Green Task Scheduling Algorithm in Green-Cloud

Mysha Maliha Priyanka[1], K. M. Safin Kamal[1], Md. Hasibur Rahman[1], Ayesha Siddiqa Yasmin[1], *Ahmed Wasif Reza[1],

[1]Department of Computer Science and Engineering, East West University, Dhaka-1212, Bangladesh

Emails: malihapriyanka81@gmail.com,kmsafinkamal@gmail.com, mdhasibur.rh@gmail.com, ayeshayasmin890@gmail.com, wasif@ewubd.edu

Abstract. Being eco-friendly helps cloud-dedicated servers meet green computing requirements. Task scheduling is one of the major concerns in cloud-based solutions that must be taken into account for improving system effectiveness and the user experience. The phrase "green cloud computing" describes how to effectively employ information technology and other technical breakthroughs for the environment's benefit. The prime goal of this paper is to find an algorithm that focuses mostly on decreasing execution times of green cloud computing and is efficient for the environment. We compared some task scheduling algorithms contingent on the execution time, like FCFS, SJF, and Round Robin, with the algorithm we advised, the generalized priority (GP) algorithm. Using the CloudSim 3.0.3 simulator, we performed an experiment to validate our approach. The algorithm we suggested has the shortest runtime out of all the algorithms which is 97.91.

Keywords: Cloud computing, virtual machine (VM), task scheduling algorithm, CloudSim 3.0.3 simulator.

1 Introduction

The term "cloud computing" specifically refers to everything that involves hosting services online. There are three type of cloud services like IaaS, PaaS and SaaS [1]. For clients, the services are pay as you go or available as needed. These services, which are mostly global and diversified, can be deployed and released quickly with hardly any management work or managed service involvement. A cloud - based data center with additional servers is typically deployed. When it comes to overall expense and usage, cloud computing has a number of benefits. The cloud service provider manages values provided by data users [2]. Making use of technology innovations like computing as well as other IT services in a sustainable way for significant environmental advantages is known as "green cloud computing."

A computer program's ability to do work in a particular order is controlled by a set of standards known as scheduling [3]. Among all the algorithms for scheduling,

there is task scheduling algorithm. The primary benefit of the task scheduling technique is the optimum system efficiency and high computing performance. We can employ task scheduling methods to cut costs, save time. In this paper, the most important facts are,

I. We have proposed an algorithm which is called generalized priority (GP) algorithm.
II. We have compared some algorithms like FCFS, RR, SJF algorithm and GP (generalized priority) algorithm to find the most efficient algorithm which minimizes the time.

The content of this paper is as described in the following: in section 1 the basic introduction of the cloud computing and task scheduling algorithm is given. Then, the section 2 is about the related work in the same topic. After that in section 3, research methodology were discussed. The results and discussion is in the 4th section. At last in the section 5, the conclusion of this paper is included.

2 Related Work

In reference [4], the algorithms which have been used are "Green cloud scheduling algorithm" and compared with some other algorithms for task schedule such as HEROS scheduler, Random scheduler and round robin algorithm. Rao, G.J. and Babu, G.S. used Green cloud simulator to compare the algorithms. The main intention of the paper was to reduce carbon emission and energy efficiency.

In 2016, Maharana, D., Sahoo, D.B. and Sethi, D.S. suggested EDVS / dynamic energy efficient real-time tasks scheduling which was assessed by using CloudSim toolkit. [7] The primary goals of the paper are to shrink energy usage in data centers which are cloud based and enhance system performance.

Ismail, L. and Materwala, H. in 2018, [1] proposed a task scheduling algorithm which is called "EATSVM (Energy-aware task scheduling on Cloud Virtual Machines)". The paper is formed a cloud-based computing system with five different types of virtual machines which optimizes the energy in a heterogeneous cloud method. The research claimed that "EATSVM algorithm" conserves more energy than their compered "ECTC algorithm".

In 2022, Mansouri, Najme and Ghafari, R. proposed an algorithm which is Cost-Efficient Task Scheduling Algorithm (CETSA) [3]. The primary goal was to reduce makespan, energy expenditure and cost in cloud computing. The algorithms MSDE, CPSO, CJS, and FUGE were compared with the CETSA algorithm using the CloudSim simulator.

In reference [2], Kak, S.M.K.M., Agarwal, P.A. and Alam, M.A. summarized the relevant work on the methods of the energy-efficient task scheduling methods that are now in use. In order to attain the goal of decreased energy usage and CO2 emission in a cloud environment, they have presented a model. The machine

learning technique have been used in this paper. The objective of this paper was to reduce makespan and energy consumption, maximize resource utilization and minimize the execution time. The model will eventually be implemented within MATLAB and would be evaluated on a number of factors, including makespan, execution time, and resource use, QoS, and energy usage.

3 Research Methodology

Networking, parallel, distributed, and cloud computing performance has all been significantly impacted by resource allocation and scheduling. For an effective scheduling, assignment and scaling all the given resources in the cloud, several researchers have proposed various algorithms. However here, we are basically discussing four scheduling algorithms.

1. First come first serve
2. Round Robin
3. Shortest job first scheduling and
4. Generalized Priority Algorithm (GPA) which is our proposed algorithm.

3.1 Description of used algorithms:

Here we are showing the description of algorithms and their flowchart and procedure.

- **FCFS (First Come First Serve)**

FCFS is one of the very well-known task scheduling algorithms that chooses the resources for the requests that are incoming for parallel processing. The algorithm, FCFS's full form is First Come First Serve and it works like the same. Which process comes first, the algorithm serves it first, which means schedules it first. It is the one with the shortest wait time [8]. The cloud sim provides with FCFS scheduling techniques and also for its internal task scheduling. The virtual machine (VM) components of a cloud-based datacenter are responsible for the allocation of app-specific VMs to the hosts. However, FCFS has a very slow turnaround and response time [9]. However, the algorithm schedules like the non-preemptive method. For this reason, after finishing a full task, the next task will start. As a result, FCFS may have gotten into a starvation problem. The shortest jobs at the back of the queue have to wait for a long period of time because they have to wait until the front of the queue is completed [10]. In figure 1, we are showing the flowchart of FCFS. Here we can see how the algorithm works.

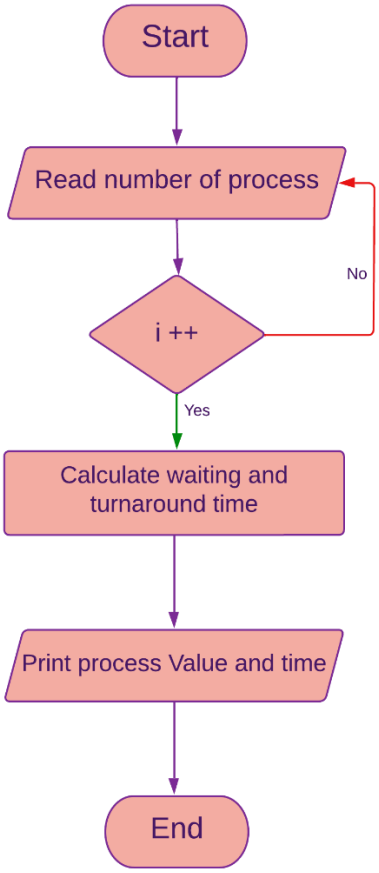

Fig 1. Flowchart of FCFS

- **Round Robin (RR) Algorithm**

The round-robin algorithm is one of the most popular task scheduling algorithms. It gives a starvation-free task schedule and also has a very easy and simple implementation. Round Robin algorithm is named after a well-known principle called 'Round Robin Principle'. Which stands for, turn by turn, each of the persons will get equal weighted share of something. The cloud Sim provides with RR scheduling techniques and also for its internal task scheduling. The virtual machine (VM) components of a cloud-based datacenter are responsible for the allocation of app-specific VMs to the hosts. It performs the time allocation without assigning the priority rather than handles each task in an identical proportion. This algorithm is better than FCFS [6]. In figure 2, we are showing the flowchart of RR. Here we can see how the algorithm works.

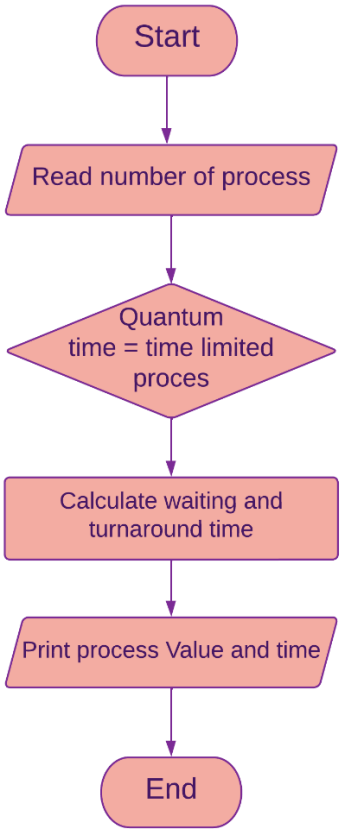

Fig 2. Flowchart of RR

- **SJF (Shortest Job First)**

Shortest Job First, or SJF, is another well-known algorithm. SJF term stands for short job. It works like its title: selecting the waiting request or process with a short execution time means the waiting process with the shortest execution time will be scheduled and executed first. The cloud Sim provides with SJF scheduling techniques and also for its internal task scheduling. The virtual machine (VM) elements of a cloud-based datacenter are responsible for the allocation of app-specific VMs to the hosts. We normally use this algorithm for long schedules. SJF algorithm is of two types [5][10]. It can be worked like preemptive and also can be worked like non preemptive. However, SJF can be a cause of long-term starvation problems as it has a long turn-around (TA) time. Also, it cannot reduce the average turnaround time [13]. In figure 3, we are showing the flowchart. Here we can see how the algorithm works.

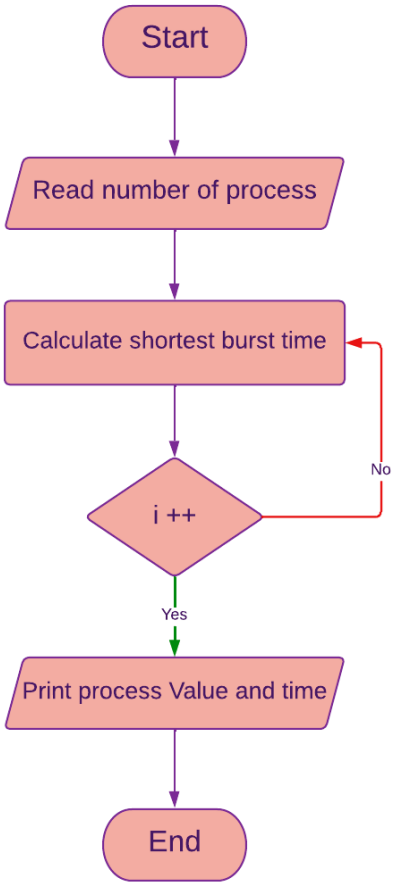

Fig 3. Flowchart of SJF

- **Generalized Priority (GP) algorithm**

Customer establishes the priority in accordance with user demand, and you must provide the cloudlet's parameters such as bandwidth size of the cloudlets, memory, scheduling rules. The given tasks are originally ranked in the suggested method according to their size, with the work with the largest size receiving the top position. The MIPS value of each Virtual Machine is also used to rank (prioritize) them, with the highest MIPS having the highest rank. As a result, the size of the work and the MIPS of the VM are important factors in assigning tasks a priority. In comparison to FCFS and SJF, this approach performs better. This is our proposed algorithm. In the below, we are showing flowchart (fig 4) and algorithm procedure. Here we can see how the algorithm works.

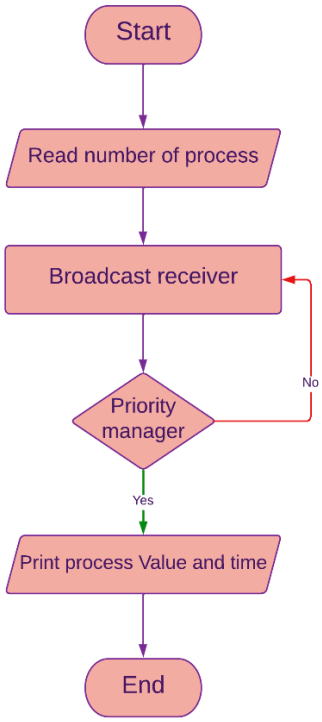

Fig 4. Flowchart of GPA

Algorithm procedure:

1. **Begin**
2. Push first request or node
3. **While Stack is not Empty do**
4. Obtain the unvisited request adjacent next to the Stack top
5. **if** no adjacent no then
6. **If** previous = Top on the stack
7. Insert all contents of stack to VM list
8. **End if**
9. **End if**
10. Pop
11. **if** Stack = Empty then
12. Previous = Top on the stack
13. **else**
14. The node or request is visited
15. Push adjacent requests
16. **End if**
17. **End**

3.2 Simulation environment setup:

For the testing, we have to create an environment for simulation. Here in Figure 5 and 6, we are showing a framework and how task scheduling works.

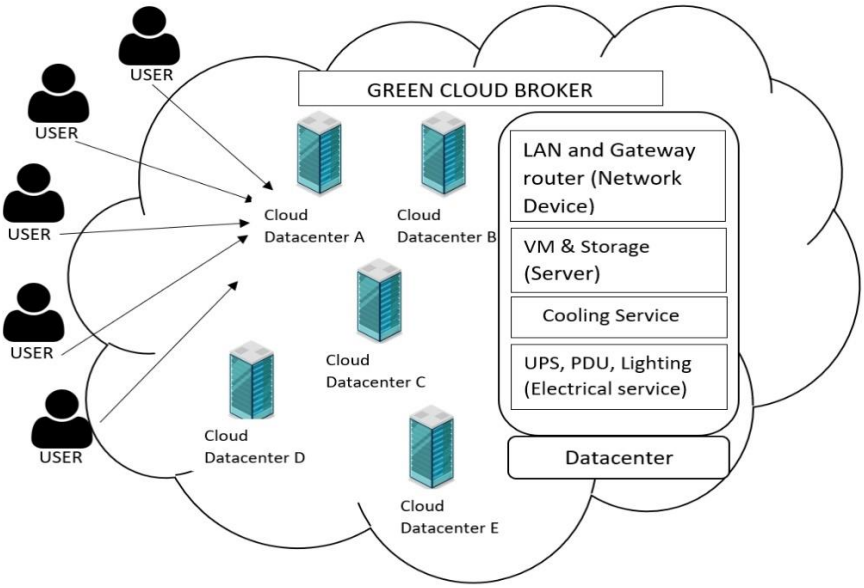

Fig 5. Cloud framework

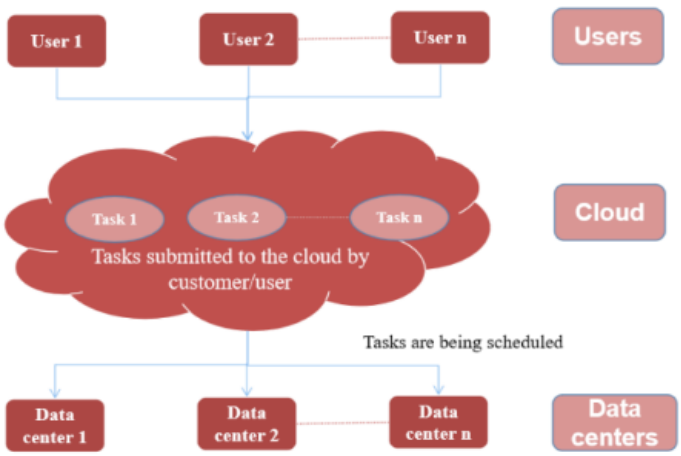

Fig 6. Task scheduling in cloud

Users send their request to the datacenter where there is lot of data servers. When there are a lot of requests, the task scheduler sends them according to its algorithm[15].

Based on the outcomes of the simulation, efficiency of the task scheduling algorithms is evaluated. The CloudSim 3.0.3 simulator was used for the experiment. One of the most well-liked simulators for the cloud datacenter is CloudSim. The performance of different task scheduling algorithm can be compared using this simulator [11]. The following instructions are for operating CloudSim:

- Launch a JAVA IDE and initialize the CloudSim package.
- To organize and distribute resources, establish a data center and broker.
- Create virtual machines (VMs), add them to a list, and then transmit the list to the datacenter broker.
- Create a cloud task set. After creating a cloud task set and including the cloud task cloudlet in it, submit the cloud task set to the broker.
- Start the simulation, and once it is finished, print the results.

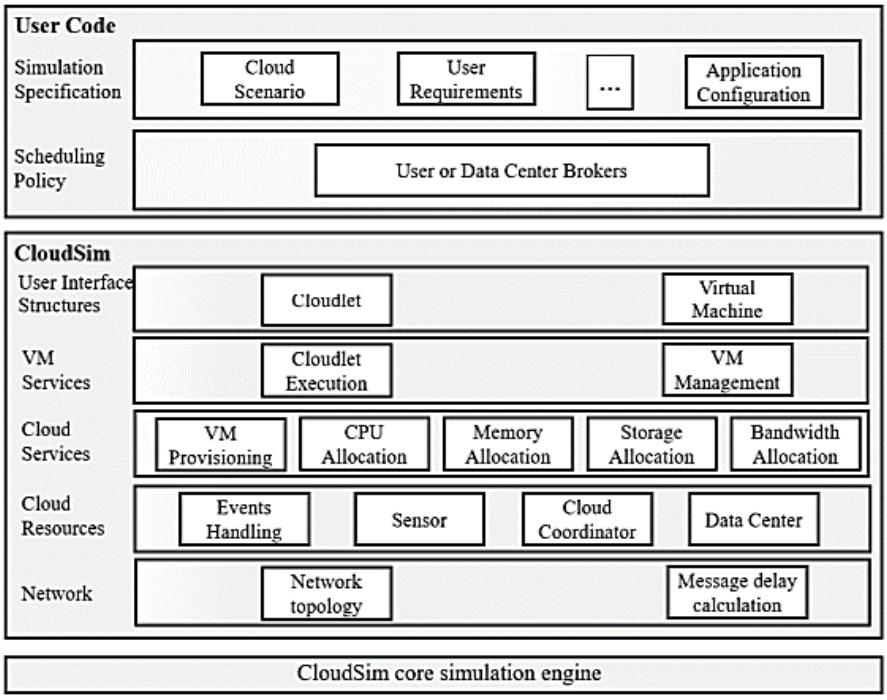

Fig 7. CloudSim structure

There is total three layers in the CloudSim. In the first layer, users insert their codes. Key element execution is controlled by the second layer like VM service, Cloud service, cloud resource and network. Lastly the third layer is the core simulation engine of CloudSim [12][16].

There are other parameters in cloudsim, like the number of VMs, cloudlets, MIPS, and others, that we had to initiate [14]. Table 1 shows the parameters and values we have used for our research in simulation environment for the experiments:

Table 1. Parameter and values for CloudSim environment

Parameters	**Value**
Number of data centers	1
Operating system	Linux
Total number of VMs	100
MIPS of processing element	250
VM memory (RAM)	512MB
Number of CPUs of VM	1
Tasks size of VM	1000MB
Number of Cloudlets	12-100
Length of cloudlets	4000
File size	300
Host memory (MB)	102400 MB
Host storage	1TB
Time zone	10.0
Processing cost	0.3$/Core (hourly)
Memory cost	0.05$/GB (hourly)
Storage cost	0.001$/GB (monthly)

We have set up the simulation environment on our Intel i5 7th Gen 4Core computer with 16GB RAM and an NVIDIA GTX 1060 GPU.

4 Results and Discussion

After setting up CloudSim environment, we simulated the task scheduling algorithm. In the simulation, we have got execution times. In this table 2, we are trying to compare between four algorithms which is FCFS, SJF, Round Robin and Generalized priority algorithm. Here, we used 13 cloudlets and 100 VMs randomly.

Table 2. Output for all algorithm in data center ID 2.

FCFS			SJF			Round Robin Algorithm			Generalized Priority Algorithm		
Cloudlet ID	VM	Time	Cloudlet ID	VM	Time	Cloudlet ID	VM	Time	Cloudlet ID	VM	Time
0	0	41.83	0	0	41.83	0	0	41.80	0	0	32.71
1	1	68.55	1	1	68.55	1	1	67.55	1	1	52.63
2	2	82.2	2	2	79.93	2	2	82.1	2	2	62.22
3	4	92.88	3	4	92.15	3	4	90.88	3	4	71.43
5	3	96.09	5	3	95.31	5	3	91.09	5	3	71.7
4	6	104.8	4	6	103.9	4	6	103.8	7	5	80.25
7	5	110.2	7	5	109.3	7	5	108.2	4	6	80.57
6	8	116.5	6	8	115.6	6	8	111.5	6	8	88.86
9	7	124.6	9	7	123.5	9	7	120.5	9	7	89.23
8	10	128.1	8	11	127.1	8	12	125.1	8	11	97.27
11	9	139.1	11	9	139.1	11	9	138.2	11	9	97.68
10	11	139.4	10	10	137.98	10	10	137.43	10	10	105.48
13	13	150.63	13	12	152.9	13	11	145.63	13	13	105.94

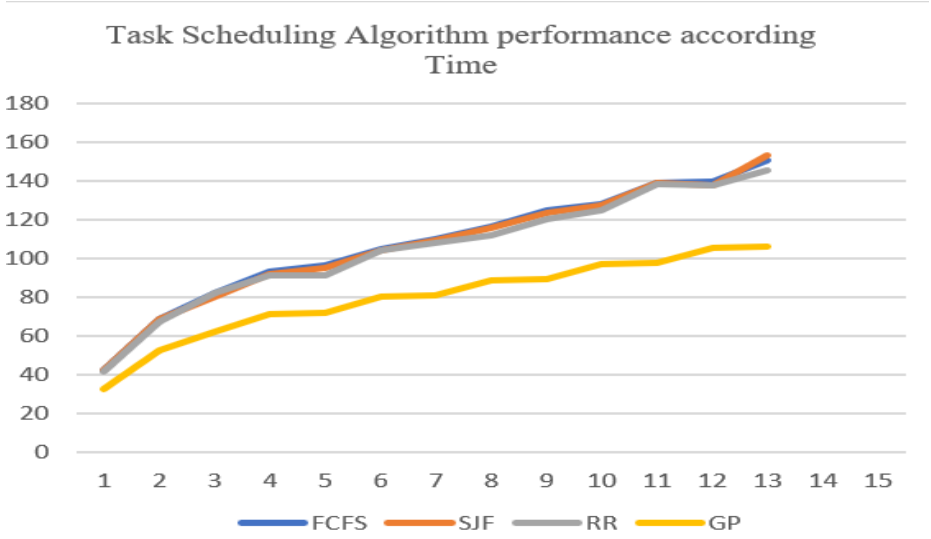

Fig 8. Graph of request number vs. execution time of 4 algorithms

Average Time: In table 3 we are trying to compare the average time of four algorithms which is FCFS, SJF, Round Robin and Generalized priority algorithm.

Table 3. Average time of all algorithms

FCFS	SJF	Round Robin (RR)	Generalized Priority(GP)
131.53	131.98	104.90	97.91

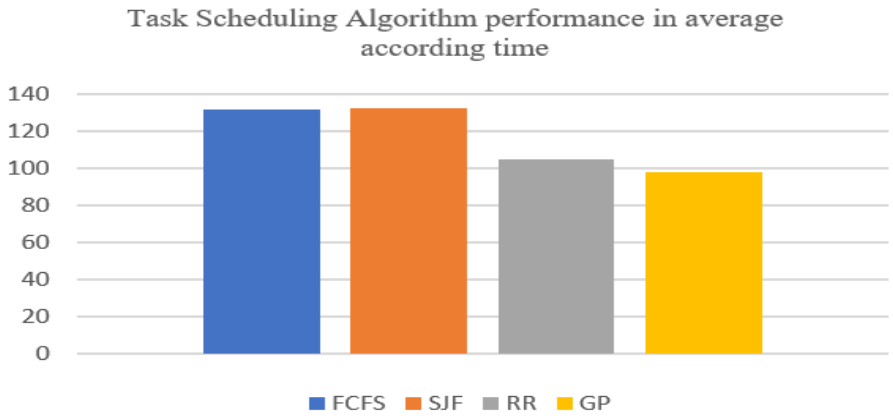

Fig 9. Bar chart of the average times

Here in the task scheduling algorithm's performance according to time, the FCFS algorithm gives different values for different cloudlet IDs and VM IDs. We took 13 cloudlets and 100 VMS to measure the performance. Time started from 41.83s for 1st data and for 13th data its 150.63s. In the case of SJF, Time started from 41.83s for 1st data and for 13th data its 152.91s. For Round Robin Time started from 41.80s for 1st data and for 13th data its 145.63s. Lastly our proposed one GP algorithm Time started from 32.71s for 1st data and for 13th data its 105.94 s. So, all in all, if we see the graphs (Figs. 8 and 9) and analyze them, our generalized priority algorithm works better. It takes less time, so its energy consumption is lower than others. If we consider the average time of these four algorithms, we get better results for the Generalized Priority algorithm.

5 Conclusion

One of the most crucial duties in a green cloud computing environment is scheduling. When task taking less time to execute it will helps to carbon emission less which will affect our environment for making green. Various scheduling

algorithms that effectively plan out computational work in a green-cloud setting have been examined in this research. We tested the FCFS, SJF, and round-robin scheduling algorithms, and we also suggested the GPA (generalized priority algorithm) as a novel scheduling technique. In cloud systems, priority is a key factor in job scheduling. The experiment is run with various workload traces and virtual machine counts. Round Robin, SJF, and FCFS are contrasted with the experiment that was undertaken. The outcome demonstrates the suggested algorithm is much more effective than FCFS, SJF, and Round Robin. In this paper, we primarily analyze four algorithms. We proposed an algorithm that is a generalized priority task scheduling algorithm using a small number of tasks. We have a target on future that we will add some more tasks and attempts like ML, AI, deep learning to make shorter the time that need for execution. We also propose this algorithm for a big grid eco system and will compare the times that need for execution between the grid and the green cloud.

References

[1] Ismail, L. and Materwala, H. (2018) "EATSVM: Energy-aware task scheduling on Cloud Virtual Machines," Procedia Computer Science, 135, pp. 248–258. Available at: https://doi.org/10.1016/j.procs.2018.08.172.

[2] Kak, S.M., Agarwal, P. and Alam, M.A. (2022) "Task scheduling techniques for energy efficiency in the cloud," EAI Endorsed Transactions on Energy Web, 9(39). https://doi.org/10.4108/ew.v9i39.1509.

[3] Mansouri, Najme and Ghafari, R. (2022) "Cost-Efficient Task Scheduling Algorithm for Reducing Energy Consumption and Makespan of Cloud Computing," 5, pp. 1–10. Available at: https://doi.org/10.22067/CKE.2022.70000.1008.

[4] Rao, G.J. and Babu, G.S. (2017) "Energy Analysis of task scheduling algorithms in Green Cloud," *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* [Preprint]. Available at: https://doi.org/10.1109/icimia.2017.7975624.

[5] Greenberg, S., Mills, E. and Tschudi, B. (no date) "Best Practices for Data Centers: Lessons Learned from Benchmarking 22 Data Centers," SUMMER, pp. 76–87

[6] Ismail, L. and Fardoun, A.A. (2017) "Towards energy-aware task scheduling (EATS) framework for divisible-load applications in cloud computing infrastructure," *2017 Annual IEEE International Systems Conference (SysCon)* https://doi.org/10.1109/syscon.2017.7934791.

[7] Maharana, D., Sahoo, D.B. and Sethi, D.S. (no date) "Energy-Efficient Real-Time Tasks Scheduling in Cloud Data Centers," pp. 768–773. https://doi.org/https://core.ac.uk/download/pdf/235196825.pdf.

[8] Bhat, R.V. *et al.* (2020) "Survey on Energy and Cost Optimization Techniques using Green Cloud computing," XVI (V), pp. 337–344.

[9] Siahaan, A.P. (2017) "Comparison analysis of CPU scheduling FCFS, SJF and round robin." Available at: https://doi.org/10.31227/osf.io/6dq3p.

[10] Mhedheb, Y. and Streit, A. (2016) "Energy-efficient task scheduling in Data Centers," *Proceedings of the 6th International Conference on Cloud Computing and Services Science* [Preprint]. Available at: https://doi.org/10.5220/0005880802730282.

[11] KRA, Y.A.O.F.R.A.N.C.O.I.S.M.I.C.H.A.E.L. *et al.* (2019) *Energy Efficiency AndScheduling Techniques Using Cloud Computing Methods Within Data Centers*, 97.

[12] S. R. Jena, R. Shanmugam, K. Saini, and S. Kumar., "Cloud computing tools: inside views and analysis", Procedia Computer Science, vol. 173, pp. 382–391, 2020. (Between cloud)

[13] Maharana, D., Sahoo, D.B. and Sethi, D.S. (2016) *Energy-Efficient Real-Time TasksScheduling in Cloud Data Centers*, pp. 768–773.

[14] Atiewi, S., Yussof, S. and Ezanee, M. (2015) "A comparative analysis of task schedulingalgorithms of Virtual Machines in cloud environment," *Journal of Computer Science*,11(6), pp. 804–812. Available at: https://doi.org/10.3844/jcssp.2015.804.812.

[15] Gade, A. (2018) "Survey on energy efficient cloud: A novel approach towards green computing," *HELIX*, 8(5), pp.3976–3979. Available at: https://doi.org/10.29042/2018-3976-3979

[16] Hamann, H.F. (2013) "Optimization algorithms for energy-efficient data centers," *Volume 2: Thermal Management; Data Centers and Energy Efficient Electronic Systems* [Preprint]. Available at: https://doi.org/10.1115/ipack2013-73066.