem Arif [10] , in 18/11/2020 Propose An Effective Load Balancing Algorithm Based on Deadline Constraint Under Cloud Computing (ELBAD) To reduce makespan And maximing resource exploitation. The ELBAD appropriates the closest deadline tasks every Once to the maximum speed Virtual Machines (VMs) then it Balances burden work between VMs. The suggested algorithm is compared with other existing algorithms such as FCFS, SJF, Min-Min and EDF and Demo results show Excellenced of ELBAD over others Super.

Mohit Kumara, Kalka Dubey, S.C.Sharma [11] , Procedia Computer Science 125 (2018) 717–724, Suggest algorithm Elastic and flexible deadline constraint load Balancing algorithm for loud Computing to balanced the load at all virtual machines and increase the ratio of task meet with deadline utilize the flexibility notion (saving and cancel saving of resources) .

3. Problem Statement:

In cloud computing, task scheduler needs to find an optimal assignment of tasks to virtual machines, given deadline constraints. The problem solved is to find the best order of tasks against the best order of virtual machines (choosing the optimal solution from among the solutions) to achieve the best meeting ratio and reduce the rejection rate so that a genetic algorithm is used to improve the results. The proposed algorithm relied on choosing a random set of addresses to arrange the tasks and another random set of addresses to arrange the virtual machines, and the meeting ratio was calculated each time. This process is repeated several times to obtain a set of solutions and the optimal solution is selected from among the solutions. The accepted tasks should be greater than the rejected tasks, for this purpose the proposed algorithm increased the percentage of meetings and reduced the percentage of rejections, this is done by optimizing the logical distribution of tasks to the number of available virtual machines using the genetic algorithm optimization process

The proposed algorithm in Matlab code was implemented on a Lenovo laptop with a Core i7 (2.6 GHz) CPU and 16 GB RAM on a 64-bit operating system.

4. Depending on the type of task to be scheduled, different scheduling methods can be utilized.

The scheduling techniques can improve execution efficiency while maintaining system load balancing. The methods utilized for work scheduling determine the cloud's efficiency. Preemptive or non-preemptive scheduling algorithms are also possible. No force can stop a job from being executed in a non-preemptive scheduling method. A task execution, on the other hand, may be slowed in the preemptive scheduling method due to a variety of circumstances. A set of criteria, including both turnaround time and waiting time, is required to evaluate the efficiency of scheduling algorithms [12] The turnaround time is the overall time it takes to complete a work, whereas the waiting time is the total time a task has been waiting in the ready queue. The CPU scheduling algorithm has no effect on the amount of time it takes for a job to run or deliver information; it only affects the amount of time it takes for a task to wait in the ready queue. A job can frequently offer some output legitimately early and can continue to figure out new outcomes while delivering previous results to the customer [13]. The following are some of the most common scheduling algorithms: First Come First Serve Scheduling Algorithm, Priority Scheduling Algorithm, and Genetic Algorithm [14].

1. **First-Come-First-Served Scheduling algorithm (FCFS):** is a simple scheduling algorithm in which processes are sorted according to their arrival time and submitted to the CPU.

2. Priority Scheduling algorithm: This scheduling algorithm is preemptive, and it is based on priority. Shortest Job First Scheduling Algorithm (SJF) or (Min-Min algorithm):

It achieves the shortest average waiting time by placing a short task time before a long one. It might be preemptive or non-preemptive in nature. For example, a new job arriving at the ready queue has the shortest burst time, but a previous task executing has the longest burst time. When a new task is being executed, the burst period is the shortest. A preemptive SJF algorithm will terminate the currently operating process, whereas a non-preemptive SJF method will allow the currently running task to complete its CPU burst [15] .

A. **Round-Robin Scheduling Algorithm (RR):** another method of prioritization scheduling. This is the simplest, most equitable, and widely utilized scheduling algorithm. The smallest unit of time, known as time slices or quantum, is used to run all processes in a circular queue.

B. At time 0, the following set of processes appears, with the CPU-burst time recorded in milliseconds. Algorithm Max-Min: It is the antithesis of SJF in that it prioritizes the larger tasks for completion first.

Genetic Algorithm (GA):- is a type of algorithm that is used to solve problems. It is a problem-solving strategy based on the genetics model. GA is a search technique for locating an optimal solution [16].

5. Problem Formulation and Proposed Algorithm

The proposed technique consists of two parts they are:

- The fitness function.
- main algorithm.

5.1. The fitness function

The fitness function contains logical steps to solve the problem by distributing the tasks on the virtual machines available in the system figure(1) shows the fitness function steps.

```
Fitness function
Inputs:
        Tasks and VMs according new populations and crossover.
Outputs:
        Meakspane, average of response time , ARU, AWT, meting ratio, rejection ratio
start
1. Compute capacity of each VM
2. Compute CT[i][j] for each task i on every VM j.
3. Set metcount = 0; rejcount = 0
4. For i = 0 to N-1 do
5. For j = 0 to M-1 do
6. If ((VM[j].load + CT[i][j]) ≤ VM[j].capacity) then
7. If ((VM[j].load + CT[i][j]) ≤ task[i].deadline) then
8. Task[i].start_time = VM[j].load;
9. VM[j].load = VM[j].load + CT[i][j];
10. Task[i].finish_time = VM[j].load;
11. Metcount = metcount + 1;
12. Break;
13. Else
14. Rejcount = rejcount + 1;
15. Break;
16. End if
17. Else
18. Continue;
19. End if
20. End for
21. End for
22. Calculate makespan ;
23. Calculate average of response time;
24. Compute ARU;
25. Compute AWT;
26. Compute meeting ratio and rejecting ratio.
27. End
```

figure(1) :The fitness function steps.

- **The fitness function steps**

Step1: In this step, the capacity of each virtual machines is calculated. To calculate the capacity of each virtual machines, equation No. 1 is used.

$$VM_j.capacity = (VM_j.MIPS / \sum_{k=0}^{M-1} VM_k.MIPS) * 100 \quad \text{.........................(1)}$$

Where VM.MIPS is the Million Instruction per task and k=0,1,2,3….,M-1.

Step2: calculate the Completion Time (CT), To calculate the Completion Time , equation No. 2 is used.

$$CT_{ij} = t_i / (VM_j * PE) \ldots\ldots(2)$$

Where PE denotes processing element and i=1,2,….N, j=1,2,….M.

Steps 3-21: In these steps, two loops are created, one for tasks and the other for virtual machines, Each time it is tested if the load of the virtual machine plus Completion Time is smaller and equal to the capacity of virtual machine and smaller and equal to the deadline of the task. If the condition is met, the virtual machine is allocated to perform the task, otherwise it will search for another virtual machine and if the condition is not met on all available virtual machines the task will be rejected.

Step 22: The makespan is calculated according to Equation No. 3

$$Makespan = \max\{Load\,VM_j\} \ldots\ldots(3)$$

Where j = 1, 2, …..,M.

Step 23: The average of response time(ART) is calculated according to Equation No. 4 and according to Equation No. 5.

$$RT_i = task_i.start\_time - task_i.submit\_time \ldots\ldots(4)$$

Where i=0,1,2,…..,N-1; and N number of tasks.

$$ART = (\sum_{i=0}^{N-1} RT) / N \ldots\ldots(5)$$

Step 24: The Average Resource Utilization (ARU) is calculated according to Equation No. 6

$$ARU = (\sum_{j=0}^{M-1} Load\,VM_j) / Makespan * M) \ldots\ldots(6)$$

Where j=0,1,2,…..,M and M is number of virtual machines.

Step 25: The average of waiting time(AWT) is calculated according to Equation No. 7 and according to Equation No. 8.

$$WT_i = FT_i - (FT_i - ST_i) - SUBT_i \ldots\ldots(7)$$

Where FT(i) refers to Finish Time of task (i) and ST (i) means its start time, while SUBT (i) denotes its submit time.

$$\text{AWT}=\sum_{i=0}^{N-1} WT_i / N \ldots\ldots(8)$$

Where i=0,1,2,…..N; and N is number of tasks.

Step 26: calculate The ratio of tasks meeting their deadline from Equation No. 9

$$\text{Meeting ratio} = \text{NO.meeting tasks} / \text{N} * 100 \ldots\ldots(9)$$

Where and N is number of tasks

5.2 The main algorithm

The genetic algorithm was used to optimization the proposed technique with the aim of improving the meeting ratio and reduce the rejection ratio, This algorithm is always used to find the best solutions by selecting a large number of random inputs on the fitness function, and then choosing the best solution from the output solutions. figure(2) shows the main algorithm steps.

```
Algorithm optimization
Inputs:
        Tasks, VMs and iteration number
Output:
        Best solution.
Start
1. for i=1: iteration numbers
2.select random N  and M population
3. exchange position of tasks by N population and exchange position of VMs by M population
4. crossover tasks and VMs indexes by random indexes
5. call fitness function and get solution
6. end for
7.output best solution
8.end
```

figure(2) :The main algorithm steps.

5.2. The main algorithm steps

Step1: Choose the number of iteration required for the optimization process, The process of obtaining solutions from the fitness function will be repeated by the number of iterations , If the number of iteration increases, better solutions will be obtained. After selecting the number of the iteration a loop will be made for the purpose of implementing steps 2 to 6, which represent the steps of the main program.

Step 2: Select random number N and M populations, N represents the new indexes for tasks and M represents the new indexes for the virtual machines.

Step 3: Exchange position of tasks by N population and exchange position of VMs by M population.

Step 4: Crossover tasks and VMs indexes by random indexes, A number of new random indexes are chosen in order to replace them with a number of tasks and VMs indexes.

Step 5-6: The fitness function is called and the inputs are given in the new order and the solution is obtained and stored until all the solutions are completed and the number of them is the number of iterations. The loop is terminated when the counter is equal to the number of iteration.

Step 7: The best solution, which represents the highest value for the meeting ratio, is selected.

6. Analysis and Comparison of Experimental Results

The proposed algorithm was applied on set of random tasks and virtual machine values with different number of tasks and different number of virtual machine. Flowing tables (1), (2), (3), (4), (5) and (6) shows the results of some metrics as meeting ratio, rejection ratio, Resource Utilization, Average waiting time, Response time and Makespan. The results showed a significant improvement in meeting ratio, rejection ratio and Resource Utilization.

Table (1): the ratio of meeting constraint (acceptance).

Number of tasks	Number of VMs	ELBAD algorithm	ELB Algorithm	Proposal algorithm
1000	25	99.2	98.8	100
1200	30	96.66	96.25	99.5
1400	30	98	97.42	98.28
1600	35	96.81	95	97.12
1800	35	96	95.38	96.61
2000	40	95.85	95.3	96.15
2200	40	93.59	95	95.59
Average		96.587	96.164	97.607

Table (2): The ratio of rejected tasks.

Number of tasks	Number of VMs	ELBAD algorithm	ELB Algorithm	Proposal algorithm
1000	25	0.8	1.2	0
1200	30	3.33	3.75	0.5
1400	30	2	2.57	1.71
1600	35	3.18	5	2.87
1800	35	4	4.61	3.38
2000	40	4.15	4.7	3.85
2200	40	6.4	5	4.40

Table (3): Resource utilization measured in seconds

Number of tasks	Number of VMs	ELBAD algorithm	ELB Algorithm	Proposal algorithm
1000	25	0.32	0.61	0.16
1200	30	0.43	0.58	0.2
1400	30	0.55	0.82	0.3
1600	35	0.49	0.61	0.28
1800	35	0.49	0.66	0.34
2000	40	0.69	0.72	0.36

2200	40	0.48	0.51	0.38

Table (4): Average waiting time measured in seconds.

Number of tasks	Number of VMs	ELBAD algorithm	ELB Algorithm	Proposal algorithm
1000	25	2935.68	2703.73	2925.62
1200	30	2672.02	2037.59	2820.85
1400	30	2651.11	2317.04	2535.98
1600	35	2984.03	2634.03	2973.72
1800	35	3327.98	2670.73	3188.79
2000	40	2824.93	2526.32	3257.62
2200	40	3982.47	3415.66	3597.80

Table (5) : Average response time measured in seconds.

Number of tasks	Number of VMs	ELBAD algorithm	ELB Algorithm	Proposal algorithm
1000	25	2937.26	2708.59	2926.28
1200	30	2677.74	2046.61	2822.57
1400	30	2655.62	2325.73	2540.26
1600	35	2991.42	2646.55	2980.34
1800	35	3338.01	2683.28	3198.3
2000	40	2835.89	2542.20	3267.76
2200	40	3999.56	3432.86	3609.34

Table (6): Makespan results measured in seconds

Number of tasks	Number of VMs	ELBAD algorithm	ELB Algorithm	Proposal algorithm
1000	25	6.4	6.4	6.37
1200	30	5.63	5.7	5.33
1400	30	3.99	4.05	3.9
1600	35	4.63	4.66	4.74
1800	35	4.26	4.28	3.88
2000	40	3.41	3.45	3.80
2200	40	4.78	4.84	3.84

Average	4.7286	4.7686	4.5514

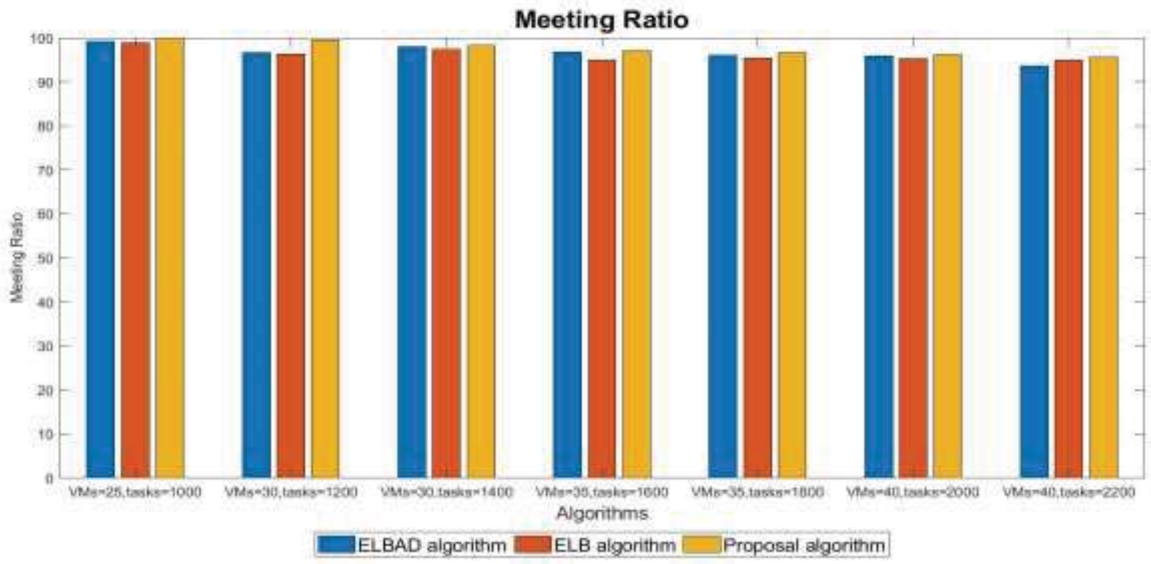

Figure (1) Meeting Ratio

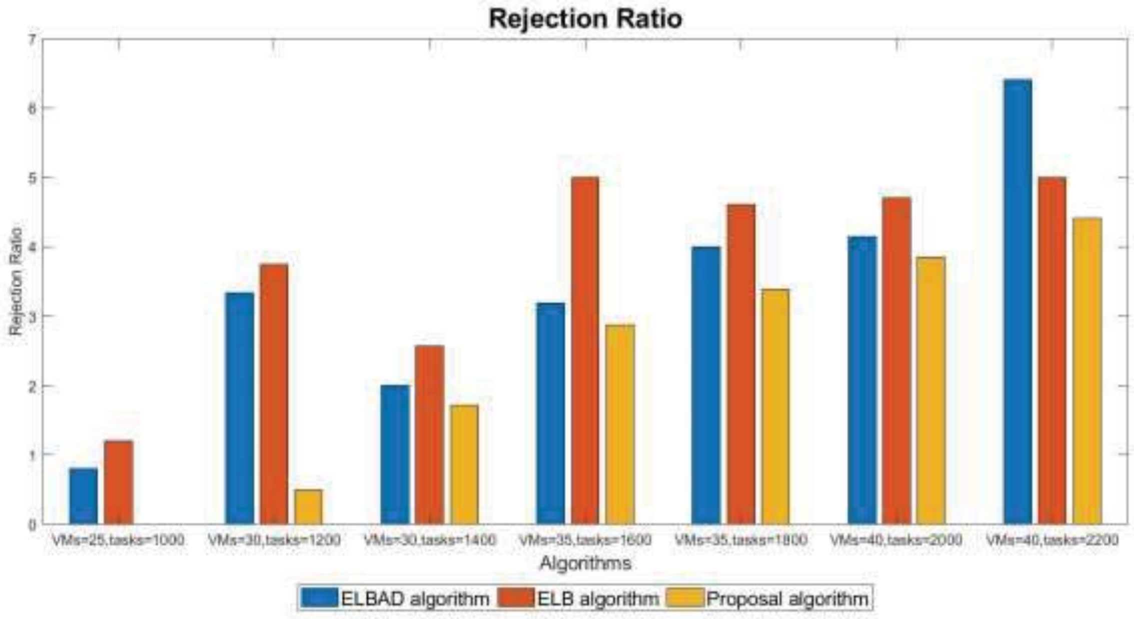

Figure (2) Rejection Ratio

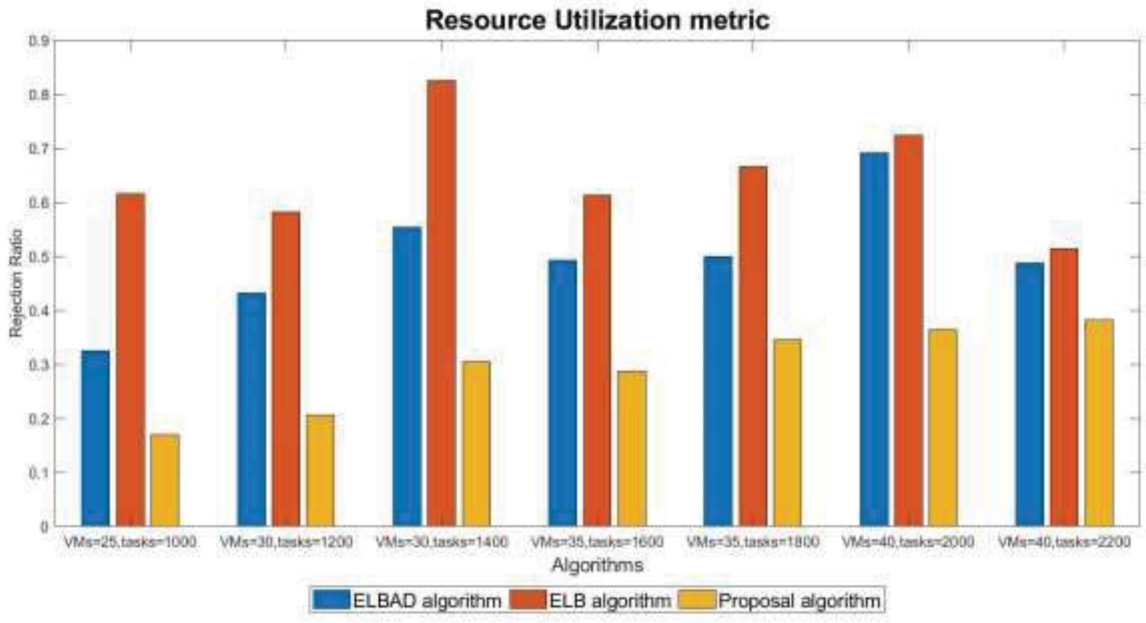

Figure (3) Resource Utilization metric

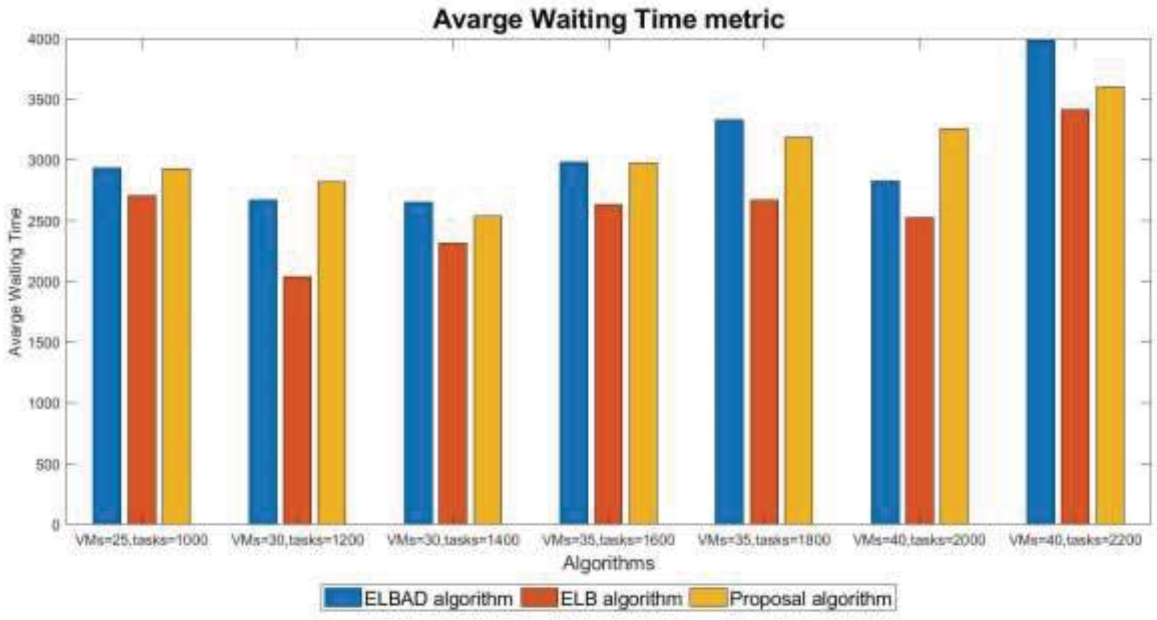

Figure (4) Average Waiting Time

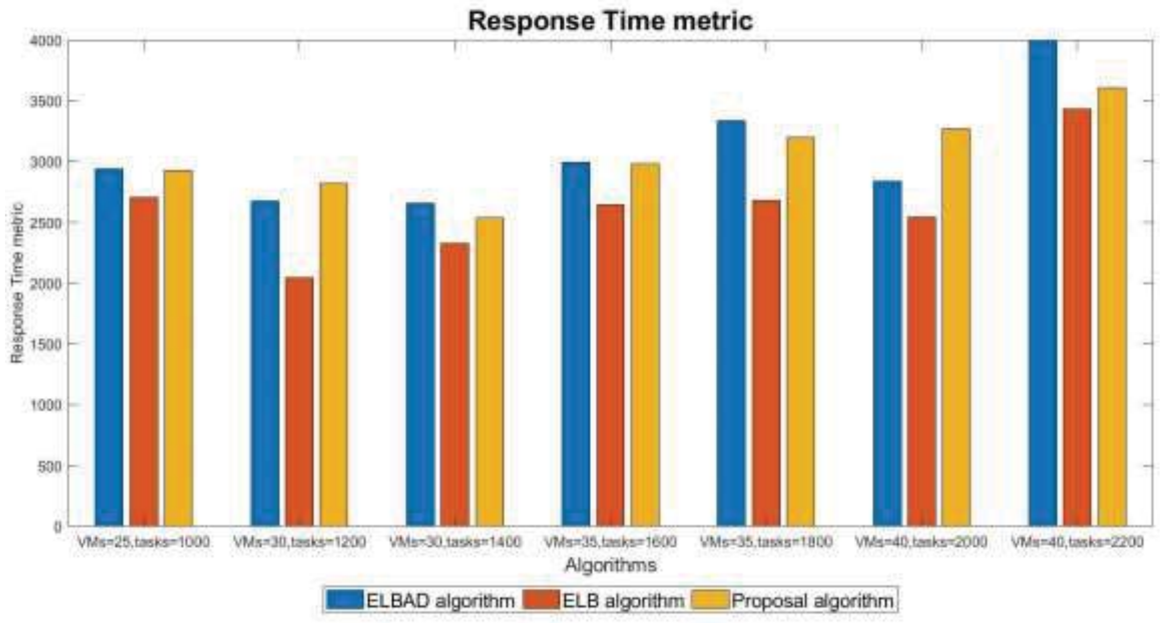

Figure (5) Response Time metric

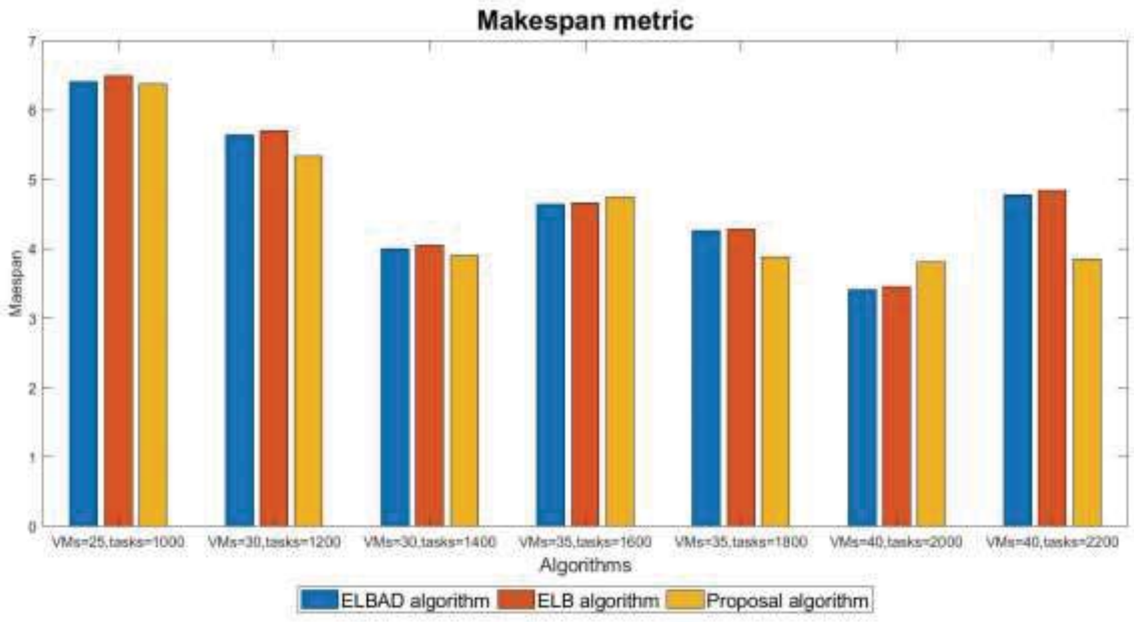

Figure (6) Makespan metric

7. Discussion experimental results

In this section, the results obtained from the simulation of the proposed algorithm and their comparison with the ELBAD and ELB algorithms will be discussed.

Where it was observed that the results in the first, second, and third tables, which represent the Meeting Ratio, rejection Ratio, and Resource Utilization metrics of proposed algorithm Significantly improved compared to the Effective Load Balancing Algorithm with Deadline constraint (ELBAD) and Effective Load Balancing (ELB) algorithms so the proposed algorithm outperforms the rest of the algorithms.

8. Conclusion and future work:

From comparing the results of the proposed algorithm with other known algorithms, the proposed algorithm is best than others and the meeting ratio that observed from results is better than other algorithms and the rejection ratio that observed from results is less than other algorithms. The genetic algorithm optimization it has a significant impact on improving the results. The resource utilization was much lower than the rest of the algorithms, although the proposed algorithm performs more tasks. The average of Makespan that observed from results is less than other algorithms .Modify the proposed algorithm by using whale optimization. Modify the proposed algorithm by using PSO optimization. We can introduce neural network to obtain optimal distribution. Propose new algorithm for tasks distribution by using a statistical manner.

Reference:

[1] (Aridah, Mohammed Ibrahim, 2016) "Task Scheduling Using Best-Level-Job-First on Private Cloud Computing. "A thesis Submitted in Partial Fulfillment of the Requirements for the Master Degree in Computer Science Department of Computer Science Faculty of Information Technology Middle East University August .

[2] (Yuan, J. W. Ge and Y. S., 2013) "Research of cloud computing task scheduling algorithm based on improved genetic algorithm," in Applied Mechanics and Materials , pp. 2426-2429.

[3] (S. Ravichandran and D. E. Naganathan, 2013) "Dynamic Scheduling of Data Using Genetic Algorithm in Cloud Computing," International Journal of Computing Algorithm, vol. 2, pp. 127-133 .

[4] (R. Kaur and S. Kinge, 2014)"Enhanced Genetic Algorithm based Task Scheduling in Cloud Computing," International Journal of Computer Applications, vol. 101 .

[5] (V. Vignesh, K. Sendhil Kumar, and N. Jaisankar, 2013)"Resource management and scheduling in cloud environment," International Journal of Scientific and Research Publications, vol. 3, p. 1 .

[6] (V. V. Kumar and S. Palaniswami, 2012)"A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing," Journal of computer science, vol. 8, p. 780 .

[7] (Z. Zheng, R. Wang, H. Zhong, and X. Zhang, 2011)"An approach for cloud resource scheduling based on Parallel Genetic Algorithm," in Computer Research and Development (ICCRD), 2011 3rd International Conference on, pp. 444-447.

[8] (K. Thyagarajan, S. Vasu, and S. S. Harsha, 2013) , "A Model for an Optimal Approach for Job Scheduling in Cloud Computing," in International Journal of Engineering Research and Technology .

[9] (S. Singh and M. Kalra, 2014)"Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm," in Computational Intelligence and Communication Networks (CICN), International Conference on, pp. 565-569.

[10] (Khaldun Ibraheem Arif, 2020) "An Effective Load Balancing Algorithm Based on Deadline Constraint Under Cloud Computing" , IOP Conf. Ser.: Mater. Sci. Eng. 928 032070.

[11] (Mohit Kumar , Kalka Dubey , S.C.Sharma, 2017)"Elastic and flexible deadline constraint load Balancing algorithm for cloud computing " , ICSCC ,7-8 December 2017, Kurukshetra ,India.

[12] (Salot, P., 2013) , "A survey of various scheduling algorithm in cloud computing environment,", International Journal of Research and Engineering Technology (IJRET), Vol. 2(2), pp. 2319-1163.

[13] (Goel, N.; and Garg, R. B, 2012)"A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics and Image Processing, Vol. 2(4), pp. 245-251 .

[14] (Shimpy, E., &Sidhu, M. J, 2014) “ Different Scheduling Algorithms In Different Cloud Environment”. Algorithms, International Journal of Advanced Research in Computer and Communication Engineering, 3(9),8003-8006.

[15] (Kl Arif,AS Ketab)“Dynamic Time Quantum for an Efficient Round Robin in Cloud Computing” ,Journal of computational and Theoretical Nanoscience 16(5/6),2404-2409.

[16] (Kaur, R., &Kinger, S, 2014) “ Analysis of Job Scheduling Algorithms in Cloud Computing”. International Journal of Computer Trends and Technology (IJCTT),9(7), 379- 386.