International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 5, October 2023, pp. 5898~5907
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i5.pp5898-5907

Resource-efficient workload task scheduling for cloud-assisted internet of things environment

Naveen Kumar Chowdaiah[1], Annapurna Dammur[2]
[1]Department of Computer Science and Engineering, PESIT-Bangalore South Campus, Bengaluru, India
[2]Department of Computer Science and Engineering and Information Science and Engineering, PESIT-Bangalore South Campus, Bengaluru, India

Article Info

Article history:

Received Nov 16, 2022
Revised Dec 23, 2022
Accepted Feb 4, 2023

Keywords:

Cloud-assisted
Internet of things
Machine learning
Montage
Resource efficient

ABSTRACT

One of the most challenging tasks in the internet of things-cloud-based environment is the resource allocation for the tasks. The cloud provides various resources such as virtual machines, computational cores, networks, and other resources for the execution of the various tasks of the internet of things (IoT). Moreover, some methods are used for executing IoT tasks using an optimal resource management system but these methods are not efficient. Hence, in this research, we present a resource-efficient workload task scheduling (RWTS) model for a cloud-assisted IoT environment to execute the IoT task which utilizes few numbers of resources to bring a good tradeoff, achieve high performance using fewer resources of the cloud, compute the number of resources required for the execution of the IoT task such as bandwidth and computational core. Furthermore, this model mainly focuses to reduce energy consumption and also provides a task scheduling model to schedule the IoT tasks in an IoT-cloud-based environment. The experimentation has been done using the Montage workflow and the results have been obtained in terms of execution time, power sum, average power, and energy consumption. When compared with the existing model, the RWTS model performs better when the size of the tasks is increased.

Corresponding Author:

Naveen Kumar Chowdaiah
Department of Computer Science and Engineering, PESIT-Bangalore South Campus
Bengaluru, Karnataka, India
Email: naveenphd872@gmail.com

1. INTRODUCTION

The internet of things (IoT) is a platform where various devices and applications are connected to the internet [1], [2]. Nowadays, the IoT plays a major role in various fields such as home security, environment protection, intelligent traffic, and health [3]. The IoT sensors can obtain data very easily and this helps humans to acquire the data in high quality [4]. Furthermore, the IoT resource management technique can monitor and control various sensors without overriding resources [4]. Resource management includes various applications, networks, and different kinds of system management [5]. This resource management also helps to check the utilization of the resources, and performance and configures the different networks for the extraction of the data. In most scenarios, the IoT is used to collect the data from various sensors which the humans convert the IoT data for useful information using the data analysis process. To analyze the data, most users use the cloud-based IoT environment because it helps to analyze the data more efficiently. Furthermore, the cloud computing area has a major role to play in the IoT-cloud-based environment [6]. The cloud provides various resources such as virtual machines, computational cores, networks, and various other resources for the execution of the various tasks of the IoT. One of the most challenging tasks in the

IoT-cloud-based environment is the resource allocation for the tasks. Furthermore, there are some methods to execute IoT tasks using an optimal resource management system that is not efficient. Hence, present a model which computes the number of resources required such as the computational core, bandwidth, and resources, to process a certain real-time task in the cloud environment [7], [8]. Furthermore, we present resource-efficient workload task scheduling (RWTS) model for cloud-assisted IoT environment to execute the IoT tasks using minimal resources to bring a good tradeoff and also to achieve high performance with the least resources utilized. In the next section we have presented the literature survey about the various IoT-cloud-based environments and how have they handled the problem of resource utilization to execute IoT tasks.

2. LITERATURE SURVEY

In [5], they have explained the various key technologies and challenges in IoT edge computing. They have explained how cloud services help the various IoT applications to compute their various operations and store data in the cloud. In [6], they have designed an architecture for the computation of IoT operations at the highest performance. In this model, they have achieved the highest performance by letting the computational nodes automatically change dynamically according to the task requirements. Furthermore, they have achieved security for their model using the blockchain. In [7], they have designed a resource management system for the WebIDE cloud server where the IoT tasks are prescheduled. The experimental results have shown that by using this method, the time is reduced for the execution of IoT tasks, and also the cloud resources are used efficiently. In [8], they have developed a model for the resource management system which manages the quality of service (QoS) requirement automatically to achieve the service level agreements and its violations. The service level agreement (SLA) aware automatic resource management technique reduces the service level agreement violation for better resource management in the cloud server. The results show that by optimizing the quality of service and also by reducing service level agreement violations, the resources can be allocated efficiently. In [9], they have presented a cost-effective algorithm that completes the execution of the tasks of the workflow in a given time constraint. The results show that the model reduces the cost and time. In [10] they have presented a dynamic cost-effective model which executes the various tasks of the workload in a given time constraint. They have used the advantage of the virtual machines in the cloud to execute the tasks. The results show that the model has better performance when compared with the existing current state-of-art models. In [11], they have presented a model which reduces the cost and energy in a given deadline constraint for the execution of the tasks of the workflow tasks in the cloud. The model cost and energy aware scheduling (CEAS) reduces the cost of the execution and energy by using the virtual machine (VM) selection algorithm. They have used CloudSim to experiment on the real-time dataset. In [12], they surveyed various challenges and methods used for scheduling the workflow in the cloud. In this review, they have analyzed and compared various models that are currently being used for workflow scheduling. In [13], they have proposed a framework that provides a resource management system for real-time IoT data in the cloud environment. This helps to reduce the traffic in the network, reduce energy consumption, and increase the network lifetime which gives a good tradeoff. In [14], they have reviewed various issues and challenges for the cloud-based resource allocation for the IoT environment. Moreover, they have also investigated the various parameters which affect resource management such as load balancing, service level agreement violation, which increases energy consumption and cost. In [15], they have presented a model which utilizes fewer resources to execute the IoT sensor tasks by a given deadline. To execute this model, they have presented an algorithm, energy-efficient collaborative task computation offloading (ECTCO) algorithm. The results show that this model reduces the cost and energy using fewer resources when compared with the current state-of-art models.

In [16], they have discussed the various challenges such as the energy consumption and resource utilization faced by the VM in the cloud. Hence, they have developed an algorithm that provides a better resource utilization technique to execute the tasks of the workflow in the virtualized cloud environment. The experiments have been performed using a real-time dataset and the results show that it has better performance in terms of makespan, energy consumption, and load balancing. In [17], they have presented two methods for the execution of the tasks in an infrastructure-as-a-service (IaaS) cloud. This model has reduced the cost by 100% by providing better resources for the execution of the tasks. In [18], they have presented a model for the multi-objective workflow which provides a good trade-off solution for the workflow scheduling of the tasks in the cloud. The results show that the model has better performance in terms of reduced cost and optimized makespan when compared with the state-of-the-art scheduling methods. In [19], they have presented a model which executes the real-time stream workflows using the genetic algorithm which provides good tradeoffs and reduces the cost by reducing resource utilization. In [20], [21], they presented a model for scheduling the tasks and allocating the tasks according to the task requirement in the cloud-fog environment. In [22], they have proposed an efficient task scheduling approach in an IoT environment. They have compared the results with the particle swarm optimization technique, artificial bee colony (ABC), and genetic algorithm (GA). In [23], they have proposed deep reinforcement learning algorithms for task scheduling in a fog environment. In [24], they have proposed a task scheduling approach using grey wolf optimization for providing a better quality of service. Most of the above models have not focused to reduce the computational resources (bandwidth and computational core) and to provide a better resource management model. Hence, we present a RWTS model for cloud-assisted IoT environment which computes the number of resources required such as bandwidth and computational core for processing certain real-time tasks. Further, we design an efficient resource provisioning/scheduling algorithm for executing IoT tasks using minimal resource usage for bringing a good tradeoff between achieving high performance with minimal resource usage which has been shown in the next section.

3. METHODOLOGY

This section provides a model to calculate the number of resources required such as computational core, processor, and bandwidth to process a given real-time task. This section also provides an efficient provisioning/scheduling method to execute an IoT task with the utilization of the least resources to bring a good tradeoff, reduce energy consumption, and also achieve high performance. In this section first, we present a system model to execute the workload in proposed architecture described in Figure 1, then we present a model to calculate the consumption of energy required during the execution of the task by the machines and finally we present a model for the task scheduling as described in flow diagram of Figure 2.

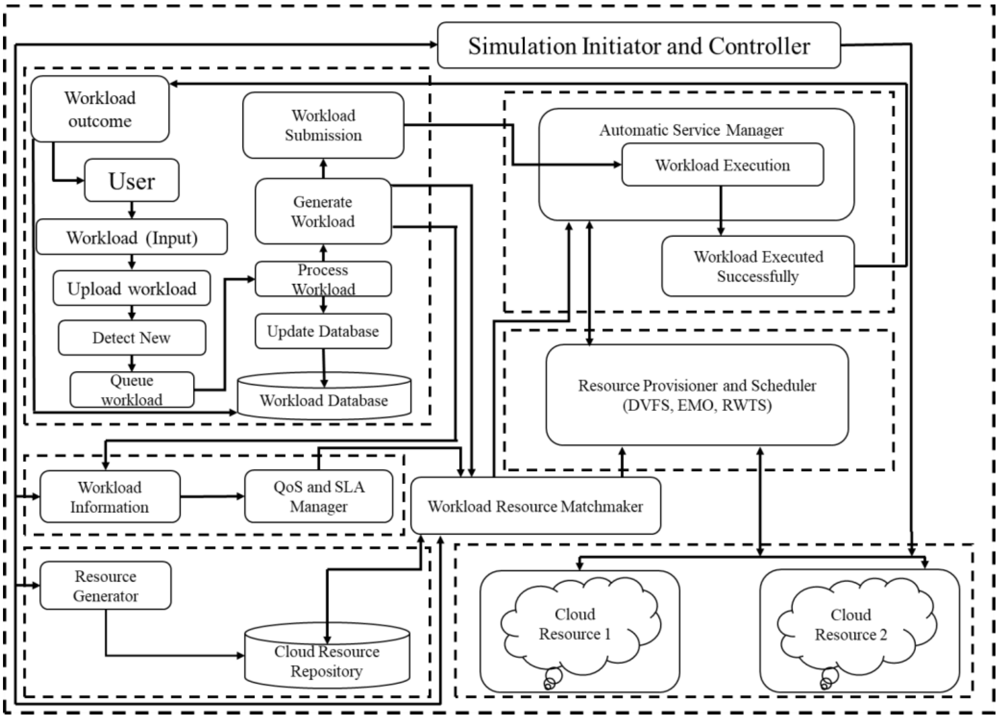

Figure 1. Architecture of the proposed model

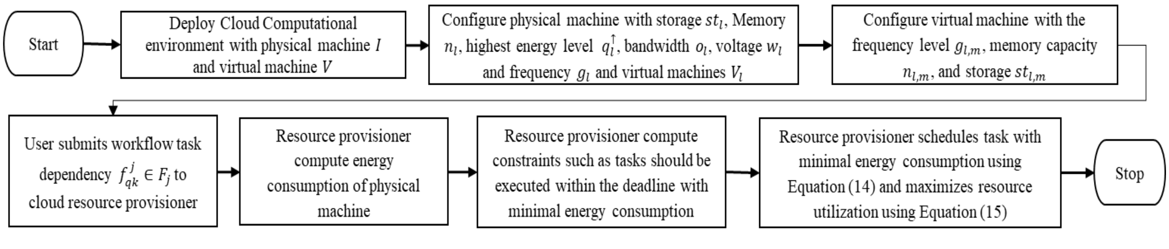
Start
Deploy Cloud Computational environment with physical machine I and virtual machine V
Configure physical machine with storage st_l, Memory n_l, highest energy level $q_l^{\uparrow}$, bandwidth o_l, voltage w_l and frequency g_l and virtual machines V_l
Configure virtual machine with the frequency level $g_{l,m}$, memory capacity $n_{l,m}$, and storage $st_{l,m}$
User submits workflow task dependency $f_{qk}^{j} \in F_j$ to cloud resource provisioner
Resource provisioner compute energy consumption of physical machine
Resource provisioner compute constraints such as tasks should be executed within the deadline with minimal energy consumption
Resource provisioner schedules task with minimal energy consumption using Equation (14) and maximizes resource utilization using Equation (15)
Stop

Figure 2. Flowchart of the proposed model

3.1. Model to execute the workload

A basic cloud computing server consists of users, the internet, physical machines, and virtual machines. Consider a cloud-assisted IoT environment in which there are many physical machines represented using I which can be represented using (1):

$$I = \{I_1, I_2, I_3, \dots, I_o\} \tag{1}$$

In (1), o is used to represent the physical machines that are currently in the cloud-assisted IoT environment. There are different parameters for a given physical machine such as memory capacity, capacity, and storage size. All these parameters for a given physical machine can be represented using I_l. Hence, for a given physical machine $I_l \in I$. The I_l can be represented as (2):

$$I_l = \{st_l, n_l, q_l^{\uparrow}, o_l, (g_l, w_l), V_l\} \tag{2}$$

In (2), st_l is used to define the size of the storage, n_l is used to define the capacity of the memory, $q_l^{\uparrow}$ is used to define the highest energy level, o_l is used to define the capacity of the bandwidth, (g_l, w_l) is used to define the levels of voltage and frequency and the V_l is used to define the number of virtual machines present in the physical machine. Moreover, the levels of voltage and frequency (g_l, w_l) for the given physical machine I_l can be represented using (3):

$$(g_l, w_l) = \{(g_l^1, w_l^1), (g_l^2, w_l^2), \dots, (g_l^{\uparrow}, w_l^{\uparrow})\} \tag{3}$$

In the same way, the virtual machines present in the physical machine I_l can be represented using (4):

$$V_l = \{v_{l,1}, v_{l,2}, \dots, v_{l,|V_k|}\} \tag{4}$$

Furthermore, every virtual machine can be defined using (5):

$$v_{l,m} = \{g_{l,m}, n_{l,m}, st_{l,m}\} \tag{5}$$

In (5), $g_{l,m}$ is used to define the frequency level of the virtual machine, $n_{l,m}$ is used to define the memory capacity of the virtual machine and $st_{l,m}$ is used to define the size of the storage. In this model, the hardware resources can be utilized by the virtual machines and these virtual machines can migrate among the physical machine according to the requirement of the task.

3.2. Model for the energy consumption in the physical machines

In this section, we discuss the energy consumption by the physical machines during the execution of the task and present a model which reduces the energy consumption during the execution process. For every execution of a task, the processor executes the task, and to run the processor energy is consumed. Consider a processor running and it is currently executing a given task, then the energy consumed by the processor can be represented using t_l and the highest peak of energy consumed by the physical machine can be represented using i_l. From the existing method [14], the consumption of energy by a given physical machine can be represented using (6)

$$I_l = t_l * q_l^{\uparrow} * z_l^u + (1 - t_l) * q_l^{\uparrow} * \left(\left(g_l^{\uparrow}\right)^3\right)^{-1} * (g_l)^3 \tag{6}$$

In (6), t_l is used to define the highest energy consumed, $q_l^{\uparrow}$ is used to define the highest energy level, z_l^u is used to define if the physical machine is currently active or not, also $z_l^u \in \{1,0\}$. g_l is used to define the frequency of CPU at the time u, and $g_l^{\uparrow}$ is used to define the highest frequency of the CPU utilized. Using all the different parameters the energy consumed by each physical machine can be calculated using (7):

$$\mathcal{E} = \sum_{j=1}^{o} \int_{\mathrm{xt}}^{\mathrm{yt}} \left(t_l * q_l^{\uparrow} * z_l^u + (1 - t_l) * q_l^{\uparrow} * \left(\left(g_l^{\uparrow}\right)^3\right)^{-1} * (g_l)^3\right) dt \tag{7}$$

In (7), all the variables are not dependent on the time except g_l and z_l^u which can vary according to the given time; hence they are time-dependent. The variables which do not vary with time are static.

3.3. Model for task scheduling the workload in a cloud-assisted IoT environment

To execute the task in a cloud-assisted IoT environment we require a physical machine with some number of virtual machines to execute the whole task. Consider a physical machine in which the virtual machines $v_{l,m}$ has to execute various tasks u_k^j. To execute the tasks, consider a mapping association $y_{k,lm}^j$ between the virtual machines $v_{l,m}$ and various tasks u_k^j. If the mapping association $y_{k,lm}^j$ equals to 0 then we can say that the given tasks have been mapped to the virtual machine. If the mapping association $y_{k,lm}^j$ equals to 1 then we can say that the given tasks have not been mapped to the virtual machine. This can be expressed mathematically using (8):

$$y_{k,lm}^j = \begin{cases} 0, \; if \; u_k^j \text{ is not mapped to } v_{l,m} \\ 1, \qquad\qquad \text{otherwise.} \end{cases} \tag{8}$$

The tasks that have to be executed have different dependencies and sometimes require more time to execute the given task. This problem of data dependency on the given task can be summarized using (9):

$$gu_{q,lm}^j + uu_{qk}^j \le st_{k,lm}^j, \;\; \forall f_{qk}^j \in F_j \tag{9}$$

In (9), $gu_{q,lm}^j$ is used to define the completion time of the given task u_q^j, uu_{qk}^j is used to define the time utilized for the transmission of data from one task u_q to another task u_j. After the whole process of the execution of the tasks in the virtual machines $v_{l,m}$, the maximum time utilized for the execution of the task u_k^j is given by (10):

$$gu_j = \max_{u_q^j \in U_j} \{gu_{q,lm}^j\}. \tag{10}$$

Furthermore, to maintain the allocation of resources for a given task u_k^j, the tasks have to be executed within a given deadline. The task deadline is given using (11):

$$gu_j \le e_j, \;\; \forall x_j \in X \tag{11}$$

Moreover, if an existing task is currently running and if a new task arrives and the job requirement of that task has more priority then they have to be allocated some of the resources from the currently running tasks. This problem of restriction of the resources in the physical machine can be described using (12) and (13):

$$g_l^{\uparrow} - \sum_{m=1}^{|V_l|} g_{l,m} \ge 0, \;\;\; \forall i_l \in I \tag{12}$$

$$n_l - \sum_{m=1}^{|V_l|} n_{l,m} \ge 0, \;\;\; \forall i_l \in I \tag{13}$$

The main focus of this model is to resolve the constraints which have been mentioned in (9)-(13) and also to reduce the consumption of energy during the allocation of the resources for the given tasks of the workload, (14) is used to resolve all the constraints.

$$Min \sum_{l=1}^{o} \int_{xt}^{yt} \left(t_l * q_l^{\uparrow} * z_l^u + (1 - t_l) * q_l^{\uparrow} * \left(\left(g_l^{\uparrow} \right)^3 \right)^{-1} * (g_l)^3 \right) dt \tag{14}$$

In (14), o is used to define the size of the physical machine, xt is used to define the starting time of the execution of the workload, and yt is used to define the ending time of the execution of the workload. Moreover, in (14), all the variables are not dependent on the time expected g_l and z_l^u which can vary according to the given time; hence they are time-dependent. The variables which do not vary with time are static. Furthermore, this model also focuses to provide good tradeoffs, allocating the resources efficiently, and also providing better performance by utilizing fewer resources, hence, the following equation is given by (15):

$$Max \left(\sum_{j=1}^{n} \sum_{k=1}^{|U_j|} cpu_k^j * \mathcal{T}_k^j \right) / \left(\sum_{l=1}^{o} g_l^{\uparrow} * \mathcal{A}_l \right), \tag{15}$$

In (15), n is used to define the size of the workload of X, $|U_j|$ is used to define the size of the task in the workload x_j, cpu_k^j is used to define the frequency of the CPU for the given task u_k^j, $\mathcal{T}_k^j$ is used to define the execution of the complete workload, o is used to define the size of the physical machine in a cloud-assisted IoT environment and $\mathcal{A}_l$ is used to define the active state of the physical machine. Moreover, scheduling tasks on various computing machines is an NP-Hard problem. The proposed RWTS described in algorithm 1 is efficient in obtaining ideal resources heuristically meeting constraints such as reducing energy consumption and meeting task deadline.

Algorithm 1. Resource-efficient workload task scheduling (RWTS)

```
Step 1. Start
Step 2. Deploy cloud computational environment composed of physical machines $I$ and virtual machines $V$.
    ➔ Configure physical machines with storage $st_l$, Memory $n_l$, highest energy level $q_l^{\uparrow}$, bandwidth $o_l$, voltage $w_l$ and frequency $g_l$ and virtual machines $V_l$ using (2).
    ➔ Configure the virtual machine with the frequency level $g_{l,m}$, memory capacity $n_{l,m}$, and storage $st_{l,m}$ using (1).
Step 3. The user submits workflow task dependency $f_{qk}^j \in F_j$ to cloud resource provisioner
Step 4. The resource provisioner according to the task deadline finds resources from the cloud service provider and schedules tasks with constraints defined in (9)-(13).
    ➔ Compute energy consumption of physical machine using (7).
    ➔ Compute constraints such as tasks should be executed within the deadline with minimal energy consumption using (9)-(13).
    ➔ The resource provisioner schedules task with minimal energy consumption (using (14) and maximizes resource utilization using (15).
Step 5. Measure the scheduling efficiency of RWTS in terms of minimizing execution time and minimizing energy consumption.
Step 6. Stop
```

4. RESULTS AND DISCUSSION

In this section, the results obtained during the experimentation have been discussed. The Montage workflow has been considered which has been shown in Figure 3 [25]. The Montage workflow has different workloads such as Montage 25, Montage 50, Montage 100, and Montage 1,000. We have considered all the given montage workflows for the experimentation of the proposed model. The experiments have been conducted on a Windows 10 operating system containing 8 GB RAM and 500 GB of hard disk. The experiments have been conducted using CloudSim. The experiments have been performed based on execution time, power sum, average power, and energy consumption required by the resources for the execution of the tasks of the Montage workflow. Host and VM sizes are heterogeneous.

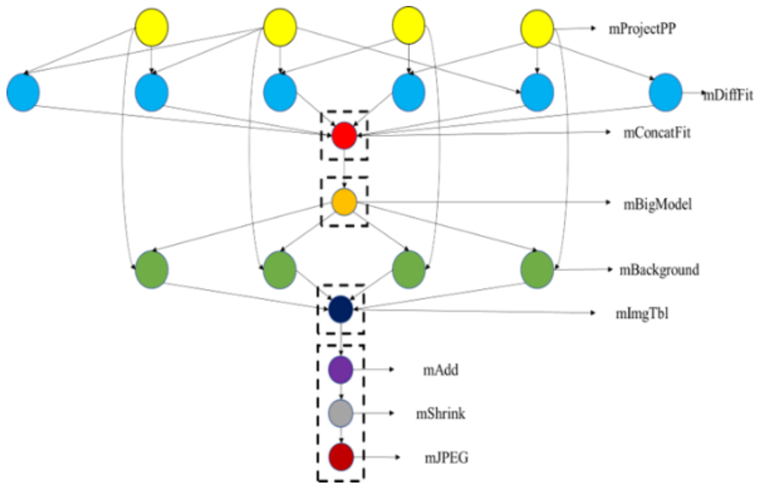

Figure 3. Montage workflow

4.1. Execution time

In this section, the execution time taken for the execution of the Montage workflow by the proposed RWTS model and the existing dynamic voltage frequency scaling (DVFS) model has been discussed [18]. Figure 4 shows the execution time taken for both models. In Figure 4 it can be seen that the DVFS model executes the workflow in 174.67, 387.53, 817.35, and 8591.39 seconds for the execution of the Montage 25, Montage 50, Montage 100, and Montage 1,000 respectively whereas the proposed model RWTS executes the Montage workflow in 57.72, 71.77, 113.31 and 873.82 seconds for the execution of the Montage 25, Montage 50, Montage 100 and Montage 1,000 respectively. From the results, it can be seen that as the size of the Montage workflow increases proposed model consumes less time and resources for the execution of the tasks and gives a better result. The figure shows the proposed model tends to perform extremely well for larger workloads; this is due to the adoption resource maximization function introduced in (15), which aids in fully utilizing the system resource meeting processing time concerning energy constraints.

4.2. Power sum

In this section, the power sum consumed for the execution of the Montage workflow by the proposed RWTS model and the existing DVFS model [18] has been discussed. Figure 5 shows the power sum consumed for both models. In Figure 5 it can be seen that the DVFS model consumes 500514.3738, 1110499.188, 2342175.633, 24619248.96 Watts of energy for the execution of the Montage 25, Montage 50, Montage 100, and Montage 1,000 respectively whereas proposed model RWTS executes the Montage workflow in 126969.3334, 157874.1816, 249266.5583 and 1922339.057 Watts of energy for the execution of the Montage 25, Montage 50, Montage 100 and Montage 1,000 respectively. From the results, it can be seen that as the size of the Montage workflow increases proposed model consumes power sum and resources for the execution of the tasks and gives a better result. The figure shows the proposed model tends to perform extremely well for larger workloads; this is due to the adoption energy minimization function introduced in (14), which aid in reducing the energy required for the execution of respective task time requirement. Thus, more task is executed by utilizing resource efficiently i.e., using a smaller number of physical resources.

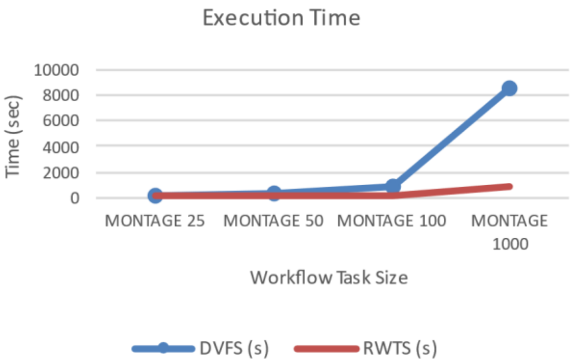

Figure 4. Execution time comparison with the existing DVFS system

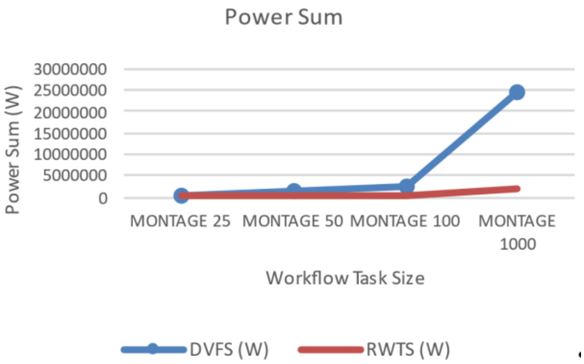

Figure 5. Power sum comparison with the existing DVFS system

4.3. Average power

In this section, the average power consumed for the execution of the Montage workflow by the proposed RWTS model and the existing DVFS model [18] has been discussed. Figure 6 shows the average power consumption for both models. In Figure 6 it can be seen that the DVFS model consumes 28.65562182, 28.65567714, 28.655701, and 28.6557204 Watts of energy for the execution of the Montage 25, Montage 50, Montage 100, and Montage 1,000 respectively whereas the proposed model RWTS executes the Montage workflow in 21.99901827, 21.99910567, 21.99923734 and 21.99943531 Watts of energy for the execution of the Montage 25, Montage 50, Montage 100 and Montage 1,000 respectively. From the results, it can be seen that as the size of the Montage workflow increases proposed model consumes less power and resources for the execution of the tasks and gives a better result. The average power reduction is due to the adoption of (14) and (15) for scheduling tasks with high resource utilization that meets energy constraints considering the application deadline prerequisite.

4.4. Energy consumption

In this section, the energy consumed by the resources for the execution of the Montage workflow by the proposed RWTS model and the existing DVFS model [18] has been discussed. Figure 7 shows the energy consumption of the resources for both models. In Figure 7 it can be seen that the DVFS model consumes 3271084, 432.306718, 1191.416193, and 68107.65154 Watt-hour for the execution of the Montage 25, Montage 50, Montage 100, and Montage 1,000 respectively whereas the proposed model RWTS consumes

39.18879852, 52.60213278, 95.8242739 and 937.90767914 Watt-hour for the execution of the Montage 25, Montage 50, Montage 100 and Montage 1,000 respectively. From the results, it can be seen that as the size of the Montage workflow increases proposed model consumes less energy and resources for the execution of the tasks and gives a better result. The figure shows the proposed model tends to perform extremely well for larger workloads; this is due to the adoption energy minimization function introduced in (14), which aid in reducing the energy required for the execution of respective task time requirement. Thus, more task is executed by utilizing resource efficiently i.e., using a smaller number of physical resources.

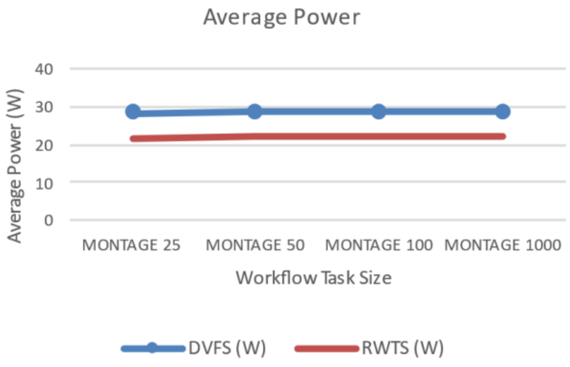

Figure 6. Average power comparison with the existing DVFS system

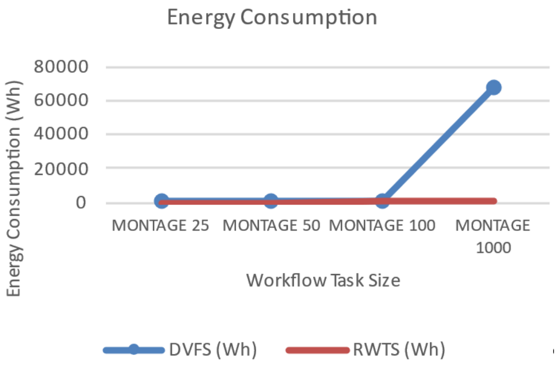

Figure 7. Energy consumption comparison with the existing DVFS system

4.5. Average execution time comparison

In this section, the average execution time consumed to execute the different montage workloads using different models has been discussed. Figure 8 shows the graphical representation of the average energy time consumed by each model for Montage 25, Montage 50, Montage 100, and Montage 1,000. The evolutionary multi-objective (EMO) model [19] has taken 8.44, 9.78, 10.58, and 11.36 seconds to execute the Montage 25, Montage 50, Montage 100, and Montage 1,000 workflow respectively. The DVFS model [18] has taken 6.9868, 7.7506, 8.1735, and 8.591 seconds to execute the Montage 25, Montage 50, Montage 100, and Montage 1,000 workflow respectively. The proposed RWTS model has taken 2.2188, 1.435, 1.331, and 0.9532 seconds to execute the Montage 25, Montage 50, Montage 100, and Montage 1,000 workflow respectively. The figure shows the proposed model tends to perform extremely well for larger workloads; this is due to the adoption resource maximization function introduced in (15), which aids in fully utilizing the system resource meeting processing time concerning energy constraints.

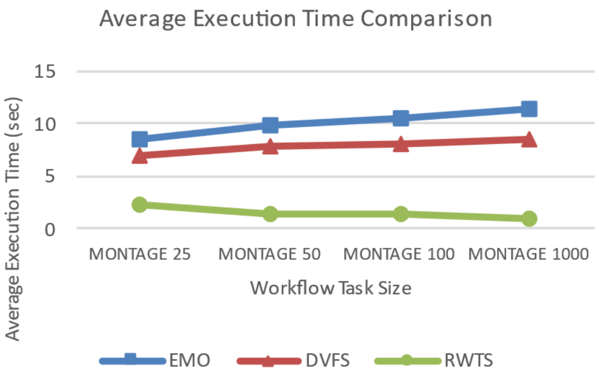

Figure 8. Average execution time comparison with the existing systems

From all the above results shown above, the proposed model performs better in terms of execution time, power sum, average power, and energy consumption. Moreover, the proposed model consumes fewer resources to execute the complex montage workflow and the proposed model has outperformed the existing models. The significant performance enhancement is due to the adoption of the energy minimization function introduced in (14) and the resource utilization maximization function introduced in (15) considering the makespan-energy constraint meeting application deadline prerequisite.

5. CONCLUSION

The research work presents a RWTS model for a cloud-assisted IoT environment to execute the IoT task which utilizes fewer amount of resources to bring a good tradeoff and also to achieve high performance using fewer resources of the cloud. The work has also presented a brief survey about the challenges of the existing models and presented a solution for these challenges using the proposed model. We have presented a model which reduces the energy consumed by the resources and also presented a task scheduling process in which the resources are used efficiently for the execution of the tasks of the workload. Furthermore, the results have been presented for a proposed model which has been compared with the existing DVFS model. The results show that the proposed model performs better in terms of execution time, power sum, average power, and energy consumption. The proposed model also consumes less time when compared with the EMO model and DVFS model. The results have shown that the proposed model consumes fewer resources and time when the workflow is large. When the workflow is small it performs well than the existing systems. Hence, the proposed model can provide efficient resources for workload task scheduling in a cloud-assisted IoT environment. Future work would consider testing the model under different complex workflow datasets such as the cybershake, epigenomics, and inspiral.

REFERENCES

[1] S. P. M, D. Annapurna, S. Ibrahim, K. Venkata, and S. Shravan, "Remote weather assistant using LoRa network, an enhancement of IoT design," *Advancement of Computer Technology and Its Applications*, vol. 2, no. 2, pp. 1–9, 2019.

[2] N. Kumar, J. R. Desai, and D. Annapurna, "ACHs-LEACH: efficient and enhanced LEACH protocol for wireless sensor networks," in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Jul. 2020, pp. 1–6, doi: 10.1109/CONECCT50063.2020.9198666.

[3] N. K. C and D. Annapurna, "Efficient data transmission model for internet of things application," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Feb. 2021, pp. 343–348, doi: 10.1109/ICICV50876.2021.9388614.

[4] C. N. Kumar and D. Annapurna, "Efficient data transmission model for internet of things application," *International Journal on Information Technologies and Security*, vol. 14, no. 2, pp. 3–14, 2022.

[5] S. Naveen and M. R. Kounte, "Key technologies and challenges in IoT edge computing," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Dec. 2019, pp. 61–65, doi: 10.1109/I-SMAC47947.2019.9032541.

[6] M. A. Lopez Pena and I. M. Fernandez, "SAT-IoT: an architectural model for a high-performance Fog/edge/cloud IoT platform," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Apr. 2019, pp. 633–638, doi: 10.1109/WF-IoT.2019.8767282.

[7] H. Wang, J. Li, J. Tian, and K. Wang, "WebIDE cloud server resource allocation with task pre-scheduling in IoT application development," *IEEE Access*, vol. 8, pp. 16216–16224, 2020, doi: 10.1109/ACCESS.2020.2967790.

[8] S. Singh, I. Chana, and R. Buyya, "STAR: SLA-aware autonomic management of cloud resources," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1040–1053, Oct. 2020, doi: 10.1109/TCC.2017.2648788.

[9] A. A. Nasr, N. A. El-Bahnasawy, G. Attiya, and A. El-Sayed, "Cost-effective algorithm for workflow scheduling in cloud computing under deadline constraint," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3765–3780, Apr. 2019, doi: 10.1007/s13369-018-3664-6.

[10] J. Sahni and P. Vidyarthi, "A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 2–18, Jan. 2018, doi: 10.1109/TCC.2015.2451649.

[11] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," *IEEE Transactions on Services Computing*, vol. 11, no. 4, pp. 713–726, Jul. 2018, doi: 10.1109/TSC.2015.2466545.

[12] S. Yassir, Z. Mostapha, and T. Claude, "Workflow scheduling issues and techniques in cloud computing: a systematic literature review," in *Lecture Notes in Networks and Systems*, vol. 49, Springer International Publishing, 2019, pp. 241–263.

[13] Y. Elbanoby, M. Aborizka, and F. Maghraby, "Real-time data management for IoT in cloud environment," in *2019 IEEE Global Conference on Internet of Things (GCIoT)*, Dec. 2019, pp. 1–7, doi: 10.1109/GCIoT47977.2019.9058394.

[14] S. A. Ali, M. Ansari, and M. Alam, "Resource management techniques for cloud-based IoT environment," in *Internet of Things (IoT)*, Cham: Springer International Publishing, 2020, pp. 63–87.

[15] F. Liu, Z. Huang, and L. Wang, "Energy-efficient collaborative task computation offloading in cloud-assisted edge computing for IoT sensors," *Sensors*, vol. 19, no. 5, Mar. 2019, doi: 10.3390/s19051105.

[16] N. Malik, M. Sardaraz, M. Tahir, B. Shah, G. Ali, and F. Moreira, "Energy-efficient load balancing algorithm for workflow scheduling in cloud data centers using queuing and thresholds," *Applied Sciences*, vol. 11, no. 13, Jun. 2021, doi: 10.3390/app11135849.

[17] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, and M. Chen, "Cost and makespan-aware workflow scheduling in hybrid clouds," *Journal of Systems Architecture*, vol. 100, Nov. 2019, doi: 10.1016/j.sysarc.2019.08.004.

[18] B. Hu, Z. Cao, and M. Zhou, "Energy-minimized scheduling of real-time parallel workflows on heterogeneous distributed computing systems," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2766–2779, Sep. 2022, doi: 10.1109/TSC.2021.3054754.

[19] T.-P. Pham and T. Fahringer, "Evolutionary multi-objective workflow scheduling for volatile resources in the cloud," *IEEE*

Transactions on Cloud Computing, vol. 10, no. 3, pp. 1780–1791, Jul. 2022, doi: 10.1109/TCC.2020.2993250.

[20] S. V, P. M, and M. K. P, “Energy-efficient task scheduling and resource allocation for improving the performance of a cloud–Fog environment,” *Symmetry*, vol. 14, no. 11, Nov. 2022, doi: 10.3390/sym14112340.

[21] A. S. Abohamama, A. El-Ghamry, and E. Hamouda, “Real-time task scheduling algorithm for IoT-based spplications in the cloud–Fog environment,” *Journal of Network and Systems Management*, vol. 30, no. 4, Oct. 2022, doi: 10.1007/s10922-022-09664-6.

[22] Z. Movahedi, B. Defude, and A. M. Hosseininia, “An efficient population-based multi-objective task scheduling approach in fog computing systems,” *Journal of Cloud Computing*, vol. 10, no. 1, Dec. 2021, doi: 10.1186/s13677-021-00264-4.

[23] S. Swarup, E. M. Shakshuki, and A. Yasar, “Energy efficient task scheduling in Fog environment using deep reinforcement learning approach,” *Procedia Computer Science*, vol. 191, pp. 65–75, 2021, doi: 10.1016/j.procs.2021.07.012.

[24] N. Bansal and A. K. Singh, “Effective task scheduling algorithm in cloud computing with quality of service alert bees and grey wolf optimization,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 25, no. 1, pp. 550–560, Jan. 2022, doi: 10.11591/ijeecs.v25.i1.pp550-560.

[25] J. Viil and S. N. Srirama, “Framework for automated partitioning and execution of scientific workflows in the cloud,” *The Journal of Supercomputing*, vol. 74, no. 6, pp. 2656–2683, Jun. 2018, doi: 10.1007/s11227-018-2296-7.

BIOGRAPHIES OF AUTHORS

Naveen Kumar Chowdaiah received his M.Tech. degree in 2011 from VTU and pursuing his Ph.D. from Visveswaraya Technological University in the area of Internet of Things and currently working as Assistant Professor in the Department of Computer Science and Engineering at Global Academy of Technology, Bangalore. Scholar has published research papers in the areas of IoT. He holds professional body membership from ISTE and his research area includes sensor networks, IoT and cloud computing. He can be contacted at email: naveenphd872@gmail.com.

Annapurna Dammur received her Ph.D. in the year of 2015 from Bangalore University and currently working as Professor and Head for the Department of Computer Science and Engineering at PES Institute of Technology. She has 25 years of rich experience in the Academics and 10 years of research experience. She is a senior member of IEEE, member of WIE and an active member of many technical events. Her research area includes wireless ad hoc networks, sensor networks, cloud computing and internet of things. She can be contacted at email: annapurnad@pes.edu.