Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 3, March 2025, pp. 1607~1615
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i3.pp1607-1615

Marine scientific workflow execution resource provisioner in edge-cloud platform

Sheetal Bagali[1], Pradeep Kumar Siddana[2], Poongothai C[1], Kavita Avinash Patil[3], Vani Bengaluru Purushothama[4], Ashwini Kumari P[5]

[1]Department of Electronics and Communication, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India
[2]Department of Electronics and Communication, Nitte Meenakshi Institute of Technology, Bengaluru, India
[3]Department of Electronics and Telecommunication, M S Ramaiah Institute of Technology, Bengaluru, India
[4]Department of Electronics and Communication, Sai Vidya Institute of Technology, Bengaluru, India
[5]Department of Electrical and Electronics, Reva University, Bengaluru, India

Article Info

Article history:

Received May 28, 2024
Revised Sep 21, 2024
Accepted Sep 30, 2024

Keywords:

Cloud computing classification
Cost reduction
Energy efficiency
Makespan efficiency
Resource provisioning
Workflow scheduling

ABSTRACT

Workload application for deadline-intensive marine science, usually represented by directed acyclic graphs (DAGs), consists of interconnected operations that communicate huge quantities of information and operate on cutting-edge computational platforms. Massive communications of information among jobs running on separate computing servers, nonetheless, might come with substantial processing time, power consumption, and financial expenses. Therefore, there is room for further study of exchanging certain communications for computing to lower total interaction expenses. In addressing research this work introduces an effective resource provisioner for deadline-intensive scientific workload executer (ERP-DISWE) to reduce the makespan, consumption of energy, and cost of edge-computing platforms. The performance of ERP-DISWE is validated with the current resource provisioner using the sipht marine workload and shows superior performance concerning makespan, energy, and cost.

Corresponding Author:

Pradeep Kumar Siddana
Department of Electronics and Communication, Nitte Meenakshi Institute of Technology
Bengaluru, India
Email: pradeepkumar.s@nmit.ac.in

1. INTRODUCTION

The use of virtualization technologically in conjunction with cloud-based internet-of-things computing [1] offers a broad opportunity for investigation into all key disciplines and services. As the amount of information in globally keeps growing, computerized information collection is becoming increasingly necessary. This holds not only for astrophysics but also for fields like genomics, geological sciences, geographic information systems, marine genomics, and phenotype study. During the purposes of study, such occupations gather an abundance of data. Occasionally information that is gathered from those locations is kept in a structure of marine scientific workflows [2] represented by directed acyclic graphs (DAGs), which are made up of interconnected jobs that exchange information via links; the result of one activity can be utilized as the starting point for another. These marine workflows, which can include hundreds of jobs, are often carried out on sizable, distributed systems, like edge-cloud [3]. This allows enabling the processing of distinct tasks concurrently, lowering both the processing time and cost. In an edge-cloud platform, scheduling becomes a non-polynomial (NP)-hard issue [4]. To schedule the newly arrived processes via process flow, different systems like the cloudsim simulator [5], [6], and edge workflow simulator [7] have been utilized recently enabling the execution of these assignments.

Numerous techniques are being proposed for managing these newly arrived jobs [8]. Heterogenous earliest time first (HEFT) [9], enhancement-HEFT [10], ant colony optimization (ACO) [11], particle swarm optimization (PSO) [12], and energy and cost aware scheduler employing various optimization strategies [13]. More details of different scheduling methods have been discussed in section 2. The survey shows the current methods are able to attain effective processing efficiency, however, considering large marine complex applications the current methodologies failed to support execution of bigger workloads with lesser processing time and cost [14], [15].

In addressing the research issue the work introduces an effective resource provisioner for deadline-intensive scientific workload executer (ERP-DISWE) using an edge-cloud platform. The proposed model mainly focuses on reducing energy dissipation, and cost and completing the job execution within its deadline in an edge-cloud platform. The proposed ERP-DISWE provides the following contribution. The research provides a resource provisioning scheduler model with a focus on reducing overall cost with energy and processing time minimization constraints of marine applications in an edge-cloud platform. Comparative analysis to evaluate the results with the various scientific workflow resource provisioning scheduler models has been provided. The result shows ERP-DISWE attains much-improved performance concerning cost reduction, energy usage, and makespan reduction in the edge-cloud platform.

Manuscript organization is given. Section 2 studies different schedulers and resource provisioners designed specifically for the execution of scientific workflows. Highlight the current research gaps by analyzing the advantages and disadvantages of current methods. Section 3, the working of the proposed resource provisioner scheduler is designed as a mathematical model. Section 4 provides the outcome of ERP-DISWE with existing resource provisioner schedulers. In the last section, the research significance along with future enhancement is provided.

2. RELATED WORK

This section focused on analyzing different schedulers in the edge-cloud platform; and identifying the advantages, disadvantages, research significance, and limitations of the current scheduler in executing marine workload in the edge-cloud platform [14], [15]. A task replication scheduler (TRA) method has been presented by Yao *et al.* [16] to lower costs along with processing time in cloud environments. Researchers put forward both options in the technique to address cost-time optimization. They explore one conventional scientific workload and three randomized workloads. According to the study's outcomes, the suggested approach has a 31.6% cost reduction and a 17.4% processing time reduction. Sindhu *et al.* [17], an approach enabling job schedulers to lower power usage and costs while improving efficiency in an edge-based computational system has been suggested. The jobs had been regarded as DAG, where workload jobs will be carried out according to their due dates. The Markov decision process (MDP) was additionally utilized in the above framework to allocate the optimal resources to perform process execution. The outcomes demonstrate that it outperforms the current algorithms on the basis of performance. A method for organizing work schedules was presented by Abohamama *et al.* [18] for an edge-cloud environment. Resource planning may be seen by such a method as an algorithmic problem that utilizes permutations. Researchers utilized a new iteration of the genetic algorithm (GA) for calculating the assignments in order to present the aforementioned technique. Depending upon each combination, the jobs are assigned to a server with adequate assets and a short makespan to execute. The trials were carried out, and the outcomes were contrasted against the most effective current systems. In terms of failure rate, time, and cost. The findings are being evaluated with other algorithms; the suggested method performs well.

A job scheduler methodology has been presented by Movahedi *et al.* [19] to decrease the makespan and power usage in an edge-cloud environment. Researchers had drawn out a plan to handle the process's arrival of new jobs in the edge-cloud environment. In order to enhance how well it performs, researchers also suggest a chaotic whale optimizer (CWO) approach and an integer linear program (ILP) method. The outcomes were contrasted using existing evolutionary optimizer methods. The findings demonstrate that, as opposed to the current processes, the suggested job scheduler method achieves higher efficiency. To lower costs, power consumption, and processing time, Swarup *et al.* [20] presented an Internet of Things job scheduler using an edge-cloud platform. An approach based on deep-reinforcement learning has been suggested enabling a job scheduler using an edge-cloud platform. The present research employed an integrated buffering strategy for addressing the problem of job execution planning on computer servers. A particle swarm optimizer (PSO) technique has been presented by Liu *et al.* [21] for an edge-based job scheduler. The approach lowers expenses when the jobs are computed. The calculation was carried out using the cloudsim simulator, while findings were contrasted to those of each of the existing schedulers. The outcomes demonstrate how effectively and economically this strategy maximized both computing cost and time.

A cost optimizer heuristic strategy (COHS) was suggested by Konjaang *et al.* [22] and offers an approach to decrease costs when planning job execution in a cloud platform. The process under this framework has to be broken into smaller jobs to ensure every subprocess can be completed within the allotted due date. Researchers took into account the application procedures for assessing their prototype. Research indicates splitting the assignments into smaller ones allows for significantly quicker server assignments throughout the processing of these assignments. As compared with their previous framework, researchers decreased expenses by 1.2% for the cybershake workload, 3.9% for the montage workload, and 32.5% for the sipht process. To decrease both time and expenses, Konjaang *et al.* [23] expanded their previously suggested COHS research by carrying out a multi-objective workflow optimization method (MOWOS). Both algorithms the highest computational server and lowest computational server have been suggested in this paradigm to compute the jobs derived by the process. According to MOWOS outcomes, processing time is shortened by 10%, and costs are cut by 8%. Masoudi *et al.* [24], focused on addressing the energy constraint issue through effective virtual machine allocation design. Further, both Rui *et al.* [25], and Bacanin *et al.* [26], showed the importance of edge computing in enhancing the quality of service and energy efficiency. Mangalampalli *et al.* [27], showed the importance of considering multi-objective parameters through optimization employing deep reinforcement learning (DRL) aid reducing makespan with minimal consumption; However, effective VM placement according to quality-of-service (QoS) requirement is not done; thereby additional delay and makespan is experienced.

3. PROPOSED METHOD

This section discusses the working of the proposed effective resource provisioner for deadline-intensive marine scientific workload executer using an edge-cloud environment. The cloud-based resource provisioner is composed of physical machines (PMs) and virtual machines (VMs) as shown in Figure 1 [24]. The architecture also has cloud resource monitoring (CRM) to measure computational resources and plan execution according to arrival jobs from the cloud users. The CRM performs VM placement and migration plans according to arrival jobs.

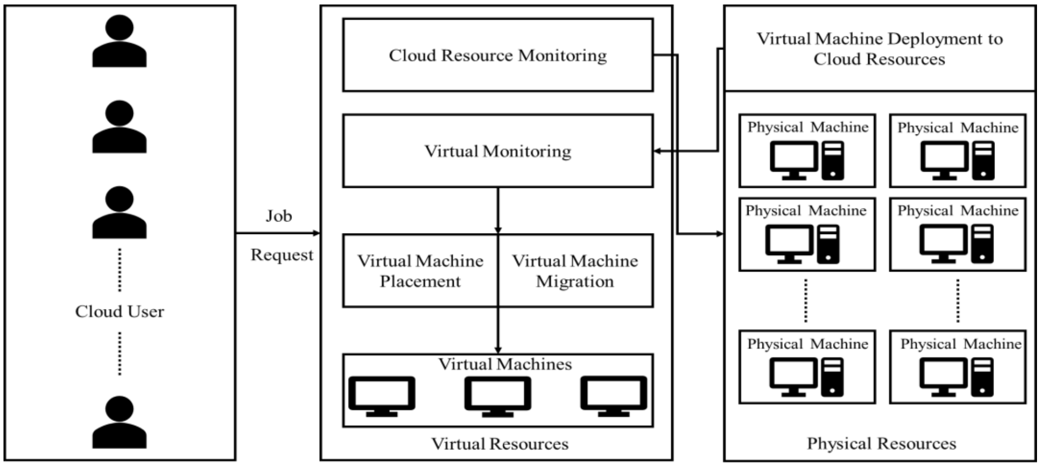

Figure 1. Virtual machine placement optimization inside the physical machine in a cloud computational environment [1]

3.1. Computational resource of edge-cloud

The edge-cloud data center is composed of multiple heterogenous servers M including servers placed in edge and cloud platforms as shown in Figure 2. The data center network (DCN) takes the arrival load from the user through the cloud gateway server. Then, put them in a data block sequence (DBS) according to their arrival and service deadline requirements. Finally, from DBS, the virtual machine controller (VMC) starts performing resource provisioner plans according to the arrival job requirement. As a large number of jobs are being submitted to the cloud platform and arrival is heterogenous in nature; job scheduling is challenging in the edge-cloud cloud platform. Thus, to perform tradeoff optimization the server is kept on during execution, and the rest of the time it is switched off; the focus of the proposed scheduler design is to fully utilize the active server and reduce unutilized idle server; this process will aid in improving overall resource utilization.

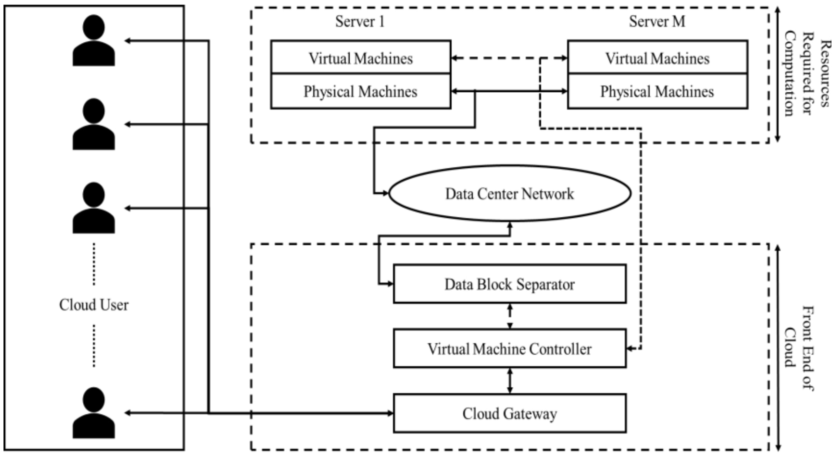

Figure 2. Proposed architecture for the effective resource provisioner for deadline-intensive marine workload execution in the edge-cloud environment

3.2. Workflow model

The marine workload is composed of different heterogeneous subtasks. The task is represented as DAG with size defined in bits using parameter L_b. The DAG is further segmented into N according to the idle and active VMs with $(N = n)$. As every task comes with strict quality requirements with strict deadlines. Therefore, the task should be executed by the VMs within the predefined deadline $of\ S_s$ seconds.

3.3. PMs and VMs classification

The energy-aware resource provisioner process of a number of PMs x needed and how the VMs must be placed to assure the overall energy according to the arrival marine workload task is given in (1).

$$\{freq_x^I, freq_x^{highest}, E_x^I, P_a(x), C_e(x)\}, x = 1,2,3, \cdots, N \tag{1}$$

In (1), $freq_x^I$ defines the idle state processer frequency of the PMs, $freq_x^{highest}$ defines the peak operating frequency of active states processor in respective PMs, E_x^I, $P_a(x)$ defines energy usage of the processor in idle state, $P_a(x)$ defines active PMs ratio and $C_e(x)$ defines capacitor traffic. The processing rate maximum capacity is represented by the parameter ${R_p}^{highest}$ as defined in (2).

$${R_p}^{highest} = freq_x^{highest} \tag{2}$$

3.4. Computation of a task

The constraint $freq_{disc}$ defines different operating frequencies of VMs employing dynamic voltage-frequency optimization. Therefore, the minimum operating frequency of idle VMs is given in (3).

$$freq^{highest} \triangleq freq_Z > freq_{z-1} > freq_{Z-2} > \cdots > freq_1 > freq^I \triangleq freq_0 \tag{3}$$

The current method provides static energy optimization; further, it provides a tenth of a second quicker optimization. However, in proposed model provides dynamic optimization; thus, faster with minimal energy consumption of PMs as given in (4).

$$E_{run-time} = P_a * C_e * freq * vs^2 \tag{4}$$

In (4), $E_{run-time}$ represents runtime energy optimization of the processor inside the VMs and PMs, $freq$ defines the processor frequency of the PMs and vs^2 defines the corresponding voltage used which are measured in (5).

$$freq = \mathbb{C} * \left[\frac{vs^{-1}}{(vs - vs_\tau)^{-2}}\right] \tag{5}$$

In (5), $\mathbb{C}$ defines a static predefined parameter, vs_τ expresses threshold voltage which is significantly less compared with real voltage input vs. Using (4) and (5), by considering VMs in idle state with $E^I \geq 0$, processing cost is measure in (6).

$$SUM_{C-comp}(x) \triangleq \sum_{y=0}^{Z} P_a{}' * C_e * s_{xy} * \frac{1}{freq_y^{-3}},\ x = 1, 2, \dots, N \quad (6)$$

In (6), $SUM_{C-comp}(x)$ is the computation cost, $P_a{}' = \mathbb{C}^{-1} * P_a$, s_{xy} defines the makespan of VMs and PMs operating at respective frequency $freq_y$. The bounds of x within s_{xy} will vary between one N. Similarly, the bound of y within s_{xy} varies between zero to Z. The parameter Z defines varying frequency ranges of VMs with $Z + 1$ diverse bounds.

3.5. Task reconfiguration

The VMC is used to perform task reconfiguration considering PM which hosts VMs with different frequency $freq_x$. The work assumes frequency ranges from $freq_1$ to $freq_2$. The cost involved in reconfiguration from one frequency $freq_1$ to another $freq_2$ is defined by parameter $SUM_{C-reconf}$ as in (7).

$$SUM_{C-reconf}(freq_1; freq_2) = \mathcal{E}_c * \frac{1}{(freq_1 - freq_2)^{-2}} \quad Joule \quad (7)$$

In (7), $\mathcal{E}_c\ Joules/Hz^2$ expresses the computational cost for reconfiguring frequency. The parameters $freq_1$ and $freq_2$ are associated with the lower and upper bound of VMs $VM(x)$. The proposed work is mainly focused on reducing overall frequency switching costs.

3.6. Communication of a task

In measuring the communication cost of different marine workload task execution, the transfer rate parameter $TrnsRate_x$ measured in bits/seconds is used. The value of x varies between one to N. The cost for communication on x^{th} link is measured using parameter E_x^{C-comm} measured in watts is defined in (8).

$$E_x^{C-comm} \equiv E_{TotalTime}^{C-comm}(x) + E_{TrnsRate}^{C-comm}(x) \quad (8)$$

Where $E_{TotalTime}^{C-comm}(x)$ defines switching energy cost and $E_{TrnsRate}^{C-comm}(x)$ defines overall transfer energy cost at the destination. The overall communication cost considering $x = 1\ to\ N$ is measured using (9).

$$E_x^{C-comm}(TrnsRate_x) = \delta_x(\overline{TRT_x} * TrnsRate_x)^2 + E_x^I,\ x = 1\ to\ N \quad (9)$$

In (9), $\delta_x \triangleq \left(R_{gain}\right)^{-1} * \left(K^{-1} * \sqrt{\frac{2*\theta}{3}}\right)^2, x = 1\ to\ N$; where parameter K defines the highest data block size measured in bits and $\theta \in \{1,2\}$ defining how many times it is being admitted, R_{gain} coding gain considering the presence of noise considering link x, $\overline{TRT_x}$ are utilized round-trip average makespan, and E_x^I defines energy cost under idle x^{th} link. Hence, using this the delay for transmission in a single direction can be defined in (10).

$$TrnsDelay(x) = \sum_{y=1}^{Z} R_{p_y} s_{xy} \,/\, TrnsRate_x \quad (10)$$

Using (10), the communication cost is measured considering the overall delay in (11).

$$SUM^{C-comm}(x) \triangleq E_x^{C-comm}(TrnsRate_x) * \left(\sum_{y=1}^{Z} R_{p_y} s_{xy} / TrnsRate_x\right), \quad (11)$$

Then, using (6), (7), and (11) the final cost $\mathcal{C}$ is measured considering the multi-objective minimization function of edge-cloud in (12).

$$\mathcal{C} = \min[SUM_{C-comp}(x) + SUM_{C-reconf}(freq_1; freq_2) + SUM^{C-comm}(x)] \quad (12)$$

Where, $SUM_{C-comp}(x)$ defines computation cost, $SUM_{C-reconf}(freq_1; freq_2)$ defines reconfiguration cost, $SUM^{C-comm}(x)$ defines communication cost. The proposed model optimized employing the dragonfly algorithm [28], [29] provides efficient resource provisioning of workflow achieving better performance and reducing cost is proved in the next section.

4. RESULT AND ANALYSIS

The proposed algorithm (ERP-DISWE) is executed under the sipht scenario to evaluate the makespan, energy consumption, and cost. The ERP-DISWE algorithm is compared with the MOWOS [23] energy minimized scheduler (EMS) [11], modified firefly algorithm-workflow scheduler (MFA-WS) [26], multi-objective deep reinforcement learning-priority-aware workflow scheduler (MODRL-PWS) [27] models. The four different tasks of sipht have been considered for the evaluation of the results. These four tasks include sipht 30 and sipht 100. All experiments are carried out on a computer with Intel(R) Core i5 2.2. gigahertz, 16 gigabytes of memory, Windows 10, 64-bit operating system. To simulate and evaluate the proposed and state-of-art workflow resource provisioning scheduler algorithm's performance, we used cloudsim.

4.1. Makespan performance

In Figures 3 and 4, the makespan for sipht 30 and sipht 100 has been evaluated. It has been compared with the other existing models. The results show that the MFA-WS and EMS models take more makespan for the computation of the tasks. The MODRL-PWS model has reduced the makespan when compared with the MOWOS models. The proposed ERP-DISWE model has reduced the makespan by 42.12 % and 61.44 % when compared with the MODRL-PWS model considering the sipht workflow size of 30 and 100, respectively. The significant makespan reduction is due adoption of optimization done using (10) by leveraging resource provisioning scheduling optimization in an edge-cloud platform.

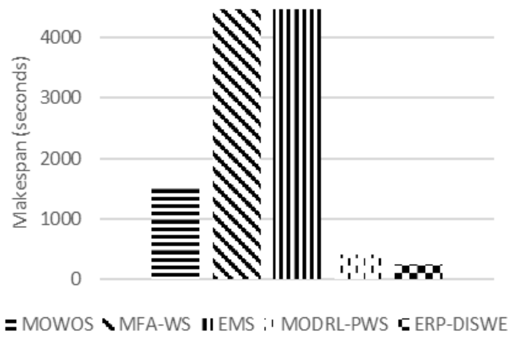

Figure 3. Makespan for sipht 30

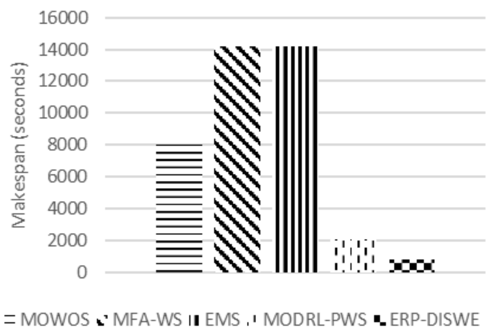

Figure 4. Makespan for sipht 100

4.2. Energy consumption performance

In Figures 5 and 6 the energy consumption for sipht 30 and sipht 100 has been evaluated. It has been compared with the other existing models. The results show that the MFA-WS and EMS models consume more energy for the computation of the tasks. Further, the MOWOS consumers slightly more than MODRL-PWS; however, the proposed model namely the ERP-DISWE model has reduced the energy consumption by 3.8 % and 3.15 % when compared with MODRL-PWS model considering the sipht workflow size of 30 and 100, respectively. The significant energy reduction is due adoption of optimization done using (10) meeting constraints in reconfiguration cost defined in (5) by leveraging resource provisioning scheduling optimizing in the edge-cloud platform.

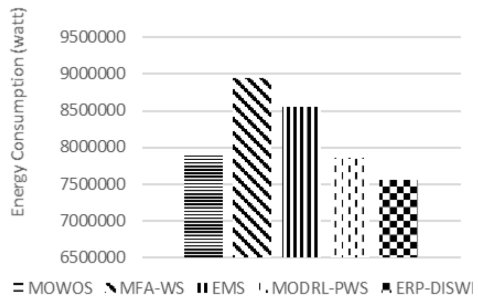

Figure 5. Energy consumption for sipht 30

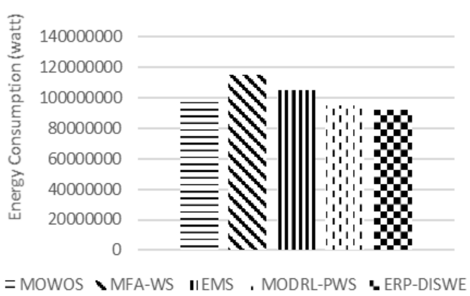

Figure 6. Energy consumption for sipht 100

4.3. Computation cost

In Figures 7 and 8, the computation cost for Sipht 30 and Sipht 1000 has been evaluated. The results show that the existing model's cost is higher when compared with the proposed ERP-DISWE model. EMS exhibits significantly lesser cost than MFA-WS and MOWOS and MODRL-PWS exhibit much lesser cost than EMS; however, the computational cost for the computation of each task has been reduced by the proposed model with a cost reduction 64.95 % 70.66 % over MODRL-PWS model considering Sipht workflow size of 30 and 100, respectively. The significant cost reduction is due to the reduction of computation cost, reconfiguration, and communication cost as defined in (5), (6), and (7), respectively.

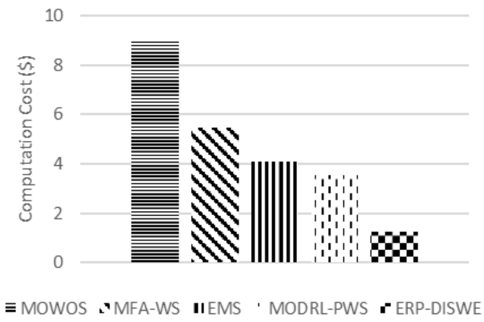

Figure 7. Computation cost for sipht 30

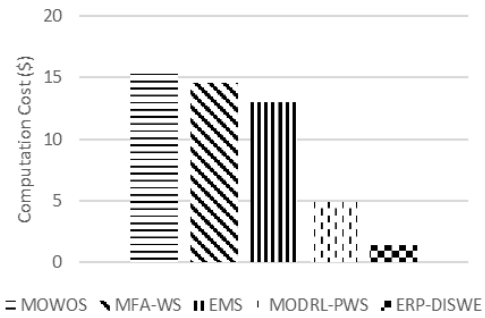

Figure 8. Computation cost for sipht 100

5. CONCLUSION

To execute marine processes via an edge-cloud system, it is suggested in the above investigation an effective resource provisioner for delay-intensive workload execution. They provided an approach within the above scenario that minimizes energy consumption, costs, and makespan yet calculating the process's inbound jobs. The existing methods were examined. As contrasted to the present designs, the findings demonstrate the use of the suggested ERP-DISWE system will minimize the makespan, energy, and cost especially when performing the Sipht workload. Because the MFA-WS and EMS models were unable to shorten the job's execution time and use less energy, their costs were significantly increased. The EMS concept did not shorten the makespan; instead, it concentrated primarily on cutting costs and energy. When taking into account varying workloads, the MODRL-PWS model produced superior outcomes with regard to makespan, cost, and energy consumption, but it was unable to lower overall costs. Each of the above issues have been resolved in the suggested model ERP-DISWE, which has improved efficiency in regard to cost, makespan, and energy usage. As a result, the suggested model outperforms the current one. Improved efficiency is demonstrated by outcomes in terms of costs, energy usage, and makespan. Throughout the future, this work may be utilized for the implementation of additional scientific workflows, such as epigenomics, montage, and inspiration. Future work considers validating the model using workloads like montage, inspiral, and epigenomics. Alongside this, consider validating the proposed model by considering other parameters like deadline and reliability outcomes.

REFERENCES

[1] S. Qin, D. Pi, Z. Shao, Y. Xu, and Y. Chen, "Reliability-aware multi-objective memetic algorithm for workflow scheduling problem in multi-cloud system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1343–1361, Apr. 2023, doi: 10.1109/TPDS.2023.3245089.
[2] X. Tang, "Reliability-aware cost-efficient scientific workflows scheduling strategy on multi-cloud systems," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2909–2919, Oct. 2022, doi: 10.1109/TCC.2021.3057422.
[3] M. Menaka and K. S. Sendhil Kumar, "Workflow scheduling in cloud environment – challenges, tools, limitations & methodologies: a review," *Measurement: Sensors*, vol. 24, p. 100436, Dec. 2022, doi: 10.1016/j.measen.2022.100436.
[4] H. Ma, P. Huang, Z. Zhou, X. Zhang, and X. Chen, "Greenedge: joint green energy scheduling and dynamic task offloading in multi-tier edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4322–4335, Apr. 2022, doi: 10.1109/TVT.2022.3147027.
[5] J. Perez-Valero, A. Banchs, P. Serrano, J. Ortin, J. Garcia-Reinoso, and X. Costa-Perez, "Energy-aware adaptive scaling of server farms for NFV with reliability requirements," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4273–4284, May 2024, doi: 10.1109/TMC.2023.3288604.
[6] X. Wang, H. Xing, F. Song, S. Luo, P. Dai, and B. Zhao, "On jointly optimizing partial offloading and SFC mapping: a cooperative dual-agent deep reinforcement learning approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 8, pp. 2479–2497, Aug. 2023, doi: 10.1109/TPDS.2023.3287633.
[7] J. Xu, R. Ding, X. Liu, X. Li, J. Grundy, and Y. Yang, "Edgeworkflow: one click to test and deploy your workflow applications to the edge," *Journal of Systems and Software*, vol. 193, p. 111456, Nov. 2022, doi: 10.1016/j.jss.2022.111456.

[8] R. Farahani, A. Bentaleb, C. Timmerer, M. Shojafar, R. Prodan, and H. Hellwagner, "SARENA: SFC-enabled architecture for adaptive video streaming applications," in *IEEE International Conference on Communications*, IEEE, May 2023, pp. 864–870. doi: 10.1109/ICC45041.2023.10279262.
[9] B. Hu, Z. Cao, and M. C. 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.
[10] P. K. Thiruvasagam, A. Chakraborty, A. Mathew, and C. S. R. Murthy, "Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwarized 5G networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1491–1507, Jun. 2021, doi: 10.1109/TNSM.2021.3056917.
[11] L. Ye, L. Yang, Y. Xia, and X. Zhao, "A cost-driven intelligence scheduling approach for deadline-constrained IoT workflow applications in cloud computing," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16033–16047, May 2024, doi: 10.1109/JIOT.2024.3351630.
[12] S. A. Alsaidy, A. D. Abbood, and M. A. Sahib, "Heuristic initialization of PSO task scheduling algorithm in cloud computing," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2370–2382, Jun. 2022, doi: 10.1016/j.jksuci.2020.11.002.
[13] R. Lin, L. He, S. Luo, and M. Zukerman, "Energy-aware service function chaining embedding in NFV networks," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1158–1171, Mar. 2023, doi: 10.1109/TSC.2022.3162328.
[14] Z. Ahmad *et al.*, "Scientific workflows management and scheduling in cloud computing: taxonomy, prospects, and challenges," *IEEE Access*, vol. 9, pp. 53491–53508, 2021, doi: 10.1109/ACCESS.2021.3070785.
[15] 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*, vol. 25, no. 1, pp. 550–560, Jan. 2022, doi: 10.11591/ijeecs.v25.i1.pp550-560.
[16] F. Yao, C. Pu, and Z. Zhang, "Task duplication-based scheduling algorithm for budget-constrained workflows in cloud computing," *IEEE Access*, vol. 9, pp. 37262–37272, 2021, doi: 10.1109/ACCESS.2021.3063456.
[17] V. Sindhu, M. Prakash, and P. Mohan Kumar, "Energy-efficient task scheduling and resource allocation for improving the performance of a cloud–fog environment," *Symmetry*, vol. 14, no. 11, p. 2340, Nov. 2022, doi: 10.3390/sym14112340.
[18] A. S. Abohamama, A. El-Ghamry, and E. Hamouda, "Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment," *Journal of Network and Systems Management*, vol. 30, no. 4, p. 54, Oct. 2022, doi: 10.1007/s10922-022-09664-6.
[19] 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, p. 53, Dec. 2021, doi: 10.1186/s13677-021-00264-4.
[20] 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.
[21] L. Liu, H. Wang, Y. Liu, and M. Zhang, "Task scheduling model of edge computing for AI flow computing in internet of things," in *Proceedings - 2022 Global Conference on Robotics, Artificial Intelligence and Information Technology, GCRAIT 2022*, IEEE, Jul. 2022, pp. 256–260. doi: 10.1109/GCRAIT55928.2022.00061.
[22] J. K. Konjaang and L. Xu, "Cost optimised heuristic algorithm (COHA) for scientific workflow scheduling in IaaS cloud environment," in *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, BigDataSecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, IEEE, May 2020, pp. 162–168. doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00038.
[23] J. K. Konjaang and L. Xu, "Multi-objective workflow optimization strategy (MOWOS) for cloud computing," *Journal of Cloud Computing*, vol. 10, no. 1, p. 11, Dec. 2021, doi: 10.1186/s13677-020-00219-1.
[24] J. Masoudi, B. Barzegar, and H. Motameni, "Energy-aware virtual machine allocation in DVFS-enabled cloud data centers," *IEEE Access*, vol. 10, pp. 3617–3630, 2022, doi: 10.1109/ACCESS.2021.3136827.
[25] L. Rui *et al.*, "SFC orchestration method for edge cloud and central cloud collaboration: QoS and Energy Consumption Joint Optimization Combined With Reputation Assessment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 10, pp. 2735–2748, Oct. 2023, doi: 10.1109/TPDS.2023.3301670.
[26] N. Bacanin, M. Zivkovic, T. Bezdan, K. Venkatachalam, and M. Abouhawwash, "Modified firefly algorithm for workflow scheduling in cloud-edge environment," *Neural Computing and Applications*, vol. 34, no. 11, pp. 9043–9068, Jun. 2022, doi: 10.1007/s00521-022-06925-y.
[27] S. Mangalampalli *et al.*, "Multi objective prioritized workflow scheduling using deep reinforcement based learning in cloud computing," *IEEE Access*, vol. 12, pp. 5373–5392, 2024, doi: 10.1109/ACCESS.2024.3350741.
[28] K. D. Hanabaratti and S. F. Rodd, "Evolutionary Computing based web service composition technique for scheduling of workload under cloud environment," *Indian Journal of Science and Technology*, vol. 15, no. 2, pp. 69–80, Jan. 2022, doi: 10.17485/ijst/v15i2.1806.
[29] A. Nelli and R. Jogdand, "SLA-WS: SLA-based workload scheduling technique in multi-cloud platform," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 8, pp. 10001–10012, Aug. 2023, doi: 10.1007/s12652-021-03666-z.

BIOGRAPHIES OF AUTHORS

Sheetal Bagali holds Ph.D. degree in Underwater Sensor network Communication, M.Tech degree in VLSI & Embedded System 2, and B.E in Electronics and Communication. Currently working as Associate Professor at Sir MVIT, Bengaluru. Over all Teaching experience is 9 years. Her research interests include VLSI, Underwater Sensor Networks, and IoT. She has published over 15 papers in various journals and conferences and published 3 patents. She has reviewed manuscripts for various International Conferences and Journals. She can be contacted at email: sheetalbagali_ece@sirmvit.edu.

Pradeep Kumar Siddana is pursuing his Doctor of Philosophy in the Department of Electronics and Communication Engineering at Nitte Meenakshi Institute of Technology, located in Bengaluru, Karnataka, India. His academic contributions include more than 20 articles in journals and conferences indexed by SCOPUS. His notable achievements include winning the IEEE Mysurucon-2022 Champion Award, the Outstanding Research Contribution award from IEEE ComSoc, and managing an AESS fund grant valued at Rs. 41,12,000 for projects on aerospace and electronics applications at the same institution. Kumar's research spans several areas, such as Communication, Network Security, VLSI, Receiver Design, and Microcontrollers. He holds a Graduate Membership with both IEEE and IEEE AESS. He can be contacted at email: pradeepkumar.s@nmit.ac.in.

Dr. Poongothai C Dr. Poongothai C received her Bachelor of Engineering Degree in Electronics and Communication Engineering, Master of Engineering in VLSI design and Doctorate degree from Visvesvaraya Technological University, Karnataka in 2023. She has 17 years of teaching experience and at present she is working as Associate Professor in Sir M Visvesvaraya Institute of Technology, Bangalore, Karnataka. Her research interests include free space optics and networks. She is a life member of ISTE (Indian society for Technical Education). She can be contacted at email: pungathai_ec@sirmvit.edu.

Dr. Kavita Avinash Patil received her BE in Electronics and Communication Engineering, M. tech in VLSI Design and Embedded system and Ph.D. degree in Electronics and Communication Engineering from Visvesvaraya Technological University in June 2023. Currently she is working as Assistant Professor in the Department of Electronics & Telecommunication Engineering, at M. S. Ramaiah Institute of Technology. Her research interest includes Biomedical Engineering, machine learning, Data Science and artificial intelligence. She can be contacted at email: kavitapatil@msrit.edu.

Dr. Vani Bengaluru Purushothama has received Bachelor of Engineering in E&C, M. Tech in Digital Electronics and Ph.D. in 5G and cognitive networks. She has overall Teaching experience of 16 years and currently working as Associate Professor at Sai Vidya College of Engineering, her research area of specialization includes Wireless Communication, Computer Networks and IoT. To her credit she has total 16 publications out of which 3 are indexed in Scopus Indexed Journals, 5 are International Conference Proceedings, 8 are National Conference Proceedings. Total of 3 Patents are published in her name and guided more than 20 UG projects. She is also Resource person and conducted workshop for student on NS2 and Arduino. She is a member of different professional bodies like Institute of Engineers, Member of IEEE and ISTE. She can be contacted at email: vaniganesh2003@gmail.com.

Dr. Ashwini Kumari P has received Bachelor of Engineering in E&E, M.E. in Power and Energy and Ph.D. in Electrical Engineering from VIT University, India. She has contributed to numerous reputable conferences and journals, both as an author and reviewer. Her research interest encompass solar photovoltaic systems, application of artificial intelligence in engineering, optimization techniques, the development of Maximum Power Point Tracking (MPPT) schemes and design of inverters for grid-connected applications. She was project coordinator for a significant Solar Rooftop Project at Reva University valued at 1.5 crores with a capacity of 350kW. She holds 8 published patents and has been awarded "Best paper" honors at multiple conferences. She can be reached at ashwinikumari.p@reva.edu.in.