INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Delay Aware and Performance Efficient Workflow Scheduling of Web Servers in Hybrid Cloud Computing Environment

Sangeeta P Sangani[1], **Sunil F Rodd**[1]

1 Department of Computer Science and Engineering, KLS Gogte Institute of Technology, Karnataka, Belagavi, India

OPEN ACCESS

Received: 27.09.2021

Accepted: 24.03.2022

Published: 28.05.2022

Citation: Sangani SP, Rodd SF (2022) Delay Aware and Performance Efficient Workflow Scheduling of Web Servers in Hybrid Cloud Computing Environment. Indian Journal of Science and Technology 15(20): 965-975. https://doi.org/10.17485/IJST/v15i20.1809

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background : To design an effective workflow scheduling optimization of web servers that will bring good trade-offs to meet the workflow task delay prerequisite and performance requirement. **Methods:** This study presents a delay-aware and performance-efficient energy optimization (DAPEEO) technique for workflow execution in a heterogeneous environment (i.e., edge-cloud environment). This technique provides a workflow execution model which meets the application delay prerequisite and performance requirement. **Findings:** Our model has been designed to reduce the energy consumption, increase throughput, reduce computational cost and computational time to provide a delay-aware and performance-efficient workflow model for the web servers in hybrid cloud computing. Our model Delay Aware and Performance Efficient Workflow Scheduling of Web Servers in Hybrid Cloud Computing Environment (DAPEEO) has reduced the energy consumption by 4.217%, increased the throughput by 19.51%, and reduce the computational cost by 62.38% when compared with the existing Deadline-Constrained Cost Optimization for Hybrid Clouds (DCOH) models. Furthermore, the average energy consumption showed a reduction of 40.993% and 90.384% when compared with the DCOH and Self-Configuring and Self-Healing of Cloud-based Resources (RADAR) workload model respectively. Experiment outcome shows the DAPEEO technique achieves much superior energy efficiency, throughput and computation cost reduction when compared with the existing workflow execution model. **Novelty:** Existing model failed to balance reducing cost, and meeting workflow execution deadlines under a heterogeneous environment. On the other side, the DAPEEO is efficient in bringing trade-offs in reducing energy dissipation and meeting task deadlines with reduced cost under the edge-cloud computing model.

Keywords: Cloud Computing; EdgeCloud; DAPEEO; Energy Efficiency; Throughput; Cost

1 Introduction

Workflow scheduling optimization of web servers had been an area of research in hybrid IaaS Cloud (i.e., edge-cloud) because it is an NP-hard problem. The major focus of standard workflow scheduling generally aims to finish the execution of the workflow within the desired cost and to meet the cutoff time. The scheduler chooses the best-fit assets for the jobs to limit the cost enhance resource utilization, and minimize the execution time. In[1] to reduce the consumption of energy during the scheduling of workflow and to use more cache resources for the allocation of the tasks, an EARU method has been proposed. This model uses less energy, time, power for the execution of the workflow when compared with the other existing DVFS workflow scheduling methods. In[2] using the PSO method and algorithm, ALO has been proposed to optimize the workflow scheduling in the cloud environment. In this algorithm, a security method DES has been used to encode the cloud data when the scheduling process is executed. The experimentation results have been carried out in terms of load balancing, makespan, and cost. In[3] to provide QoS requirements in fog computing, a framework DoSP has been proposed which gives the services for both the cloud and fog nodes. This framework uses GA to provide the different services required by the user in fog cloud computing. The main focus of this methodology was to use less cost resources in the fog computing environment. In[4] for the execution of the workflow in a given deadline method, EMMOO has been proposed which executes the workflow tasks in two steps. This method uses the deadline-aware method to reduce the time and consumption of energy. In[5] a survey on various cloud computing environments (meta-heuristic, heuristic, and hybrid clouds) has been done. This survey reviews various algorithms and methods used for scheduling the workflow. In[6] to resolve the problem of security in the workflow scheduling problems, a method EAFSA based on EAFSAIPR, IC-PCP, and HECC has been proposed. The goal of this method is to complete the workflow scheduling task within the given deadline with less energy consumption and to provide more security. In[7] as the execution of the workflow task is more expensive and takes more time to execute, a method MOWOS which reduces the cost and increases the makespan has been proposed. In[8] a load balancing task scheduling method, ALTS, for the cloud environment has been proposed. As the problem of load balancing is NP-hard, the solutions for each problem keep increasing. This method maps all the incoming tasks to the virtual machines to decrease the usage of resources, makespan, and SLA violations.

In[9] studied tradeoff optimization of makespan and cost for executing workloads under cloud computing environment. Here they presented an improved heterogeneous earliest-finish-time (HEFT)-based scheduling mechanism namely FDHEFT adopting fuzzy dominance rule. Performance is evaluated considering both synthetic and real-time workloads with good makespan and cost tradeoffs performance when compared with standard workload execution models. In[10] deadline and cost optimization model namely DCOH for a hybrid cloud computing environment. DCOH aims to reduce the cost of execution workload with deadline constraints. Further, modeled multi-objective optimization technique to optimize cost and makespan simultaneously. Experiment outcome shows the model is efficient in bringing good tradeoffs between makespan and cost performance. However, reducing energy consumption plays a very important role in reducing the cost of workload execution. In[11] presented an energy-aware processor merging methodologies for executing DAG workload with deadline constraints for reducing energy consumption under a hybrid cloud environment. Further, quick energy-aware processor merging methodologies is modeled for reducing the computational overhead of energy-aware processor merging methodologies. In[12] modeled cost and energy-aware scheduling mechanism for bringing tradeoff performance of cost and energy. The algorithm is composed of the following phases such as VM selection, task merging, idle VM resource reuse policies, and slack resource optimization for saving energy. Achieves better performance than other existing workload models considering simple task workflows. In[13] presented energy and cost optimization scheduling approach for executing data-intensive IoT workload under cloud environment. The scheduling mechanism is designed using historical data of VM resource usage in different physical machines. To optimize energy dissipation of cloud environment the VM resources are scaled up or down; further, in reducing cost jobs are executed through batch processing. Their model reduces cost and improves energy efficacy by guaranteeing workload task deadline prerequisite. In[14] focused on addressing the reliability requirement of mission-critical workloads. Here a scheduling design is modeled that jointly optimizes reliability and energy efficacy considering workload QoS prerequisite. The scheduling process is composed of the following stages such as selectivity computation, task clustering, time distribution, selecting clusters with suitable V/F levels. Good energy efficacy and reliability outcome is achieved when compared with state-of-art scheduling models.

In recent time evolutionary algorithm such as swarm optimization, genetic algorithm has been used for solving multi-objective scheduling problem in a heterogeneous cloud environment for executing real-time workloads[15–19]. Further, game-theoretic and Reinforcement Learning (RL) models have also been utilized for solving multi-objective scheduling constraints for workflow execution in a cloud environment[20–22]. In[23] optimized makespan and cost by modeling Cooperative Game Theory (CGT)-based scheduling strategy for executing relatively very large workloads. In[24] presented Reinforcement Learning (RL)-based workload scheduling technique with different task selectivity. In[25] integrated fuzzy with GT for admission control and balancing load among physical machines. In[26], for optimizing the task completion time and balance load presented Q-

Learning model with weighted objective function for workload scheduling on a cloud platform. However, this model is not evaluated considering modern scientific workflows.

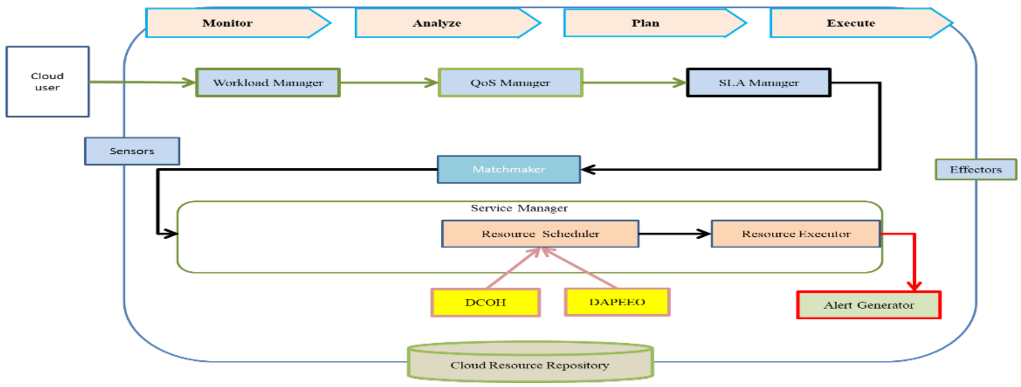

Fig 1. Workload execution on cloud computing platform [27].

In [27] presented an autonomous resource management technique namely RADAR. RADAR is designed to handle unexpected failure and dynamic resource management according to workload prerequisites. RADAR reduces execution time, execution cost, and SLA violation when compared with the standard workload execution model. In [28] showed optimal mapping among task and resource availability is needed for reliable execution of large real-time workloads; such requirement is considered to be an NP-complete problem (i.e., failed to obtain an optimal solution or no exact solution is obtained). However, a major concern is energy efficiency, makespan reduction, reliability, and fault-tolerant prerequisite modern data-intensive applications. In this work, the workload scheduling is designed in such a way that it can optimize makespan and energy parameters per application dynamic requirement and resource availability considering a multi-core computational framework. Recently, several workload scheduling models have been presented considering minimizing energy constraints and maintaining a certain level of QoS. However, existing scheduling models induces high communication cost among processors because of high memory resource usage and routing overhead. Inter-processor communication cost is very high using existing techniques as well as these algorithms are relied upon a classic model which is not realistic when applied to a hybrid cloud environment [29–31]. In recent times workflow scheduling adopting cloud computational environments have been widely studied in [31]. For overcoming research issues this paper presents a delay-aware and performance-efficient energy-optimized workflow execution model for a hybrid cloud environment. The DAPEEO model can meet task delay prerequisites and reduce energy consumption by effectively reducing task failure and attaining high-performance efficiency through the energy optimization model.

The significance of the work:

- Presented workflow execution model that meets application delay prerequisite and performance requirement.
- The DAPEEO technique improves energy efficiency and reduces computational costs for a hybrid cloud environment.

2 Delay Aware and Performance Efficient Workflow Execution Model (DAPEEO For Hybrid Cloud Environment)

This work presents a Delay-Aware and Performance-Efficient Energy Optimization (DAPEEO) technique for workflow execution in a hybrid cloud computing environment. The architecture of the hybrid cloud environment is shown in Figure 2. First, the hybrid cloud system and workflow execution model is described. Second, the workload execution data transmission model is described. Lastly, the workload execution model using the DAPEEO technique is presented.

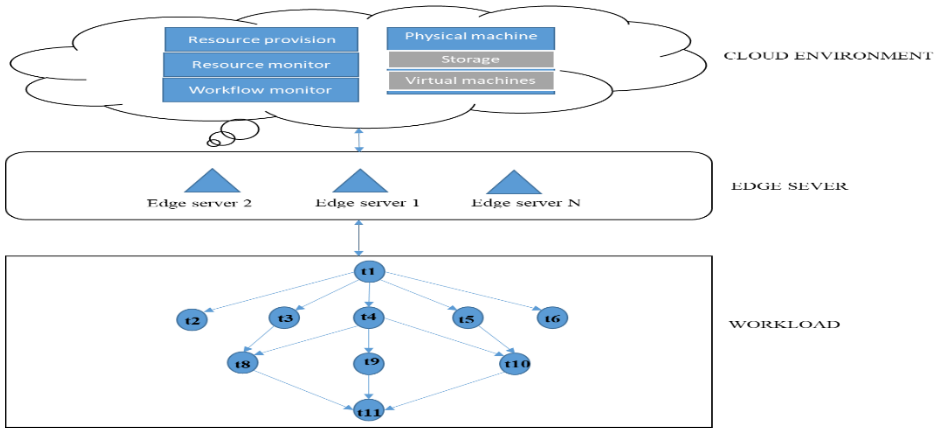

Fig 2. The architecture of a Hybrid cloud environment for workload execution

2.1 System and Workflow Execution on Hybrid Cloud Environment

This work considers a hybrid cloud environment for executing the scientific workflow. Let's consider there are a set of IoT devices that are connected for running DAG applications. An example of such a workflow is shown in Figure 1. These devices are connected to an edge server for computing different operations. Further, the edge server is connected cloud environment. The cloud environment is composed of a set of host and virtual machines as described below

$$T = \{t_1, t_2, t_3, \ldots, t_n\} \tag{1}$$

The edge device is also composed of a computing device for executing a task, and its virtual machine is represented as t_0. This work assumes that there will be a stable connection for the communication and computation process. The edge server possesses both compute-intensive and deadline aware workflows; its performance efficiency and deadline prerequisite are S_{pre} andM_{pre}, respectively. Here the edge server can execute workflow or can offload it to the cloud environment based on task performance and deadline requirement. The proposed system model will aid in improving overall system performance with minimal task failures.

The workflow model is described as follows; the task is divided into o subtasks. The association among different tasks is represented as a DAG as follows

$$H = \{U,\ F\} \tag{2}$$

The subtask in H is represented as follows

$$U = \{u_1, u_2, u_3, \ldots, u_o\} \tag{3}$$

and each subtask has predefined computational resource requirement as follows

$$\{\alpha_j, \beta_j\} \tag{4}$$

where α_j represent task size and β_j represent total processing element needed for completing the task u_j. The workflow connected edges F is represented as follows

$$F = \{f_{jl}\left(t_{k,}t_m\right)\} \tag{5}$$

As shown in Figure 1, the edge defines the association among different subtasks, and its directions define dependencies that subsequent subtasks shouldn't be initialized till the preceding subtasks are completed. For easiness, the following notation $prec\left(u_j\right)$ and $subseq\left(u_j\right)$ defines preceding and forthcoming subtasks ofu_j, respectively. The communication overhead among subtask u_j and u_l is represented by the following function$f_{jl}\left(t_k, t_m\right)$. Communication cost is dependent on the processing element of the cloud environment and the edge server. Thus, if u_j and u_l are not associated, the for random t_k andt_m, $f_{jl}\left(t_k, t_m\right) = 0$. The subtask without forthcoming subtask is represented as end subtask u_{end} and similarly, the subtask without preceding subtask is represented initialization task u_{init}.

2.2 Workload Data Transmission Model For Hybrid Cloud Environment

Here every edge server is connected with a cloud server; thus, can offload computing of subtask to cloud server. The communication bandwidth of offloading subtask u_j computing process from the edge server to cloud environment t_j is represented asw_k, and is computed using the following equation

$$w_k = X log_2\left(1 + \frac{Q_{hk}}{\sigma^2}\right) \tag{6}$$

where X represent communication data rate, Q defines transmission power of edge server, σ^2 defines its noise power, and hk describe communication gain among edge server and cloud environment t_k. The communication gain is computed using the following equation

$$hk = e_k^{-\gamma} \tag{7}$$

where e_j depicts distance among the edge server and cloud environment t_k, and γ defines the path loss component. Thus, the communication delay of offloading subtask u_j to cloud server t_k is obtained using the following equation

$$M_{jk}^{s} = \frac{\delta_j}{w_k} \tag{8}$$

The overall energy induced by the edge server for offloading subtask u_j to cloud server is computed using the following equation

$$\mu_{jk}^{s} = QM_{jk}^{s}. \tag{9}$$

2.3 Workload Execution Model for Hybrid Cloud Environment

The workload is either executed in the edge server or is offloaded and executed in the cloud environment. The delay induced for workload subtask u_j execution in an edge server environment with the processing element $G_0$is defined using the following equation

$$M_j^m = \frac{\alpha_j}{G_0} \tag{10}$$

The energy induced for executing a task in an edge server environment is obtained using the following equation

$$\mu_j^m = \varphi M_j^m \tag{11}$$

φ represent energy expended by processing element per instance of time.

Similarly, the delay induced for executing offloaded workload on cloud environment is computed. The computation capacity of the cloud environment is described as follows

$$G = (G_1, G_2, G_3, \ldots, G_n\} \tag{12}$$

The delay induced for completing workload execution is composed of communication delay and execution delay. Thus, the overall delay for completing workload execution in the cloud environment is obtained using the following equation

$$M_{jk}^{0} = \frac{\alpha_j}{G_k + M_{jk}^{s}} \tag{13}$$

For easiness, the delay induced for executing workload locally on edge server or cloud environment is measured using the following equation

$$M_{jk} = \frac{\alpha_j}{G_k + M_{jk}^{s}} \tag{14}$$

where $M_{jk}^{s} = 0$ when $k = 0$.

The task failure rate for workload subtask u_k considering Poisson distribution on processing element t_k is obtained using the following equation

$$S_{jk} = f^{-\omega_k \alpha_k / G_k} \tag{15}$$

The processing efficiency for executing DAG workload is computed as follows[25]

$$S(H) = \prod_{s_j \in T} \sum_{t_k \in T} y_{jk} S_{jk} \tag{16}$$

where y_{jk} defines subtask index, which can be shown as follows

$$y_{jk} = \begin{cases} 0, \text{ if the task } \mathrm{u_j} \text{ is given to processing element } \mathrm{t_k} \\ 1, \text{ otherwise} \end{cases} \tag{17}$$

In this work for workload task H execution, the energy consumption is minimized considering reducing delay and improving performance efficiency. Here the energy consumption for executing workload in edge server and on cloud environment is uniformly measured using the following equation

$$\mu_j = y_j \mu_j^m + \sum_{k=1}^{n} y_{jk} \mu_{jk}^s \tag{18}$$

Then, the objective problem can be defined using the following equation

$$min \sum_{j=1}^{o} \mu_j \tag{19}$$

The following constraint must be satisfied by the above-described objective function in achieving an effective workload execution solution

$$M(H) \leq M_{preq} \tag{20}$$

The above equation (20) describes the constraint of total execution delay of workload G must be lesser than given delay bounds.

$$S(H) \leq S_{preq} \tag{21}$$

The bound described in the above equation (21) described the performance efficiency of workload must be higher than given performance efficiency bounds.

$$\sum_{k=0}^{n} y_{jk} = 1 \tag{22}$$

The constraint defined in the above equation (22) describes that every subtask can be either executed in an edge server or the cloud environment.

$$\tau(u_j) \geq \tau(u_l) + \sum_{k=0}^{n} y_{lk} M_{lk}, \quad \forall u_l \in prec(u_j) \tag{23}$$

$$\tau(u_j) + \sum_{k=0}^{n} y_{jk} M_{jk} \leq \tau(u_l), \quad \forall u_l \in subseq(u_j) \tag{24}$$

$$y_{jk} \in (0, 1\} \tag{25}$$

The constraint described in Equation (23) and equation (24) describes the succeeding subtasks that have to wait till its preceding subtask has completed the subtask, where $\tau(u_j)$ is the initializing time of a subtask u_j. The main idea of the proposed methodology is to reduce task execution delay and improve performance efficiency which relies on the certain constraint described in Eq. (20) to (25); here the constraint with minimal energy consumption resources are obtained for workload execution in the hybrid cloud environment. First, the subtask is ordered in an upward manner for generating subtask sequence set $\widehat{U}$. Second, obtain the processing efficiency and delay bounds of every subtask on different processing elements in edge server and cloud environment. Lastly, when allocating resources for a subtask$\widehat{u}_j$, find a processing element with minimal energy overhead and satisfies the bounds for executing subtask$\widehat{u}_j$. Thus, the proposed workflow scheduling technique can reduce cost and energy consumption when compared with existing workflow scheduling methodologies which are experimentally shown below.

3 Results and Discussion

Here experiment is conducted for evaluating the performance of Delay Aware and Performance Efficient Energy Optimization (DAPEEO) over existing workflow scheduling techniques[10],[27]. The existing Deadline-Constrained Cost Optimization for Hybrid Clouds (DCOH)[10] for the Heterogeneous Cloud Computing Environment have mainly focused to reduce the computational cost and makespan. They have utilized the Deadline-Constrained Cost Optimization to reduce the cost of the workflow in a given deadline by 100.0%. They have also proposed another model Multi-objective Optimization for hybrid clouds (MOH) to optimize the makespan and reduce the cost simultaneously while executing the workflow. In Self-Configuring and Self-Healing of Cloud-based Resources (RADAR)[27], they have mainly focused to reduce the cost of execution, SLA violation, and execution time. The main objective of the RADAR is to solve the problems of unexpected failures and also to allocate the resources to the given task if there is any failure by a given server. The experiment is conducted using cyber-shake workflow. Energy efficiency, throughput, and computation cost are considered for evaluating performance. The IoT-edge server environment is modeled using the SENSORIA simulator and the cloud environment is modeled using CloudSim and is combined through object-oriented programming language in building a hybrid cloud environment.

3.1 Workflow Description

The scientific workflow is generated by the Pegasus group, ASKALON. Each workflow has different computation features and structures. These workflows are CPU, I/O, and memory-intensive in nature. This work evaluates the workflow execution model using Cyber-shake workflow; the cyber-shake workflow is used for characterizing earthquake hazards using synthetic seismograms by the Southern California Earthquake Center (SCEC). The tasks in cyber-shake are generally are memory hungry and CPU intensive. The cyber-shake workflow is described in Figure 3. More details of cyber-shake workflow can be obtained from Google.

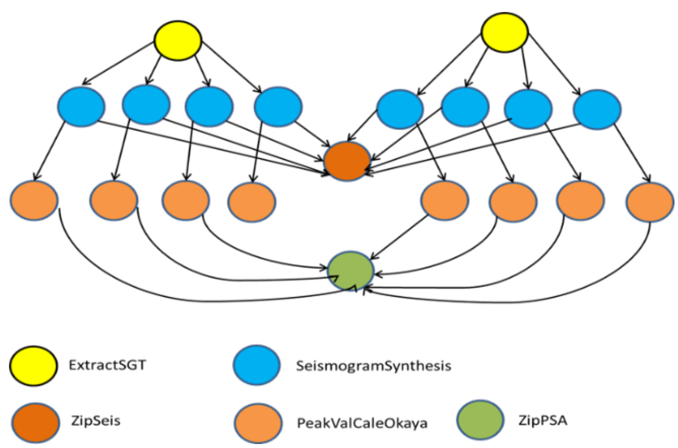

Fig 3. A sample representation of cyber shake

3.2 Energy Efficiency Performance

Here experiment is conducted for evaluating energy efficiency performance achieved using DAPEEO and the existing method[10]. Experiments have been conducted by varying CyberShake workload size 30, 50, 100, and 1000. The average energy consumed for executing workload is measured in Watts. Figure 4 shows the outcome achieved using DAPEEO over the DCOH model. From the result, it can be seen proposed DAPEEO achieves much better energy efficiency in comparison with the DCOH workload scheduling model considering varied workload sizes. An average energy efficiency performance improvement of 20.25% is achieved using DAPEEO over the DCOH workload scheduling model. Similarly, the experiment is conducted for 100 and 1000 workload sizes for evaluating the performance of other existing approaches such as RADAR[27] and DCOH as shown in Figure 5. An average energy efficiency performance improvement of 40.993% and 90.384% is achieved using DAPEEO over DCOH and RADAR workload scheduling model, respectively. From the result, it can be stated DAPEEO is very efficient for executing the larger task.

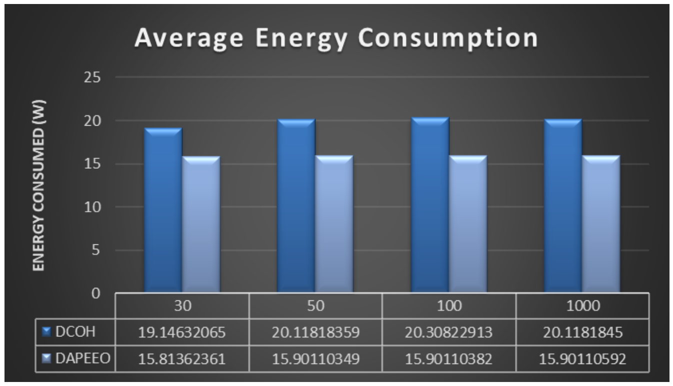

Fig 4. Average energy consumptioncomparison of DAPEEO over existing method using scientific workloadCyber-shake.

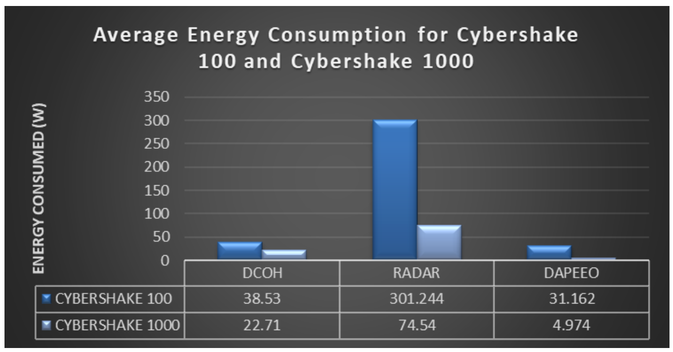

Fig 5. Average energy consumption comparison of DAPEEO over existing workloadschedulingmethod using scientific workload Cyber-shake.

3.3 Throughput

Here experiment is conducted for evaluating throughput performance achieved using DAPEEO and the existing method[10]. Here experiment is conducted by varying CyberShake workload size 30, 50, 100, and 1000. The throughput for executing workload is measured in terms of normalized percentage. Figure 6 shows the outcome achieved using DAPEEO over the DCOH model. From the result, it can be seen proposed DAPEEO achieves much better throughput performance in comparison with the DCOH workload scheduling model considering varied workload sizes. An average throughput performance improvement of 79.14% is achievement using DAPEEO over the DCOH workload scheduling model. From the result, it can be stated DAPEEO throughput performance gets better with increases in workload size. Thus, are efficient for executing the extremely very large-scale workload.

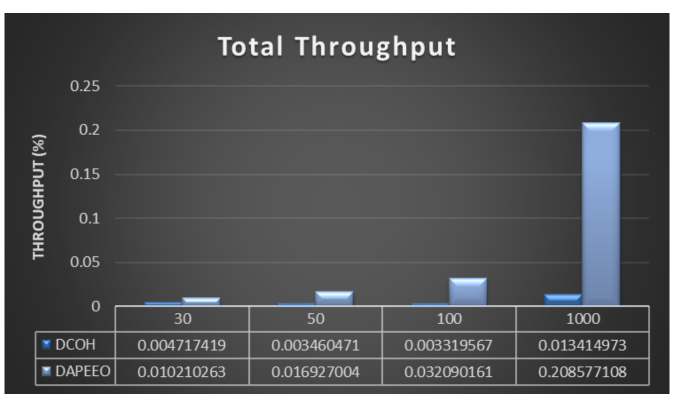

Fig 6. Throughput comparison of DAPEEO over existing workload scheduling method using scientific workloadCyber-shake.

3.4 Computation Cost

Here experiment is conducted for evaluating computation cost performance achieved using DAPEEO and existing methods such as DCOH[10]. Here experiment is conducted by varying CyberShake workload size 30, 50, 100, and 1000. The cost incurred for executing workload is measured under the Microsoft Azure cloud.Figure 7 shows the outcome achieved using DAPEEO over the DCOH model. From the result, it can be seen proposed DAPEEO achieves much better computation cost performance in comparison with the DCOH workload scheduling model considering varied workload sizes. An average computation cost reduction of 78.90% is achieved using DAPEEO over the DCOH workload scheduling model. From the result, it can be stated DAPEEO computation cost performance gets profitable with increasing workload size. Thus, are suitable for provisioning both small and very large-scale workloads with high profit.

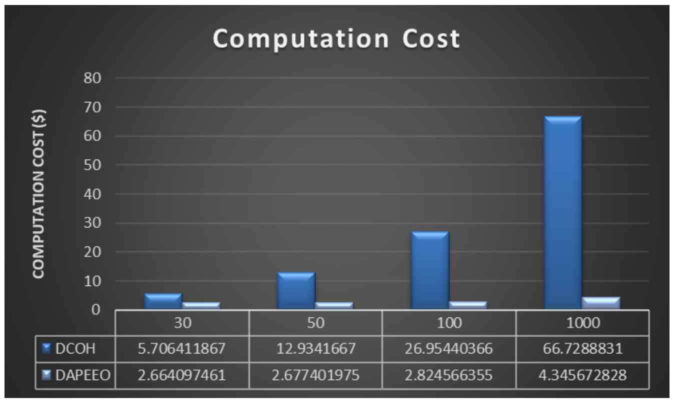

Fig 7. Computation cost comparison of DAPEEO over existing method usingscientific workload Cyber-shake.

3.5 Comparison of the Results

From the overall result achieved in the above section, it can be stated the proposed DAPPEEO is very efficient in reducing energy consumption, reducing cost, and improving throughput over DCOH, 2019[10], RADAR, 2019[10]. Further, to describe the novelty of the proposed DAPPEO, a comparison of the proposed DAPPEO over other recent workflow scheduling frameworks is described in Table 1.

Table 1. Comparison table of proposed DAPEEO over existing workflow scheduling approach

	DCOH, 2019[10]	RADAR, 2019[27]	QRAS, 2021[30]	DAPEEO
Heterogenous cloud	Yes	Yes	No	YES
Scientific workload	Yes	No	No	YES
QoS Metric used for scheduling	Cost and processing time	Cost and reliability	Energy	Energy, Delay, and processing time
Workload type support	Small and medium-large complex workload	Small and medium complex workload	Small and large simple workload	Small and large complex workload
SLA considered	No	No	No	Yes
Fault-tolerant provided	No	Yes	No	Yes

The DCOH-based workflow execution model is effective in minimizing cost and time for the small to medium complex workload. The RADAR-based workflow execution model is efficient in minimizing cost and with good reliability for the execution of the small to medium complex workload. Similarly, the QRAS-based workflow execution model is efficient in energy for the execution of small to large simple workloads. However, the proposed DAPEEO model is efficient in minimizing energy, delay, and processing time for the execution of small to large workloads. From Table 1, it can be noticed the major factor limiting the usage of the existing model is they suffer significantly for the execution of larger workload as they could not guarantee fault tolerance; however, in DAPPEO the adoption of a novel offloading mechanism with energy minimization constraint meeting QOS objective defined significantly aid in achieving better performance for both smaller and larger workloads. No prior work has considered optimizing energy minimization strategy to meet the application deadline and performance requirement of workload execution under the edge-cloud model.

4 Conclusion

This work presented various workflows scheduling algorithms for cloud and edge-cloud environments. The Delay Aware and Performance Efficient Workflow Execution (DAPPEO) model provides a method to increase the performance of the model and reduce the cost for the execution in a given deadline. DAPEEO has reduced the energy consumption by 4.217%, increased the throughput by 19.51%, and reduce the computational cost by 62.38% when compared with the existing Deadline-Constrained Cost Optimization for Hybrid Clouds (DCOH) models. Furthermore, the average energy consumption showed a reduction of 40.993% and 90.384% when compared with the DCOH and Self-Configuring and Self-Healing of Cloud-based Resources (RADAR) workload models respectively. Experiment outcome shows the DAPEEO technique achieves much superior energy efficiency, throughput and computation cost reduction when compared with the existing workflow execution model.

Future work would consider designing of fault-tolerant scheduling algorithm using the soccer league optimization technique for efficiently re-utilizing system resources in executing data-intensive workload scheduling on multi-core cloud computing architecture. The performance of both proposed workflow scheduling and the existing workflow scheduling approach will be evaluated using data-intensive real-time workload such as epigenome will be considered for evaluation.

References

1) Jambigi MV. Energy Aware Resource Utilization Technique for Workflow Scheduling in Cloud Computing Environment. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*. 2021;12(10):7461–72. Available from: https://doi.org/10.17762/turcomat.v12i10.5652.
2) Mathew S, Joy KP. Impact of Monosodium Glutamate and Corticosterone in the Hippocampus: Glucocorticoid Regulation and Caspase-3 mediated Microvascular and Neuronal Apoptosis. *The FASEB Journal*. 2020;34:1–1. Available from: https://dx.doi.org/10.1096/fasebj.2020.34.s1.02786.
3) Sriraghavendra M, Chawla P, Wu H, Gill SS, Buyya R. DoSP: A Deadline-Aware Dynamic Service Placement Algorithm for Workflow-Oriented IoT Applications in Fog-Cloud Computing Environments. In: and others, editor. Energy Conservation Solutions for Fog-Edge Computing Paradigms. Springer. 2022;p. 21–47. Available from: https://doi.org/10.1007/978-981-16-3448-2_2.
4) Ijaz S, Munir EU, Ahmad SG, Rafique MM, Rana OF. Energy-makespan optimization of workflow scheduling in fog–cloud computing. *Computing*. 2021;103(9):2033–2059. Available from: https://dx.doi.org/10.1007/s00607-021-00930-0.
5) Sana MU, Li Z. Efficiency aware scheduling techniques in cloud computing: a descriptive literature review. *PeerJ Computer Science*. 2021;7:e509–e509. Available from: https://dx.doi.org/10.7717/peerj-cs.509.
6) Ramathilagam A, Vijayalakshmi K. Workflow scheduling in cloud environment using a novel metaheuristic optimization algorithm. *International Journal of Communication Systems*. 2021;34(5):4746–4746. Available from: https://dx.doi.org/10.1002/dac.4746.
7) Konjaang JK, Xu L. Multi-objective workflow optimization strategy (MOWOS) for cloud computing. *Journal of Cloud Computing*. 2021;10(1):1–19. Available from: https://dx.doi.org/10.1186/s13677-020-00219-1.
8) Mubeen A, Ibrahim M, Bibi N, Baz M, Hamam H, Cheikhrouhou O. Alts: An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing. *Processes*. 2021;9(9):1514–1514. Available from: https://dx.doi.org/10.3390/pr9091514.
9) Zhou X, Zhang G, Sun J, Zhou J, Wei T, Hu S. Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT. *Future Generation Computer Systems*. 2019;93:278–289. Available from: https://dx.doi.org/10.1016/j.future.2018.10.046.
10) Zhou J, Wang T, Cong P, Lu P, Wei T, Chen M. Cost and makespan-aware workflow scheduling in hybrid clouds. *Journal of Systems Architecture*. 2019;100:101631–101631. Available from: https://dx.doi.org/10.1016/j.sysarc.2019.08.004.
11) Xie G, Zeng G, Li R, Li K. Energy-Aware Processor Merging Algorithms for Deadline Constrained Parallel Applications in Heterogeneous Cloud Computing. *IEEE Transactions on Sustainable Computing*. 2017;2(2):62–75. Available from: https://dx.doi.org/10.1109/tsusc.2017.2705183.
12) Li Z, Ge J, Hu H, Song W, Hu H, Luo B. Cost and Energy Aware Scheduling Algorithm for Scientific Workflows with Deadline Constraint in Clouds. *IEEE Transactions on Services Computing*. 2018;11(4):713–726. Available from: https://dx.doi.org/10.1109/tsc.2015.2466545.
13) Wen Y, Wang Z, Zhang Y, Liu J, Cao B, Chen J. Energy and cost aware scheduling with batch processing for instance-intensive IoT workflows in clouds. *Future Generation Computer Systems*. 2019;101:39–50. Available from: https://dx.doi.org/10.1016/j.future.2019.05.046.
14) Garg R, Mittal M, Son LH. Reliability and energy efficient workflow scheduling in cloud environment. *Cluster Computing*. 2019;22(4):1283–1297. Available from: https://dx.doi.org/10.1007/s10586-019-02911-7.
15) Chunlin L, Jianhang T, Youlong L. Hybrid Cloud Adaptive Scheduling Strategy for Heterogeneous Workloads. *Journal of Grid Computing*. 2019;17(3):419–446. Available from: https://dx.doi.org/10.1007/s10723-019-09481-3.
16) Li C, Tang J, Luo Y. Cost-aware scheduling for ensuring software performance and reliability under heterogeneous workloads of hybrid cloud. *Automated Software Engineering*. 2019;26(1):125–159. Available from: https://dx.doi.org/10.1007/s10515-019-00252-8.
17) Zhu Z, Zhang G, Li M, Liu X. Evolutionary Multi-Objective Workflow Scheduling in Cloud. *IEEE Transactions on Parallel and Distributed Systems*. 2016;27(5):1344–1357. Available from: https://dx.doi.org/10.1109/tpds.2015.2446459.
18) Rehman A, Hussain SS, Rehman Z, Zia S, Shamshirband S. Multi-objective approach of energy efficient workflow scheduling in cloud environments. *Concurrency and Computation: Practice and Experience*. 2019;31(8):e4949–e4949. Available from: https://dx.doi.org/10.1002/cpe.4949.
19) Sardaraz M, Tahir M. A parallel multi-objective genetic algorithm for scheduling scientific workflows in cloud computing. *International Journal of Distributed Sensor Networks*. 2020;16(8):155014772094914–155014772094914. Available from: https://dx.doi.org/10.1177/1550147720949142.
20) Bertsekas DP. Feature-based aggregation and deep reinforcement learning: a survey and some new implementations. *IEEE/CAA Journal of Automatica Sinica*. 2019;6(1):1–31. Available from: https://dx.doi.org/10.1109/jas.2018.7511249.
21) Xue L, Sun C, Wunsch D, Zhou Y, Yu F. An adaptive strategy via reinforcement learning for the prisoner dilemma game. *IEEE/CAA Journal of Automatica Sinica*. 2018;5(1):301–310. Available from: https://dx.doi.org/10.1109/jas.2017.7510466.

22) Wang H, Huang T, Liao X, Abu-Rub H, Chen G. Reinforcement Learning for Constrained Energy Trading Games With Incomplete Information. *IEEE Transactions on Cybernetics*. 2017;47(10):3404–3416. Available from: https://dx.doi.org/10.1109/tcyb.2016.2539300.
23) Duan R, Prodan R, Li X. Multi-Objective Game Theoretic Schedulingof Bag-of-Tasks Workflows on Hybrid Clouds. *IEEE Transactions on Cloud Computing*. 2014;2(1):29–42. Available from: https://dx.doi.org/10.1109/tcc.2014.2303077.
24) Cui D, Ke W, Peng Z, Zuo J. Multiple DAGs workflow scheduling algorithm based on reinforcement learning in cloud computing. In: Computational Intelligence and Intelligent Systems. Singapore. Springer. 2016;p. 305–311. Available from: https://doi.org/10.1007/978-981-10-0356-1_31.
25) Iranpour E, Sharifian S. A distributed load balancing and admission control algorithm based on Fuzzy type-2 and Game theory for large-scale SaaS cloud architectures. *Future Generation Computer Systems*. 2018;86:81–98. Available from: https://dx.doi.org/10.1016/j.future.2018.03.045.
26) Jiahao W, Zhiping P, Delong C, Qirui L, Jieguang H. A multiobject optimization cloud workflow scheduling algorithm based on reinforcement learning. In: Intelligent Computing Theories and Application;vol. 15. Cham, Switzerland. Springer. 2018;p. 550–559. doi:10.1007/978-3-319-95933-7_64.
27) Gill SS, Chana I, Singh M, Buyya R. RADAR: Self-configuring and self-healing in resource management for enhancing quality of cloud services. *Concurrency and Computation: Practice and Experience*. 2019;31(1):e4834–e4834. Available from: https://dx.doi.org/10.1002/cpe.4834.
28) Abdulhamid SM, Latiff MSA, Abdul-Salaam G, Madni SHH. Secure Scientific Applications Scheduling Technique for Cloud Computing Environment Using Global League Championship Algorithm. *PLOS ONE*. 2016;11(7):e0158102–e0158102. Available from: https://dx.doi.org/10.1371/journal.pone.0158102.
29) Lu Y, Sun N. An effective task scheduling algorithm based on dynamic energy management and efficient resource utilization in green cloud computing environment. *Cluster Computing*. 2019;22(S1):513–520. Available from: https://doi.org/10.1007/s10586-017-1272-y.
30) Singh H, Bhasin A, Kaveri PR. QRAS: efficient resource allocation for task scheduling in cloud computing. *SN Applied Sciences*. 2021;3(4). Available from: https://dx.doi.org/10.1007/s42452-021-04489-5.
31) Lis A, Sudolska A, Pietryka I, Kozakiewicz A. Cloud Computing and Energy Efficiency: Mapping the Thematic Structure of Research. *Energies*. 2020;13(16):4117–4117. Available from: https://dx.doi.org/10.3390/en13164117.