International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET222934

Enhancing Response Time of Cloud Resources Through Energy Efficient Cloud Scheduling Algorithm

Priyal Ghetiya*, **Prof. Dhaval Nimavat**
Department of Computer Engineering, R.K. University, Rajkot, Gujarat, India

Article Info
Volume 9, Issue 3

Page Number : 354-356

Publication Issue :
May-June-2022

Article History
Accepted : 20 May 2022
Published: 05 June 2022

ABSTRACT

Cloud Computing is becoming a dominant trend in providing information technology (IT) services. The cloud comprises many hardware and software resources today, and more people are switching to such services. Users' requests for cloud resources must incur a minimum amount of load on the system while getting a rapid response. In the cloud today, there is too much computational power. Load balancing makes it possible for various components of the cloud computing environment to work efficiently. To balance client requests to available resources so that the system is not overloaded, and the requested resources are delivered as quickly as possible, an effective load balancing strategy is essential. In this research article, we have presented a critical analysis of various existing cloud load balancing and scheduling algorithms. Several task scheduling approaches have been proposed in the literature review, but there appears to be a lack of scheduling algorithms for real-time task works based on historical scheduling records (HSR). The proposed algorithm uses information available in HSR to efficiently distributes incoming user requests to available virtual machines. The proposed scheduling algorithm uses the scaleup and scale down resource algorithm which helps in achieving maximum resource utilization. The algorithm tries to balance the load on VMs by scaling up and down cloud resources. WorkflowSim is used to analyze the performance of the algorithm proposed. The simulation results are compared with the existing scheduling algorithm which shows the proposed algorithm outperforms existing scheduling algorithms in terms of makespan.

Keywords : Scheduling, Reducing energy consumption, Load balancing, Virtual machine, WorkflowSim

I. INTRODUCTION

Cloud computing is an environment that is mainly connected with rapidly provisioning measured and scalable resources to users based on user demand and pay per user basis (Rai et al., 2020). “Cloud” in cloud computing is sometimes also referred to the same as “Internet” and computing concerned with internet-based computing resources such as storage, virtualization, and other services provided over the internet (Kumar, 2018). Instead of using the "own and use" strategy cloud computing employed a "pay–per–use" strategy in which the user only needs to pay for the number of services he/she used (Kumar, 2018, Rai et al., 2020).

Cloud computing achieved outstanding growth in both parallel and distributed systems. Cloud computing is the buzzword that comes into sight after many existing technologies like parallel computing, peer-to-peer technologies, distributed computing, virtualization, and many other technologies (Mishra, 2020). Cloud computing is the paradigm that provides various internet services such as storage, large infrastructure, scalability, and a pool of resources on pay per use basis. It is a new service delivery model for the IT industry and for other industries such as academia, business, etc. whose internet services are used on a "pay-per-usage" basis. Now a day, most applications reside on the internet, and users of that applications are growing rapidly. According to NIST (National Institute of Science and Technology), Cloud computing is a representation for enabling convenient and on-demand network access to shared pool memory of computing resources (e.g., networks, servers, storage, and applications) that can be rapidly conducted and released with minimal management, effort or service provider interaction (Alworafi, 2019). Before submitting your final paper, check that the format conforms to this template. Specifically, check the appearance of the title and author block, the appearance of section headings, document margins, column width, column spacing and other features.

Users request the cloud to provide them with its services given it consists of pool of resources. To improve the performance of the system, it is necessary to schedule the requests of the users properly (Tang et al., 2018). There are different stages to scheduling, including the Resource Discovery, Resource Selection, and Task Submission stages. The Resource Discovery stage involves discovering what resources are available in the system, the Resource Selection stage involves choosing what resources should be used, and the Task Submission stage involves submitting the task to the selected resource (Swarnakar, 2020).

During the last several decades, rapid improvements in computing power, storage, and networking technologies open many new opportunities for businesses and organizations to grow more and more and have allowed organizations or individuals to generate necessary information, process and share a large amount of information using the internet in dramatically new ways (Yadav, 2019). As a result, there is a need for a powerful scheduling strategy to overcome various resource constraints problems. This works aims at creating a novel scheduling policy to enhance the performance of cloud system by optimizing the response time of the system and by minimizing energy consumption.

Existing scheduling algorithms as discussed in the literature review consider some of the parameters of scheduling and also they face some drawbacks which are described in detail in the literature survey. So by analyzing these algorithms with each other a conclusion can be derived that a new efficient technique or algorithm can be developed based on the parameters of scheduling and which can eliminate the drawback of discussed algorithm and which can improve the performance of the system.

To overcome the problem discussed here a novel energy aware scheduling algorithm is developed which optimizes the overall response time of the system and enhances resource utilization by minimizing energy consumption.

II. LITERATURE REVIEW

The cloud system is currently being used by millions of people to perform their computing tasks using cloud resources. To use Cloud Computing in this way is a challenge because thousands of tasks have to be scheduled. The scheduler connects the user with the resources and allows them to manage the workload. Since the number of users in the cloud keeps on increasing, it becomes increasingly difficult to schedule the requests of the users (Mondal, 2018). Therefore a better scheduling algorithm must be adopted than what we have at the moment. It can be done by comparing and evaluating the various algorithms that exist to identify the differences between them, and thereby pinpoint what the problem is.

In this section, we present a literature review that has been conducted to evaluate the performance and functionality of different scheduling algorithms in cloud computing systems to develop a new recommendation. A more efficient scheduling algorithm was developed using the literature review published in this article and is presented as part of the proposed work section in this article.

Using energy-efficient dynamic scheduling (EDS) in a cloud data center, Panda and Jana (2019) present a method for scheduling real-time tasks. Based on historical scheduling records, the algorithm categorizes virtual machines and tasks. A schedule is then created that groups and schedules tasks of similar size and type to take advantage of all resources. A cloud simulation toolkit is then used to implement the proposed scheme. In comparing the simulation results with the existing scheduling algorithm, it is apparent from the comparison that the proposed scheme leads to greater efficiency and performance for scheduling, increased productivity for cloud data center resource utilization, further increased task reliability, and fast response times.

Zhang et. al. (2018) focus more on reducing migration costs as it leads to high payment costs to cloud users. Our proposed scheme is the first to propose a cost-effective energy-aware scheduling algorithm that eliminates the overheads of virtual machine migration while still guaranteeing the deadline of the task for users. The scheme postponed tasks having relatively loose deadline constraints and without waking-up new physical machine algorithm minimizes energy consumption. An energy and deadline-aware non-migration scheduler is based on a heuristic task scheduling algorithm called EDA-NMS (Energy and Deadline Aware Non-Migration Scheduler). The algorithm is implemented using the cloudsim simulator tool. The simulation results indicate that this algorithm outperforms other existing scheduling algorithms with respect to energy efficiency and deadline guarantee as the overhead of VM migration is minimized while achieving efficient energy consumption.

Ashikolaei et al. (2018) have answered a question that has always been a problem in distributed computing, and this question is called load balancing. As there are numerous users and a vast amount of resources, it is not surprising that many researchers consider load balancing to be an NP-hard problem. Because of the sheer amount of resources and many users using these resources, it is difficult to do. This study's goal is to construct an intelligent meta-heuristic algorithm that combines the ICA and FA elements to get the specified desired result by a meta-heuristic algorithm. Previous researchers had proposed heuristic algorithms like the imperialist competitive algorithm (ICA) and the Frefy Algorithm (FA) to solve the said problem that was referred to earlier. To analyze the performance of the proposed scheme, various parameters such as the duration of the scheme, load balancing, stability, and planning speed have been considered (Malhotra, 2018). Using the proposed algorithm, all of these parameters have been improved dramatically in all of the obtained results.

An efficient way for scheduling complex tasks for heterogeneous cloud computing systems has been presented by Seth and Singh (2018). The DHSJF algorithm is a dynamic resource provisioning scheme that reduces energy consumption by dynamically mapping user requests to heterogeneous resources (Mulla, 2019). In addition, DHSJF schedules the tasks in such a way as to minimize actual CPU time and the time taken for them to be executed throughout the entire system. This dynamic load balancing algorithm is executed when the system is actively running. A dynamic heterogeneity helps to decrease the time to perform the task, which leads to increased utilization of resources. Among the many features discussed in this paper is the inclusion of both dynamic heterogeneities of workloads as well as dynamic heterogeneities of resources. Compared to the existing algorithm, the DHSJF algorithm produces better results. CloudSim has been used to simulate the proposed algorithm.

Although there have been numerous approaches to scheduling tasks, there does not appear to be a concept for merging real-time tasks that are based on classification. Avinab et. al., (2019) presents an energy-efficient dynamic scheduling scheme (EDS) for virtualized CDCs which includes optimizing the real-time scheduling of real-time tasks across a network of nodes. Scheduling records are used to classify tasks and virtual machines. Simulation results are compared to existing algorithms with the proposed scheme being implemented in cloudsim.

III. PROPOSED SYSTEM

Cloud computing involves scheduling as one of its key concepts. The performance of the system could be increased if tasks are assigned to an appropriate resource. Surveying the existing scheduling algorithm and looking at its advantages and disadvantages, a new concept arrived in mind which could lead to better performance of the system by eliminating the drawbacks of existing algorithms which comprise maximum parameters of scheduling.

To overcome the problem discussed in the literature review, the novel scheduling algorithm is presented here. The proposed algorithm uses scaleup and consolidation mechanisms to scale up resources in case of not sufficient resources and scale down resources if resources are ideal. This strategy helps in achieving efficient resource utilization. The proposed scheduling algorithm uses a scaleup and scales down strategy in place of resource migration to save migration time, which helps in minimizing the response time of the cloud system.

The below table shows all terms used in the algorithm with their symbolic notations and their description.

TABLE I

TERMS USED IN ALGORITHM WITH A DESCRIPTION

Terms	Description
H	Set of Hosts
V	Set of VMs
T	Set of Tasks
H^{active}	Set of active hosts
H^{off}	Set of powered-off hosts
H^{idle}	Set of idle hosts
V^{r}	VM resource requirement
T^{r}	Task resource requirement
t^{idle}	Threshold ideal time
V^{idle}	Ideal time of VM
T^{d}	Task Deadline
HSR	Historical Scheduling record

Here in this section, we have presented three algorithms that can be used for optimizing scheduling in the cloud computing network. Based on the tasks and resources available to VMs, the Avinab Marahatta scheduling algorithm determines what tasks should be scheduled. This task will be assigned to VM if it is capable of handling the request. VM utilization is not considered when this algorithm determines scheduling, which leads to inefficient resource utilization.

Algorithm: 1

The proposed algorithm uses historical scheduling records (HSR) to efficiently distribute incoming requests to available VMs. The algorithm first selects the minimum loaded VM and for that, it uses historical scheduling record information. After that algorithm checks for whether the selected VM is previously allocated in the last iteration or not? If the selected VM is not previously allocated VM then the proposed scheduler checks for resource requirements. If the resource requirement of the task is less than available resources on the selected VM then the scheduler allocates the request to the selected VM else, it selects the VM with maximum processing power (MIPS). If maximum processing power VM is also not able to handle incoming requests then the scheduler goes for a scaleup algorithm and scaleup the resources of VM and allocates the request to VM. In the end, the scheduler will update HSR and run the consolidation algorithm to deallocate ideal resources.

EDS ()

Input:

1. Task T_i
2. Set of Hosts H
3. Set of VMs V

```
selectedVm = minimumLoadedVm () ##Based on HSR
    If (selectedVm^r > T_i^r) do:
      scheduledVM = selectedVm
    End if
    Else do:
      selectedVm = maximumAvailresourceVm ()
      If (selectedVm^r < T_i^r) do:
         SCALEUPRESOURCES ()
      End if
      scheduledVm = selectedVm
    End else
    Schedule T_i to scheduledVm
    Update HSR
    Mark VM Busy
    CONSOLIDATION ()
    End Procedure
```

Algorithm: 2

This algorithm scales up computing resources to accommodate high request flow. Scaleup can be achieved by creating new VMs or by adding more processing capabilities to VMs.

SCALEUPRESOURCES ()

Input:

1. Set of Hosts H
2. Set of VMs V
3. HSR: Historical Scheduling Record

Foreach VM_i in V do:
  If ($VM_i^r > T_i^r$) do:
    selectedVm = VM_i
    Break
  End if
End For
If (selectedVm == null) do:
  selectedVm = create new VM and link to Host
  Mark Host as H^{active}
  Update HSR
End if
End Procedure

Algorithm: 3

This algorithm scales down computing resources if not in use for a long time. This task can be accomplished either by deleting VMs or switching off hosts.

CONSOLIDATION ()

Input:

1. Set of Hosts H
2. Set of VMs V
3. HSR: Historical Scheduling Record

Foreach VM_i in V do:
  If ($t_{VMi}^{ideal} > t^{ideal}$) do:
    Delete VM
  End if
End For
Foreach Host in H^{ideal} do:
  Power off Host
End For

9. End Procedure

Algorithm: 2

This algorithm scales up computing resources to accommodate high request flow. Scaleup can be achieved by creating new VMs or by adding more processing capabilities to VMs.

SCALEUPRESOURCES ()

Input:

1. Set of Hosts H
2. Set of VMs V
3. HSR: Historical Scheduling Record

1. Foreach VM_i in V do:
2. If ($VM_i^r > T_i^r$) do:
3. selectedVm = VM_i
4. Break
5. End if
6. End For
7. If (selectedVm == null) do:
8. selectedVm = create new VM and link to Host
9. Mark Host as H^{active}
10. Update HSR
11. End if
12 End Procedure

Algorithm: 3

This algorithm scales down computing resources if not in use for a long time. This task can be accomplished either by deleting VMs or switching off hosts.

CONSOLIDATION ()

Input:

1. Set of Hosts H
2. Set of VMs V
3. HSR: Historical Scheduling Record

1. Foreach VM_i in V do:
2. If ($t_{VMi}^{ideal} > t^{ideal}$) do:
3. Delete VM
4. End if
5. End For
6. Foreach Host in H^{ideal} do:
7. Power off Host

8. End For
9. End Procedure

The proposed scheduling algorithm schedules real-time incoming tasks to resources so it doesn't assume anything about task property priory. The main algorithm consists of four different methods:

1. **minimumLoadedVm():** From the historical scheduling record (HSR), this method searches for the minimal loaded virtual machine from among the available virtual machines.

Here in this method, two conditions are possible based on the cloudsim properties:

A. Similar Tasks: It should then be possible to pick the VM with the least number of allocations for tasks of the same type and that possess similar properties to the minimum loaded VM.
B. Different Tasks: Many factors can be considered when computing the minimum loaded VM when tasks are of different properties. For example, RAM utilization, Storage Utilization, energy consumption, etc.

2. **maximumAvailresourceVm():** This method calculates the minimum utilized VM based on the HSR file, similar to the minimumLoadedVm() method.
3. **SCALEUPRESOURCES():** To scale up computing resources.
4. **CONSOLIDATION():** To scale down computing resources.

IV. PERFORMANCE EVALUATION

It is hard to test the performance of this application using real testbeds because some of the elements cannot be controlled and predicted by system designers/developers. To control the performance of this application, we can use tools such as WorkflowSim, CloudAnalyst, CloudSim, etc. That provides repeatability to experiments. Here proposed algorithm is implemented in the WorkflowSim simulator.
TABLE I shows the characteristics of the resource used. There is one data center that contains 25 hosts in it.

TABLE III
CHARACTERISTICS OF RESOURCES

Datacenter	
Number of Datacenters	1
Number of hosts	25
VM (Virtual Machine)	
Number of VMs	100
MIPS of PE per VM	1000-3000 MIPS
VM memory	512
Bandwidth	1000
Type of manager	Space shared

The Figure below shows the comparison of Max-Min, SJFLJF, and the proposed scheduling algorithms. Based on the results of simulations, it appears that the proposed algorithm is more efficient than the SJFLJF algorithm. The algorithm tries to minimize underloading and overloading of VMs and enhance the response time of virtual machines.

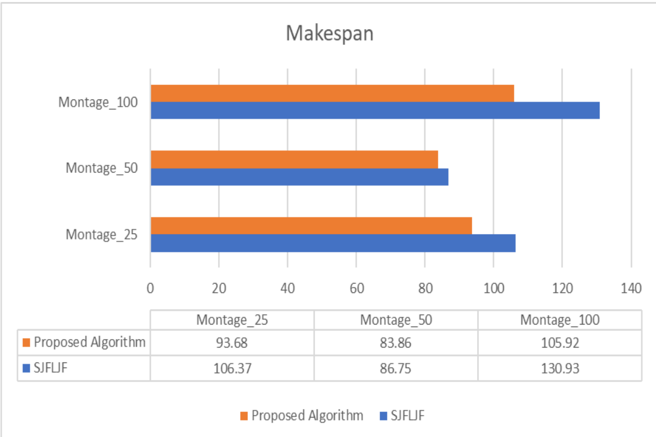

Figure 1: Makespan analysis

The image below shows the cost of processing different input files using SJFLJF and the proposed algorithm. The processing cost of the proposed scheduling algorithm is lover than that of the SJFLJF algorithm, which makes it more suitable for processing large tasks.

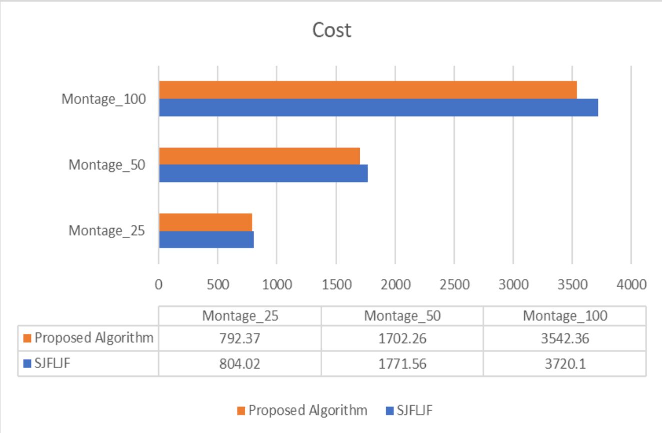

Figure 2: Cost analysis

V. FUTURE WORK

The research article presents a comparative analysis of various existing scheduling and load balancing algorithms. Many cloud simulation tools are available in the market that enables users to simulate real-world cloud scenarios such as energy-efficient scheduling, load balancing, virtual machine migration, etc. In the future analysis of various cloud simulation tools can be done to present the most suitable tool for various simulation applications. Advantages and disadvantages of these tools can be presented that help researchers in sleeting the most appropriate tool for their research and development work.

The proposed algorithm can be enhanced by integrating priority tasks. Also, comparison can be done based on multiple parameters such as processing cost, Response time, average waiting time, energy consumption, etc.

VI.CONCLUSION

An increase in the number of users in the system raised many challenges for cloud service providers. Scheduling all user requests at a rapid rate and ensuring service level agreement and QoS availability becomes the most challenging task. Scheduling is not only about mapping user requests to resources but also comprises managing resource utilization, energy consumption, load balancing, data center processing, and many more. In this research article, a comparative analysis of various scheduling and load balancing algorithms is presented. Multiple simulation tools can be used to simulate real-world cloud computing applications. The research article also highlights simulation tools used with the pros and cons of each scheduling approach. Also working of all scheduling algorithms is presented here with parameters considered for simulation and performance evaluation. Scheduling algorithms not only schedule resources but also help in minimizing energy consumption by providing efficient distribution of incoming tasks to available virtual machines. The Proposed algorithm is a dynamic energy-aware scheduling algorithm that tries to minimize the energy consumption of cloud systems and enhance the makespan/response time of the system. The proposed algorithm supports dynamic scale-up and scales down of resources which helps in enhancing resource utilization.

VII. REFERENCES

[1]. Rai, H., Ojha, S. K., & Nazarov, A. (2020, December). Cloud Load Balancing Algorithm. In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 861-865). IEEE.

[2]. Kumar, P., Bundele, M., & Somwansi, D. (2018, November). An adaptive approach for load balancing in cloud computing using MTB load balancing. In 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE) (pp. 1-5). IEEE.

[3]. Mishra, A., & Tiwari, D. (2020, December). A Proficient Load Balancing Using Priority Algorithm In Cloud Computing. In 2020 IEEE International Conference on Machine Learning

and Applied Network Technologies (ICMLANT) (pp. 1-6). IEEE

[4]. Panda, S. K., & Jana, P. K. (2019). An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems. Cluster Computing, 22(2), 509-527.

[5]. Malhotra, D. (2018, December). LD_ASG: Load Balancing Algorithm in Cloud Computing. In 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 387-392). IEEE.

[6]. Tang, L. (2018, August). Load Balancing Optimization in Cloud Computing Based on Task Scheduling. In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS) (pp. 116-120). IEEE.

[7]. Swarnakar, S., Kumar, R., Krishn, S., & Banerjee, C. (2020, September). Improved Dynamic Load Balancing Approach in Cloud Computing. In 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE) (pp. 195-199). IEEE.

[8]. Mishra, A., & Tiwari, D. (2020, December). A Proficient Load Balancing Using Priority Algorithm In Cloud Computing. In 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT) (pp. 1-6). IEEE.

[9]. Alworafi, M. A., Dhari, A., El-Booz, S. A., Nasr, A. A., Arpitha, A., & Mallappa, S. (2019). An enhanced task scheduling in cloud computing based on hybrid approach. In Data Analytics and Learning (pp. 11-25). Springer, Singapore.

[10]. Marahatta, A., Pirbhulal, S., Zhang, F., Parizi, R. M., Choo, K. K. R., & Liu, Z. (2019). Classification-based and energy-efficient dynamic task scheduling scheme for virtualized cloud data center. IEEE Transactions on Cloud Computing, 9(4), 1376-1390.

[11]. Singh, A. N., & Prakash, S. (2018). WAMLB: weighted active monitoring load balancing in cloud computing. In Big data analytics (pp. 677-685). Springer, Singapore.

[12]. Mondal, A. S., Mukhopadhyay, S., Mondal, K. C., & Chattopadhyay, S. (2021). A Double Threshold-Based Power-Aware Honey Bee Cloud Load Balancing Algorithm. SN Computer Science, 2(5), 1-16.

[13]. Seth, S., & Singh, N. (2019). Dynamic heterogeneous shortest job first (DHSJF): a task scheduling approach for heterogeneous cloud computing systems. International Journal of Information Technology, 11(4), 653-657.

[14]. Yadav, M., & Prasad, J. S. (2019, April). An Enhanced Genetic Virtual Machine Load Balancing Algorithm for Data Center. In International Conference on Advances in Computing and Data Sciences (pp. 244-253). Springer, Singapore.

[15]. Mulla, B., Krishna, C. R., & Tickoo, R. K. (2019, August). Virtual Machine Allocation in Heterogeneous Cloud for Load Balancing Based on Virtual Machine Classification. In International Conference on Inventive Computation Technologies (pp. 331-341). Springer, Cham.

[16]. Zhang, Y., Cheng, X., Chen, L., & Shen, H. (2018). Energy-efficient tasks scheduling heuristics with multi-constraints in virtualized clouds. Journal of Grid Computing, 16(3), 459-475.

[17]. Kashikolaei, S. M. G., Hosseinabadi, A. A. R., Saemi, B., Shareh, M. B., Sangaiah, A. K., & Bian, G. B. (2020). An enhancement of task scheduling in cloud computing based on imperialist competitive algorithm and firefly algorithm. The Journal of Supercomputing, 76(8), 6302-6329.

Cite this article as :

Priyal Ghetiya, Prof. Dhaval Nimavat, "Enhancing Response Time of Cloud Resources Through Energy Efficient Cloud Scheduling Algorithm", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 9 Issue 3, pp. 354-356, May-June 2022. Available at doi : https://doi.org/10.32628/IJSRSET222934

Journal URL : https://ijsrset.com/IJSRSET222934