International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.7096
***Article Received**: 21 March 2023 **Revised**: 05 May 2023 **Accepted**: 28 May 2023*

EETSQ: Energy Efficient Task Scheduling based on QoS Parameters in Cloud Computing Environment

Sanjiv Kumar Grewal[1], Dr. Neeraj Mangla[2]
[1]Research Scholar, Department of CSE
MMEC,Maharishi Markandeswar (Deemed to be University)
Mullana, Ambala, India
grewalsanjiv34@gmail.com
[2]Associate Professor, Department of CSE
MMEC,Maharishi Markandeswar (Deemed to be University)
Mullana, Ambala, India
neerajmangla@mmumullana.org

Abstract- Now a day, energy consumption is the big challenge in heterogeneous cloud computing environment that needs to be considered. Cloud service provider also needs to satisfy customer's Quality of Service (QoS) for better utilization. An energy efficient task scheduling based on QoS parameter has been proposed to address above said challenge. Firsty, all the incoming tasks are categorized into four classes based on some special attributes and prioritize according to importance of the classes. Secondly, Physical Machines (PMs) type confirmation list is selected based on the number of resource blocks and then select one PM that has maximum QoS value. All the Virtual Machines (VMs) on selected PM are prioritized according to their weight. Experimental evaluation done on CloudSim shows the effectiveness and efficiency of proposed approach.

Keywords-Energy efficient scheduling, task priority, QoS parameters, load balancing.

I. INTRODUCTION

Cloud computing is an act of storing, processing and managing user data on remote servers hosted on internet rather than local personal computers. Scheduling of incoming task is done with available resources on the network that will help in execution of the task [1]. In shortest job first (SJF) scheduling algorithm, shortest task are always scheduled first and then followed by long tasks whereas in min-min scheduling algorithm short tasks are always scheduled in parallel manner and long tasks have to wait to be scheduled later after short tasks. Sometimes, the problem of starvation may arise due to long waiting time for long tasks which degrades the performance [2]. There is a need to apply QoS parameters for tasks scheduling and physical machines (PMs) scheduling as per user requirement to make system more efficient. Virtual machines (VMs) are also need to be scheduled on the PM. Priorities of the incoming tasks and PMs are evaluated by comparing the QoS parameters of each task and PMs respectively. First task in the priority list is scheduled on first PM in PM confirmation list and then execute on the VM available in the VM priority list [3][4].

QoS is the overall effort of service performance, in which user satisfaction is determined for the available services. QoS for cloud service provider is the degree of tasks meeting their requirement efficiently. QoS parameters for the task scheduling includes execution time, waiting time, user type, reliability etc and QoS parameters for PM includes response time, throughput, system security, reliability etc [5][6]. Huge amount of resources are needed to fulfil QoS requirements of a user. Energy consumption is dependent on the number of resource consumed. Energy consumption is increasing day by day due to ever increasing resources in cloud infrastructure. High energy consumption leads to high operational cost in cloud environment and it also pollute the environment due to emission of CO_2. In 2021 around 75 billion kWh energy was consumed by cloud data centers globally [7]. Therefore, it is necessary to develop energy-efficient scheduling in cloud. However, reduction of energy consumption is still a challenge in cloud. Considering this issue, QoS aware energy-efficient task scheduling has been proposed.

Contribution of this paper is threefold. Firstly, we develop an energy efficient model based on QoS parameters. Incoming tasks are classified into four classes based on the QoS parameters i.e task user type, task priority, task execution time and latency task. These four classes are named as: 1. Urgent user class, 2. Urgent task class, 3. Long task class 4. Normal task class. Significance of 1st class is highest followed by 2nd, 3rd, and 4th class respectively. Secondly, requests are assigned to most suitable PM in the PM confirmation list and then execute on the first suitable VM found in the VM priority list. PM confirmation list are categorized based on no of resources (such as, CPU, RAM, disk etc.) needed by the incoming task. QoS of all PMs are calculated based on their response time and throughput. PM with maximum QoS value has highest priority and so on. Weights of VMs are also considered for evaluating the task requests efficiently. Number of processor; transfer speed, memory unit, processing speed; transfer speed are considered for VMs weight calculation.

Energy consumption is dependent on the number of PMs in running condition [8]. In our paper, focus is mainly on the task scheduling and PM scheduling. Energy consumption and running time of PMs are directly proportional and running time is dependent on the throughput and response time. High running time could result in high energy consumption [9]. Our focus is also on reducing the running time of PMs so that energy can be consumed.

Rest of the paper is described as follows: next section 2 describes the literature related to the current work, section 3 describes QoS aware energy efficient scheduling, section 4 includes task allocation on selected PM and section 5 includes experiment evaluation and result exploration. Conclusion and future work includes in last section of this paper.

II. RELATED WORK

Liang Hao et. al. developed a task tolerant energy consumption optimization algorithm in cloud environment. Waiting time of tasks could be reduced by increasing the execution level [10].

Energy efficient cloud data servers have received a huge concern from academia and industry. Energy aware task scheduling decreased the energy consumption in cloud environment. Some performance metrics i.e task tolerance, load balance etc. are used for scheduling the task in distributed computing [11].

Shamita et. al. introduced an energy-efficient framework that takes emission reduction and energy saving into account. Energy consumption is a big issue now a day, and some researchers have developed the models based on QoS and energy consumption. There is some mathematical relationship between user satisfaction by QoS parameters and energy consumption in cloud centers. Types of incoming tasks can be different types such as storage intensive, compute intensive tasks etc. [12], therefore there is need to allocate appropriate physical machine based on some QoS parameters to reduce the energy consumption and increase resource utilization. Power Usage Effectiveness (PUE) can also be considered for more practical results [13].

Malik et. al. presented two types of dynamic optimization algorithms for QoS enhanced energy consumption in data center. In proposed techniques, energy-consumption is achieved by VM migration technique [14]. A. paya et. al presented an energy-efficient technique for load balancing of resources. Proposed technique converts the idle PMs into sleep mode to save energy [15]. Sanjeev et. al. used ‘Dynamic Voltage and Frequency Scaling’ (DVFS) for energy saving in scheduling technique. In proposed technique, researchers also consider machine heterogeneity for better utilization of resources [16].

Arroba et. al. proposed an enhanced version of the scheduling work that refined the QoS parameters, limit the overhead of preemption of tasks and also consider the heterogeneous infrastructures in cloud environment. Proposed scheduler is energy efficient and also cost effective [17].

III.ENERGY EFFICIENT MODEL BASED ON QOS PARAMETERS

A. Task Description

Let, n, be the no of incoming independent tasks and every task contains four attributes that helps in tasks categorization:

1. Type of users: it provides the detail of the kind of users based on the urgency level.
2. Probable priorities of task: it provides the detail of the probable priority of the task scheduled is low, medium or high.
3. Length of task: it provides the details of normal, lengthy or loaded tasks.
4. Latency Time: provides details of the latency of all tasks.

In proposed algorithm, group based task scheduling is used and proposed algorithm has four Classes:

1. UrgentUser Class: it contains the tasks with user urgency level.
2. UrgentTask Class: it contains the tasks with probable priority level
3. LongTask Class: it contains loaded or lenghty tasks.
4. NormalTask Class: it includes left over tasks.

First UrgentUser class has the highest priority followed by UrgentTask class, LongTask class and NormalTask Class respectively. Tasks in the UrgentUser class are scheduled first and after completion of all the tasks in one class followed by other classes in the same way on priority bases. Figure 1 shows the flow chart for task selection.

- *Proposed EETSQ Algorithm for task selection:-*

```
1. Get n number of tasks
2. Get four QoS attributes i.e UT, TP, TL, WT of each task.
3. Classify tasks into four classes i.e Urgent User, Urgent Task, Long Task and normal task according to their QoS parameter.
4. If Urgent User Class has any task then
5. Task with minimum completion time of this class assign to PM
6. Else if Urgent Task Class has any task then
7. Task with minimum completion time of this class assign to PM
8. Else if Long Task Class has any task then
9. Task with minimum completion time of this class assign to PM
10. Else if Normal Task Class has any task then
11. Task with minimum completion time of this class assign to PM
12. End If
```

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.7096
***Article Received**: 21 March 2023 **Revised**: 05 May 2023 **Accepted**: 28 May 2023*

13. After completion of the task, delete that task from the corresponding class and update the task list.

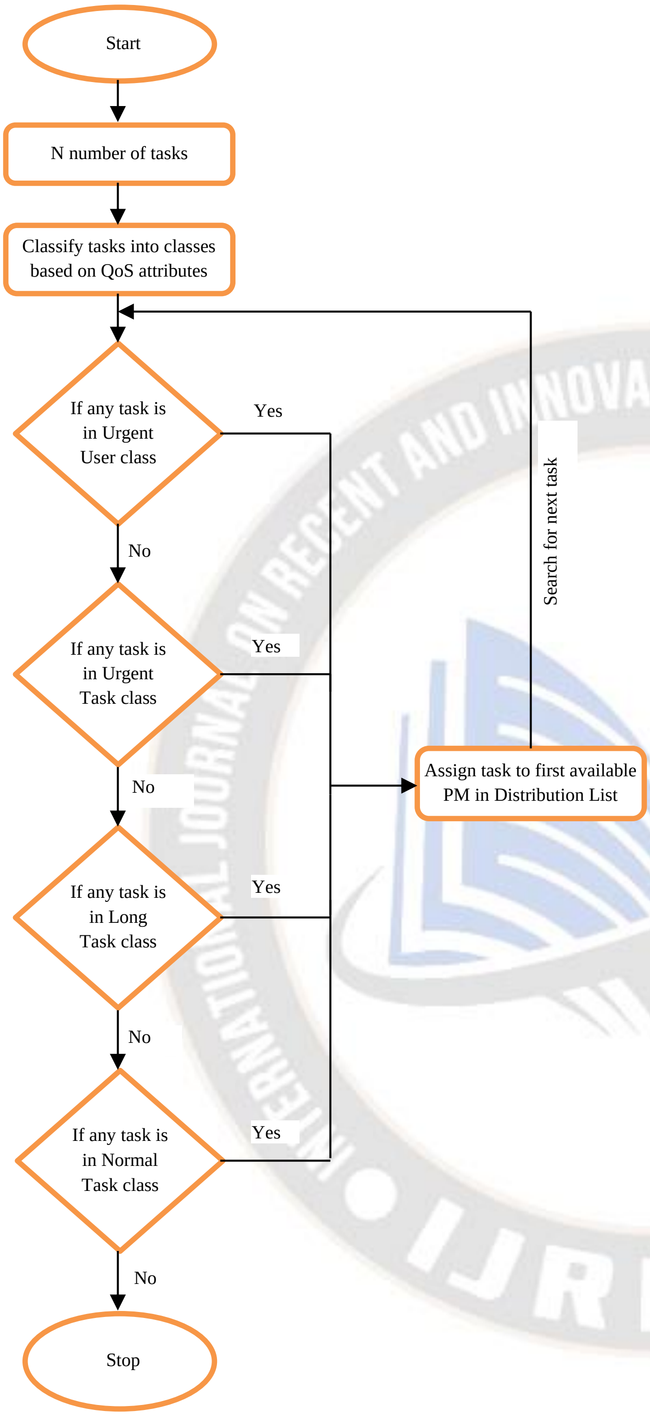
Start
N number of tasks
Classify tasks into classes based on QoS attributes
If any task is in Urgent User class
Yes
No
If any task is in Urgent Task class
Yes
No
If any task is in Long Task class
Yes
No
If any task is in Normal Task class
Yes
No
Stop
Assign task to first available PM in Distribution List
Search for next task

Figure 1. Flow Chart for task selection

B. *QoS model*

There are huge amount of users in cloud computing environment. Web contains huge number of cloud data centers, and there are lots of heterogeneous PMs on each data center. User submitted the tasks to the network and then there is need to allocate task to suitable PM according to the number of requested resources (CPU, RAM, Memory etc.) [18]. Each PM contains number of VMs to execute the allocated task. If there are too many users submitted the tasks at same time, then there will be problem of waiting time for low priority tasks. Some QoS parameters are needs to be set to satisfy the user requirement efficiently and for effective resources utilization for better energy consumption [19].

QoS parameters for the PMs are throughput, system security, response time, no. of tasks denied, system availability etc. Energy consumption and number of PMs are directly related in running state. If the running time of PM is higher, then energy consumption will be more [20]. In this paper, throughput and response time of PMs are considered as QoS parameters for better quality of service. Two QoS parameters affect the running time of PMs and energy consumption in the system.

Suppose there are N number of PMs in the cloud, such as P M = {PM$_1$, PM$_2$, . . ., PM$_i$, . . ., PM$_N$ }($1 \leq i \leq N$). Q_{rti} and Q_{tpi} is the average response time of all PMs respectively. These could be denoted as:

$$Q_{tpi} = tpi/TP \quad (1)$$

$$Q_{rti} = RT/rti \quad (2)$$

Formula to calculate TP and RT

$$T\,P = \sum_{i=1}^{N} tpi/N \quad (3)$$

$$RT = \sum_{i=1}^{N} rti/N \quad (4)$$

Then, overall QoS value of the ith PM is:

$$Q_i = \alpha \cdot Q_{tpi} + \beta \cdot Q_{rti} + \lambda \quad (5)$$

Here, α, β are the weights of throughput and response time based on SLA agreements respectively. λ is the QoS of PM$_i$ based on other QoS parameters.

C. *Energy consumption model*

In heterogeneous cloud computing, there are lots of different kinds of PM available in one data center. Each PM has different number and different kind of resources. In this paper, PM is divided into 4 types such as: 1. RAM resource intensive PM 2.CPU resource intensive PM 3.Disk resource intensive PM and 4.Cache resource intensive PM. If PM has large number of CPU resources as compared to other resources than it could be type 2 PM i.e. CPU resource intensive PM. If incoming task with the request of more CPU resources allocated to the same type of PM i.e CPU resource intensive PM. Most suitable PM is allocated to each and every task so that all the available resources utilized fully. And by doing this wastage of resources could be reduced.

EC_{total} is total energy consumed by all available resources on PM and it is calculated as given in 6:

$$EC_{total} = E_{others} + p_1.E_{CPU} + p_2.E_{RAM} + p_3.E_{Disk} + p_4.E_{Cache} \quad (6)$$

Here, E_{CPU} is the energy consumed by CPU resources, E_{RAM} is the energy consumed by the RAM resources, E_{Disk} is the energy consumed by disk resources and E_{Cache} is the energy consumed by cache resources. E_{others} is the energy consumed all the left over resources, p_1 to p_4 are the resource weights.

Main objective of the paper is to minimize the total energy consumption and it can be denoted as:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.7096
***Article Received**: 21 March 2023 **Revised**: 05 May 2023 **Accepted**: 28 May 2023*

Min (EC_{total}) = Min (E_{others}) + Min ($p_1.E_{CPU} + p_2.E_{RAM} + p_3.E_{Disk} + p_4.E_{Cache}$) (7)

IV. QOS BASED TASK ALLOCATION ON SELECTED PM

Large amount of energy is consumed while tasks execution and this energy consumption and PMs running time are directly related. Throughput and response time are the two factor that directly affect the running time of the PMs. Small throughput and long response time could result in the more running time of PMs. Thus, these two factors are considered for setting QoS value. Main objective of this paper is to reduce the total energy consumption based on QoS value. Thus, EETSQ technique is proposed for the above said problems. Main method of this paper is divided into 4 parts: i) Confirmation of PM type ii) Detection of available resources iii) Selection of PM iv) Allocation of task.

i) **Confirmation of PM Type:** PM type is confirmed based on the no. of requested resources (i.e CPU, RAM cache etc.).

ii) **Detection of available resources:** Resource utilization on the PM is recorded in time by measuring the resource usage of confirmed PM type.

iii) **Selection of PM:** Throughput and response time of each and every PM is different and it will help in finding QoS value of each PM. After monitoring QoS value of each and every PM, best PM is selected for task allocation.

iv) **Allocation of Task:** This part is totally dependent on the above three parts. After PM selection, task is allocated for the execution on particular VM on the selected PM.

Suppose, there are M number of requests or tasks in cloud centre such as Task = {$task_1$, $task_2$,…..$task_i$,….$task_M$}. $task_i$ is represented as $task_i$=(Task No., Start time, End time, Requested resources). Requested resources can be of different type such as number of CPU, RAM, disk and cache blocks. First task is to confirm PM type according to requested resources by task. Second task is to select most appropriate PM in confirmed PM list and then allocate the task to the PM and execute on available VM. Figure 2 shows the flow chart for task allocation to selected PM.

Proposed EETSQ Algorithm for task allocation to Selected PM: -

```
1.  For PM j= 1 to M
2.      Get requested resource for selected task.
3.      Get PM type confirmation according to requested resources of task.
4.      Calculate the remaining resources of PMs in PM type confirmation list.
5.      While PM have resources do
6.          Put PM in distribution list
7.      End while
8.      Get distribution list of PMs.
9.      Calculate QoS of PMs in distribution list based on response time and throughput.
10.     Arrange PMs in descending order according to their QoS value.
11.     Assign $task_i$ to $PM_j$ with maximum QoS value.
12.     End for
13. End for
14. If no PM in distribution list satisfy the task then
15.     Reject the task
16. End if
```

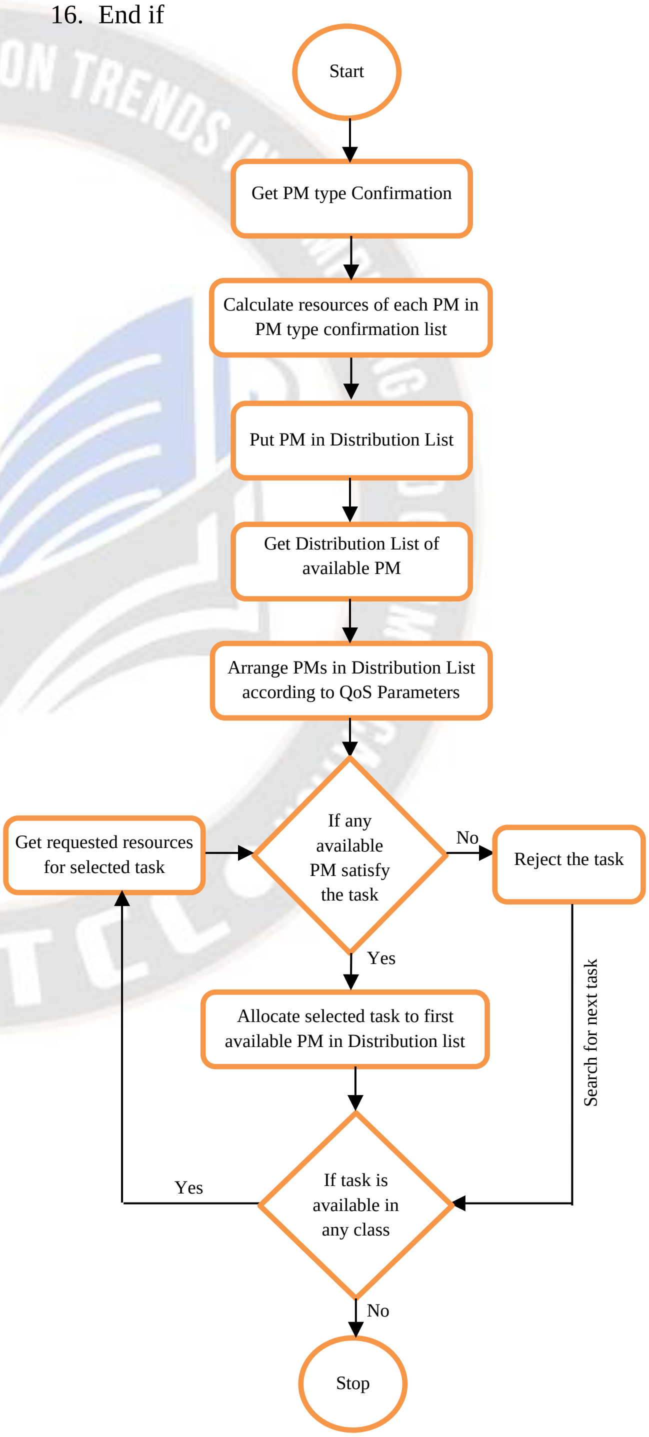
Start
Get PM type Confirmation
Calculate resources of each PM in PM type confirmation list
Put PM in Distribution List
Get Distribution List of available PM
Arrange PMs in Distribution List according to QoS Parameters
Get requested resources for selected task
If any available PM satisfy the task
No
Reject the task
Yes
Allocate selected task to first available PM in Distribution list
Search for next task
Yes
If task is available in any class
No
Stop

Figure 2. Flow chart for task allocation to Selected PM

V. EXPERIMENTAL AND RESULT ANALYSIS

This section includes the evaluation work to measure the performance of proposed algorithm. Parameters used for performance evaluation are such as execution time, latency, response time, throughput, energy consumption and cost etc.

In this paper, 7-tuple data record of task is taken, i.e. task= (018, 3.2, 4.6, 6, 2, 1, and 2). 1st value shows that task is 18th,, 2nd value shows the starting time and 3rd value shows the ending time of a task. 4th, 5th 6th and 7th values show the no of CPU blocks, cache blocks, RAM blocks and disk blocks requested respectively. Similar to task data record, 7-tuple data record of PM is taken i.e PM= (100, 9,8,7,6, 4, and 15). 1st value shows the PM number, 2nd, 3rd, 4th and 5th values show the number of CPU blocks, cache blocks, RAM blocks and disk blocks are used by PM. Last two values show the response time and throughput respectively. Table 1 shows the range of each parameter used in this paper:-

TABLE 1. SIMULATION PARAMETERS

Parameters	**Range**
No. of tasks	[100-2000]
No. of resource blocks of tasks	[1-10]
No. of resource blocks of PM	[1-10]
PM throughput	[10-15]
PM Response time	[1-10]
No. of VMs	[10-100]
MIPS	[500-2000] ms
Bandwidth	[200-1200] kbps
Cost per VM	1$

In this paper, QoS and energy consumption are two most effective parameters to measure the performance of proposed algorithm. QoS is measured on the basis of response time and throughput of the running PM. Six dataset are used to evaluate the proposed EETSQ with existing Greedy and PSO (Particle Swarm Optimization). Proposed algorithm (EETSQ) considers both QoS parameter and type of PM. Existing Greedy algorithm considers only QoS parameter but it does not consider type of PM and PSO algorithm does not consider any above said parameter.

QoS based on response time and throughput is evaluated by using formula 5. Below figure 3 shows the comparison of proposed EETSQ with existing on the basis of QoS and it shows that proposed algorithm has highest QoS value based on the throughput and response time.

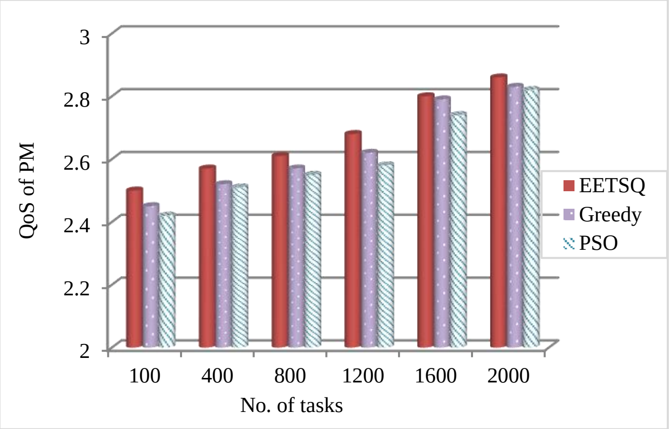

Figure 3. Average QoS values of the running PMs

Energy consumption by running PM is calculated by formula 6. Below figure4 shows the comparison of proposed EETSQ with existing on basis of energy consumption by running PM and it shows that proposed algorithm consume lowest energy as compared to others.

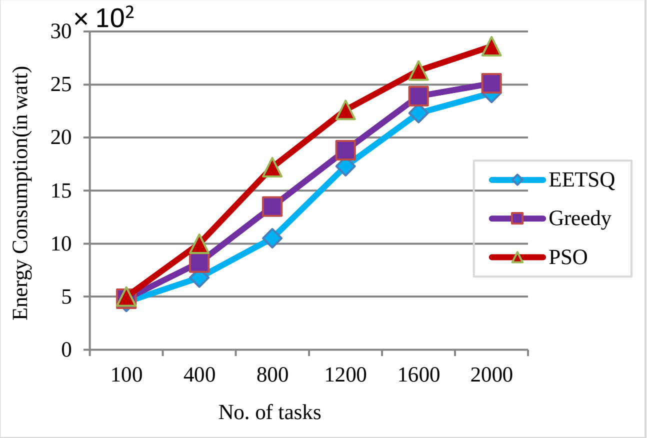

Figure 4. Energy consumption by running PMs

VI. CONCLUSION AND FUTURE WORK

This paper has main focus on reducing the energy consumption by setting QoS parameter on PMs. In proposed EETSQ, incoming task requests has been allocated to the most appropriate PM that has sufficient number of resource blocks. PMs have been prioritized based on the QoS values that are calculated by using throughput and response time. Tasks are executed on the available VM on the selected PM. Proposed technique is very fast and energy efficient.

In EETSQ, only two parameters are used to calculate QoS values but in future more parameters (i.e system security, system availability etc.) can be used. Classification of the available PMs can be more accurate in future.

REFERENCES

[1] Lopes RV, Menascé D. A taxonomy of job scheduling on distributed computing systems. IEEE Trans Parallel Distrib Syst, vol. 27, no. 12, pp. 3412–3428, 2016.

[2] Delimitrou C, Sanchez D, Kozyrakis C. Tarcil: Reconciling scheduling speed and quality in large shared clusters. In: Proceedings of the Sixth ACM Symposium on Cloud Computing SoCC '15. New York: ACM; pp. 97–110, 2015.

[3] Dubey S, Agrawal S. Qos driven task scheduling in cloud computing. Int. J. Comput. Appl. Technol. Res. vol. 2, no. 5, pp. 595–600, 2013.

[4] G. Rajasekaran, & P. Shanmugapriya. (2023). Hybrid Deep Learning and Optimization Algorithm for Breast Cancer Prediction Using Data Mining. International Journal of Intelligent Systems and Applications in Engineering, 11(1s), 14–22. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2472

[5] Potluri S, Rao KS. Simulation of QoS-Based Task Scheduling Policy for Dependent and Independent Tasks in a Cloud Environment. Smart Intelligent Computing and Applications; pp. 515–525, 2019.

[6] Al-Ansi, A. M. . (2021). Applying Information Technology-Based Knowledge Management (KM) Simulation in the Airline Industry . International Journal of New Practices in Management and Engineering, 10(02), 05–09. https://doi.org/10.17762/ijnpme.v10i02.131

[7] Delimitrou C, Kozyrakis C. Paragon: Qos-aware scheduling for heterogeneous datacenters. SIGPLAN Not; vol. 48 no.4, pp. 77–88, 2013.

[8] Ousterhout K, Wendell P, Zaharia M, Stoica I. Sparrow: Distributed, low latency scheduling. In: Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles SOSP '13. New York: ACM; pp. 69–84, 2014.

[9] Sanna Mehraj Ka, Parul Agarwal, and M. Afshar Alam, "Task Scheduling Techniques for Energy Efficiency in the Cloud," EAI Endorsed Transactions on Energy Web, Vol 9,No. 39, 2022.

[10] Linz Tom and Bindu V.R; "DYNAMIC TASK SCHEDULING BASED ON BURST TIME REQUIREMENT FOR CLOUD ENVIRONMENT," IJCNC, Vol 13, No 5, 2021.

[11] Goiri I, Julia F, Nou R, Berral JL, Guitart J, Torres J. Energy-aware scheduling in virtualized datacenters. In: Proceedings of the 2010 IEEE International Conference on Cluster Computing CLUSTER '10. Washington: IEEE Computer Society; pp. 58–67, 2012.

[12] Hao, L., Cui, G., Qu, M., Ke, W.: Resource scheduling optimization algorithm of energy consumption for cloud computing based on task tolerance. J. Softw., vol. 9, no. 4, pp. 895–901, 2014.

[13] Sofia Martinez, Machine Learning-based Fraud Detection in Financial Transactions , Machine Learning Applications Conference Proceedings, Vol 1 2021.

[14] Song, J., Li, T., Wang, Z., Zhu, Z.: Study on energy-consumption regularities of cloud computing systems by a novel evaluation model, vol. 95 no. 4, pp. 269–287, 2013.

[15] Shamita Phutane, Ankith Poojari, Saurabh Nyati and Bhavna Arora: Energy-Efficient Task Scheduling in Cloud Environment, International Research Journal of Engineering and Technology (IRJET), Vol: 09 No: 04 , pp. 2679-2684, Apr 2022.

[16] Tang, Z., Qi, L., Cheng, Z., Li, K., Khan, S.U., Li, K.: An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment. J. Grid Comput., vol. 14, no. 1, pp. 55–74, 2016.

[17] Malik, N.; Sardaraz, M.; Tahir, M.; Shah, B.; Ali, G.; Moreira, F. Energy-Efficient Load Balancing Algorithm for Workflow Scheduling in Cloud Data Centers Using Queuing and Threshold, Appl. Sci. vol. 11, 2021.

[18] Robert Roberts, Daniel Taylor, Juan Herrera, Juan Castro, Mette Christensen. Integrating Virtual Reality and Machine Learning in Education. Kuwait Journal of Machine Learning, 2(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/175

[19] Paya, A., Marinescu, D.: Energy-aware load balancing and application scaling for the cloud ecosystem. IEEE Trans. Cloud Comput., vol. 5, no. 1, pp. 15–27, 2017.

[20] Sanjeevi, P., Viswanathan, P.: Towards energy-aware job consolidation scheduling in cloud. In: International Conference on Inventive Computation Technologies. IEEE 2017.

[21] Arroba, P., Risco-Martín, J.L., Zapater, M., Moya, J.M., Ayala, J.L., Olcoz, K.: Server power modeling for run-time energy optimization of cloud computing facilities. Energy Procedia 62, pp. 401–410, 2014.

[22] Mondal, D., & Patil, S. S. (2022). EEG Signal Classification with Machine Learning model using PCA feature selection with Modified Hilbert transformation for Brain-Computer Interface Application. Machine Learning Applications in Engineering Education and Management, 2(1), 11–19. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/20

[23] Banerjee, S., Adhikari, M.: Development and analysis of a new cloudlet allocation strategy for QoS improvement in cloud. Arab. J. Sci. Eng., vol. 40, no. 5, pp. 1409–1425, 2015.

[24] Garg N, Goraya MS. Task deadline-aware energy-efficient scheduling model for a virtualized cloud. Arabian Journal for Science and Engineering.; vol. 43, no. 2, pp. 829-841, 1 Feb, 2018.

[25] Dhiman, O. ., & Sharma, D. A. . (2020). Detection of Gliomas in Spinal Cord Using U-Net++ Segmentation with Xg Boost Classification. Research Journal of Computer Systems and Engineering, 1(1), 17–22. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/20

[26] Anastasi, G.F., Carlini, E., Coppola, M., Dazzi, P.: QBROKAGE: a genetic approach for QoS cloud brokering. In: IEEE, International Conference on Cloud Computing, pp. 304–311, 2020.