Indonesian Journal of Electrical Engineering and Computer Science
Vol. 38, No. 1, April 2025, pp. 546~554
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v38.i1.pp546-554

Energy and cost-aware workload scheduler for heterogeneous cloud platform

Manjunatha Shivanandappa[1], Naveen Kumar Chowdaiah[2], Swetha Mysore Devaraje Gowda[1], Rashmi Shivaswamy[3], Vadivel Ramasamy[4], Subramani Suryakumar Prabhu Vijay[5]
[1]Department of Computer Science and Engineering, BNM Institute of Technology affiliated with Visvesvaraya Technological University, Bengaluru, India
[2]Department of Computer Science and Engineering, Global Academy of Technology, Bengaluru, India
[3]Department of Computer Science and Engineering (Data Science), Dayananda Sagar College of Engineering, Bengaluru, India
[4]Department of Artificial Intelligence and Data Science, Nitte Meenakshi Institute of Technology, Bengaluru, India
[5]Software Engineer and Data Analyst, Navshya Technologies, Bengaluru, India

Article Info

Article history:

Received Jul 6, 2024
Revised Oct 15, 2024
Accepted Oct 30, 2024

Keywords:

Cloud computing
Cost reduction
Energy efficiency
Makespan efficiency
Resource provisioning
Workload scheduling

ABSTRACT

Parallel scientific workloads, often represented as directed acyclic graphs (DAGs), consist of interdependent tasks that require significant data exchange and are executed on distributed clusters. The communication overhead between tasks running on different nodes can lead to substantial increases in makespan, energy usage, and monetary costs. Therefore, there is potential to balance communication and computation to reduce these costs. In this paper, we introduce an energy and cost-aware workload scheduler (ECAWS) tailored for executing parallel scientific workloads, generated by the internet of things (IoT), in a heterogeneous cloud environment. The performance of the proposed ECAWS model is evaluated against existing models using the Inspiral scientific workload. Results indicate that ECAWS outperforms other models in reducing makespan, costs, and energy consumption.

Corresponding Author:

Manjunatha Shivanandappa
Department of Computer Science and Engineering
BNM Institute of Technology affiliated with Visvesvaraya Technological University
Bangalore, Karnataka, India
Email: manjunaths@bnmit.in

1. INTRODUCTION

Cloud computing [1], coupled with virtualization technology, opens up extensive research opportunities across numerous domains and applications. As global data expands, the need for automated data processing is increasingly apparent. This is particularly relevant in fields such as Bioinformatics and Astronomy, where substantial data is gathered for research purposes. Often, this data is managed as scientific workloads.

Scientific workloads [2], which are frequently modeled as directed acyclic graphs (DAGs), involve interdependent tasks that communicate via file exchanges. The output from one task often serves as the input for another. These workloads can comprise thousands of tasks and are typically executed on large-scale parallel or distributed systems, including cloud computational platforms [3]. Such systems allow for parallel processing of independent tasks, thereby reducing overall costs and execution times (makespan). However, scheduling these tasks in cloud environments is a complex, non-polynomial problem [4]. To address this, platforms like cloudsim [5], [6], and edge-workload [7] schedulers have recently been employed to manage incoming tasks as shown in Figure 1.

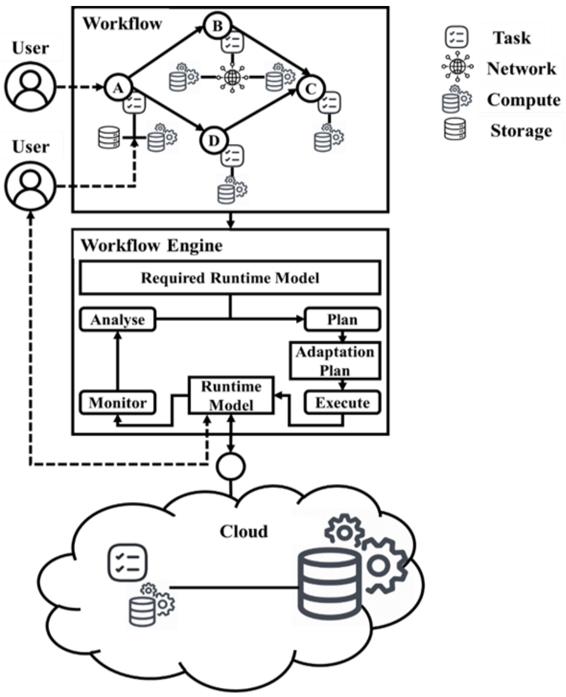

Figure 1. The architecture of workflow scheduling in a homogenous cloud platform [2]

Various algorithms have been developed for task scheduling [8]. These include particle swarm optimization (PSO) [9], ant colony optimization (ACO) [10], heterogeneous earliest time first (HEFT) [11], enhanced HEFT [12], and energy-cost-aware [13] schedulers employing different optimization strategies. More details of different scheduling methods have been discussed in section 2. While these methods [14], [15] have improved performance, they often fall short when dealing with large scientific parallel workloads. They tend to struggle with reducing both cost and makespan during computations. In this paper, we introduce a new model: the energy and cost-aware workload scheduler (ECAWS). This model is designed for parallel workload execution in heterogeneous cloud environments. Its primary goals are to minimize energy consumption, reduce costs, and meet task deadlines (makespan).

The significance of our research lies in its development of a scheduler model that enhances performance by lowering makespan, energy usage, and computational costs for scientific parallel workloads. We conduct a comparative analysis of various workload scheduler models that use different methodologies for managing scientific workloads. The results demonstrate that ECAWS significantly improves performance in terms of cost reduction, energy efficiency, and makespan reduction.

The structure of the paper is as follows: section 2 reviews different models, algorithms, architectures, and methodologies applied to the execution of scientific workloads. Section 3 introduces our proposed architecture and resource provisioning model, focusing on resource allocation for handling data-intensive scientific tasks. Section 4 presents an evaluation of the results and a comparison with existing models. Finally, section 5 offers a concise conclusion summarizing the research findings.

2. RELATED WORK

This section studies different workload scheduling for cloud and edge-cloud platforms [14], [15]. Yao *et al.* [16] introduced a task-duplication-based scheduling algorithm (TDSA) aimed at reducing costs and makespan within cloud environments. Their approach includes two primary methods and was tested on both random and scientific workloads. The results indicated a 31.6% reduction in cost and a 17.4% decrease in makespan. Sindhu *et al.* [17], additionally, the algorithm addresses energy consumption and overall computational costs, enhancing system performance in edge-fog computing environments. It utilizes DAGs for task scheduling and incorporates a Markov decision process for optimal resource allocation. The algorithm demonstrated superior performance compared to existing models.

Abohamama *et al.* [18], developed a task-scheduling algorithm for cloud-fog platforms, framing the scheduling problem as a permutation-based optimization challenge. They employed an enhanced genetic algorithm (GA) to allocate tasks to virtual machines with optimal resources and execution times. Their experiments, comparing the proposed algorithm with methods such as best-fit, first-fit, bees' life algorithm, and GA, showed improvements in cost, total computation time, and failure rate. Movahedi *et al.* [19] proposed a task scheduling model designed to minimize energy consumption and execution time in fog computing platforms. Their approach includes an architecture for managing incoming tasks and employs integer-linear programming (ILP) alongside a chaotic whale optimization algorithm. Comparisons with GA, artificial-bee-colony algorithms, and PSO revealed that their model outperformed these existing systems.

Shashank *et al.* [20] introduced a deep reinforcement learning (DRL) algorithm for IoT task scheduling in fog-based environments. Their method addresses task scheduling into virtual machines using a dual queuing technique, aiming to reduce cost, energy consumption, and makespan. Liu *et al.* [21] presented a PSO algorithm for task scheduling in edge computing environments. This algorithm aims to reduce computation costs and was evaluated using the CloudSim platform. Results indicated that their approach effectively optimized computation time and cost compared to four other task scheduling algorithms. Naveen and Annapurna [22] developed a scheduling algorithm and resource provisioning model to cut costs during task scheduling. Their method involves breaking down workload tasks into smaller sub-tasks to expedite execution and meet deadlines. Evaluations of their model, focusing on scientific workloads, showed faster virtual machine allocation and minimal cost.

Konjaang and Xu [23] proposed a multi-objective workload optimization strategy (MOWOS) to reduce makespan and cost. They introduced two algorithms namely maximum virtual machine and minimum virtual machine, to manage workload tasks. The MOWOS approach achieved an 8% reduction in cost and a 10% decrease in makespan. Masoudi *et al.* [24] tackled energy constraints through effective virtual machine allocation strategies. Studies [25], [26] underscored the role of edge computing in improving service quality and energy efficiency. Mangalampalli *et al.* [27] emphasized the need for multi-objective optimization using DRL to reduce makespan and energy consumption, though effective virtual machine placement according to quality-of-service (QoS) requirements remains an area for improvement, leading to potential delays and increased makespan.

3. PROPOSED METHODOLOGY

This section presents a novel scheduler named ECAWS for the execution of parallel workloads in the heterogeneous cloud platform. The section explains the workload and heterogeneous architecture adopted for scheduling optimization as shown in Figure 2. The ECAWS constructs different metrics namely computation cost, reconfiguration cost, and communication cost. Then, multi-objective minimization optimization metrics are presented and optimized using DRL to reduce overall energy with minimal time and cost.

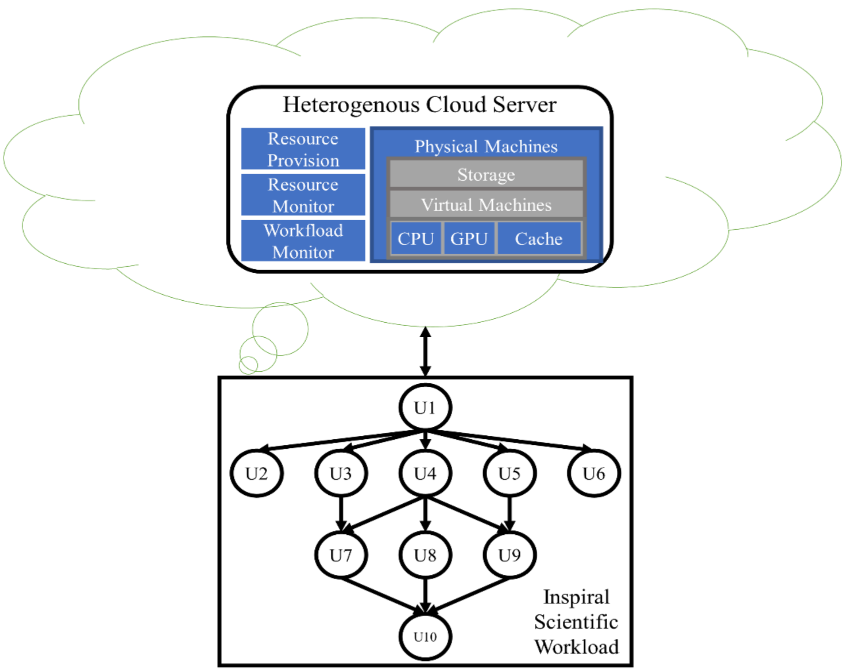

Figure 2. Architecture of heterogeneous cloud platform for energy and cost-aware workload scheduling

3.1. Workload and heterogenous computational architecture classification

The Inspiral workload significantly uses lots of memory and computational resources. The overall size of the Inspiral workload is represented through parameter L_b measured in bits. According to n computational machines, the data is segmented into N predefined numbers considering both idle and active computational machines. As the Inspiral workload is composed of multi-level QoS dependencies among tasks; thus, the task must complete the execution within time S_s. In meeting energy efficiency, the computational resource allocated considering the respective physical computational platform x is obtained in (1).

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

In (1), $freq_x^I$ expresses the idle state frequency parameter, $freq_x^\uparrow$ expresses maximal operating frequency for the execution of Inspiral workload, E_x^I, $P_a(x)$ defines non-idle state energy consumption of computational platform, $P_a(x)$ defines overall computational machines that are actively participating in Inspiral workload task execution and $C_e(x)$ expresses parameter defining load factor. In this work the parameter ${R_p}^\uparrow$ defines maximal processing capability considering the heterogenous multi-core resource optimization nature; thus, ${R_p}^\uparrow = freq_x^\uparrow$.

3.2. Computation, reconfiguration, and communication cost metrics

In reducing the energy consumption and meeting makespan minimization to reduce the overall cost of execution of Inspiral workloads this work employs the dynamic voltage and frequency scaling (DVFS) model [11] according to the multi-core resource availability. The parameter $freq_{disc}$ expresses both lower and higher operating frequency and the maximal frequency of the idle-state computational node is measured in (2).

$$freq^\uparrow \triangleq freq_Z > freq_{Z-1} > freq_{Z-2} > \cdots > freq_1 > freq^I \triangleq freq_0 \quad (2)$$

The current method cannot satisfy dynamic frequency optimization considering multi-level service optimization. Performing dynamic optimization is challenging as it needs to measure different parameters like storage, memory, and computational processing elements considering both idle and active states. Thus, the dynamic energy consumption $E_{dynamic}$ is measured through (3).

$$E_{dynamic} = P_a * C_e * freq * vs^2 \quad (3)$$

In (3), $E_{dynamic}$ expresses the parameter to measure processor dynamic energy consumption operating on a physical computational platform, $freq$ defines the physical computational platform frequency level, and vs^2 represents corresponding voltage; the association between voltage and frequency is obtained in (4).

$$freq = \mathbb{C} * \left[\frac{vs^{-1}}{(vs - vs_\tau)^{-2}}\right] \quad (4)$$

In (4), $\mathbb{C}$ defines the weight optimization parameter which remains static throughout the Inspiral workload task execution, vs_τ expresses lesser weighted voltage than required input voltage vs. Throguh optimization of (3) and (4), the idle state task computation cost $SUM_{C-computation}(x)$ by considering $E^I \geq 0$ is measured in (5).

$$SUM_{C-computation}(x) \triangleq \sum_{y=0}^{Z} {P_a}' * C_e * s_{xy} * \frac{1}{freq_y^{-3}},\ x = 1, 2, \ldots, N \quad (5)$$

In (5), ${P_a}' = \mathbb{C}^{-1} * P_a$, s_{xy} defines the overall makespan when executed in a physical computational platform operating at frequency $freq_y$. The parameter Z defines the available frequency level in the respective processing element considering $Z + 1$ bounds when x in s_{xy} ranges between 1 to N and y ranges between 0 to Z. Let's assume that the frequency changes from $freq_1$ to $freq_2$ to meet the Inspiral workload task deadlines; the reconfiguration cost $SUM_{C-reconfiguration}$ is measured through (6).

$$SUM_{C-reconfiguration}(freq_1; freq_2) = \mathcal{E}_c * \frac{1}{(freq_1 - freq_2)^{-2}} \quad \ldots \quad Joule \quad (6)$$

In (6), $\mathcal{E}_c\, Joules/Hz^2$ expresses the computational cost for changing the frequency levels according to task deadline requirements. During the reconfiguration process, extensive communication cost is involved to perform data exchange considering trannfer rate $TrnsRate_x$. Therefore, the communication cost E_x^{C-comm} is measured in (7):

$$E_x^{C-communication} \equiv E_{TotalTime}^{C-communication}(x) + E_{TransmissionRate}^{C-communication}(x) \quad (7)$$

where $E_{TotalTime}^{C-communication}(x)$ expresses the parameter defining switching energy cost and $E_{TrnsRate}^{C-communication}(x)$ expresses the parameter defining communication energy. The network may induce a certain load and delay; however, considering optimal communication the total computation cost is measured in (8).

$$E_x^{C-communication}(TransmissionRate_x) = \delta_x(\overline{\mathcal{T}}_x * TransmissionRate_x)^2 + E_x^I,\ x = 1\ to\ N \quad (8)$$

In (8), the parameter δ_x is measured in (9):

$$\delta_x \triangleq \left(R_{gain}\right)^{-1} * \left(\mathcal{K}^{-1} * \sqrt{\frac{2*\theta}{3}}\right)^2, x = 1\ to\ N \quad (9)$$

where parameter $\mathcal{K}$ defines the maximum segmentation level considering noisy coding gain R_{gain}. Thus, considering the transfer delay is measured in (10):

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

Using (10), the $TransmissionDelay(x)$, the communication cost can be finally established in (11).

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

3.3. Energy and cost-aware workload scheduler model

This section introduces a novel ECAWS employing multi-objective optimization. The computation cost in (5), reconfiguration cost in (6), and communication cost in (11) are optimized through the below minimization function $\mathcal{C}$ defined in (12).

$$\mathcal{C} = \min[SUM_{C-computation}(x) + SUM_{C-reconfiguration}(freq_1; freq_2) + SUM^{C-communication}(x)] \quad (12)$$

The $\mathcal{C}$ denotes the multi-objective optimization parameter; the overall cost of the computation in a heterogenous cloud platform through the usage of a machine learning model [28], [29] adopting deep learning evolutionary optimization model [30], [31] namely the enhanced DRL model [27] for efficient scheduling of workload tasks and achieving better performance and reducing cost as shown in the result section.

4. RESULT AND ANALYSIS

The proposed ECAWS was tested using the Inspiral scenario to assess its performance regarding makespan, energy consumption, and cost. The ECAWS algorithm was evaluated against two other models: the energy-minimized scheduling (EMS) [11] and the multi-objective DRL-based workload scheduler (MODRLWS) [27]. The evaluation involved four tasks from the Inspiral dataset, including Inspiral 30 and Inspiral 100. All experiments were conducted on a system equipped with an Intel® core i7 processor, 16 GB of RAM, and running Windows 10 (64-bit). The cloudsim platform was utilized to simulate and assess the performance of the proposed ECAWS model alongside the state-of-the-art scheduling algorithms.

4.1. Makespan performance

Figures 3 and 4 illustrate the makespan for Inspiral 30 and Inspiral 100, respectively. The results reveal that the EMS model resulted in a longer makespan compared to the MODRLWS model. The ECAWS model demonstrated a substantial reduction in makespan—42.12% for Inspiral 30 and 61.44% for Inspiral 100—when compared to MODRLWS. The overall makespan of execution is reduced employing (5) and later the parameter is optimized using an enhanced DRL model contributing to a significant reduction of makespan using ECAWS in comparison with EMS and MODRLWS. This reduction is attributed to the enhanced optimization provided by the DRL model used in ECAWS.

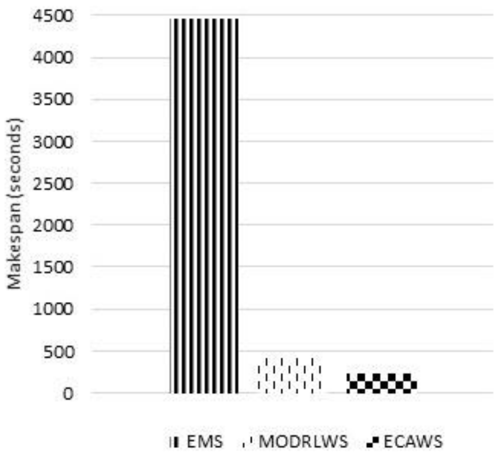

Figure 3. Makespan for inspiral 30

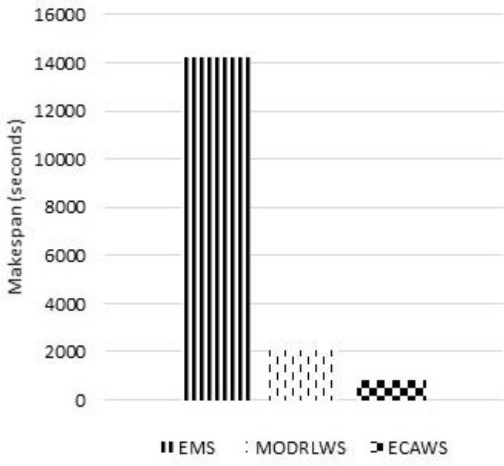

Figure 4. Makespan for inspiral 100

4.2. Energy consumption performance

Figures 5 and 6 display the energy consumption for Inspiral 30 and Inspiral 100. The EMS model showed higher energy consumption than both MODRLWS and ECAWS. Although MODRLWS consumed less energy than EMS, the ECAWS model achieved a reduction of 3.8% for Inspiral 30 and 3.15% for Inspiral 100 in energy consumption compared to MODRLWS. The overall energy of execution is reduced by employing (6) and later the parameter is optimized using an enhanced DRL model contributing to a significant reduction of energy using ECAWS in comparison with EMS and MODRLWS. The improvements are attributed to the efficient optimization techniques employed in ECAWS.

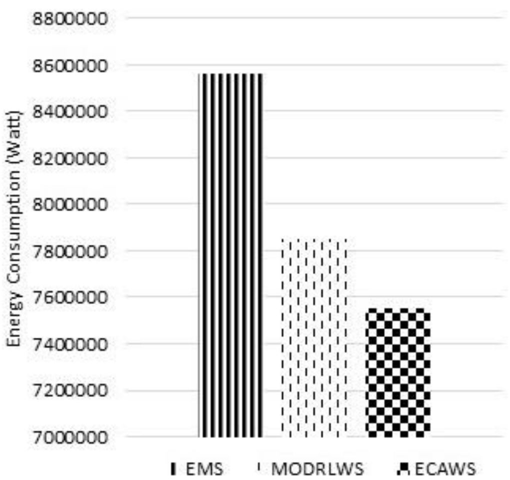

Figure 5. Energy consumption for inspiral 30

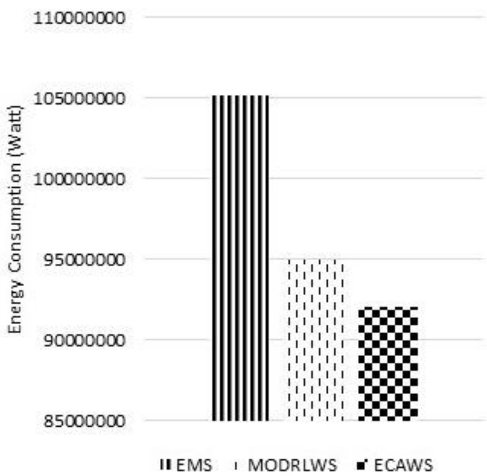

Figure 6. Energy consumption for inspiral 100

4.3. Computation cost

Figures 7 and 8 depict the computation costs for Inspiral 30 and Inspiral 100. The results indicate that the proposed ECAWS model offers a significant cost advantage over existing models. Specifically, ECAWS reduced computation costs by 64.95% and 70.66% compared to MODRLWS for Inspiral workloads of sizes 30 and 100, respectively. The overall cost of execution is reduced by employing (10) and later the parameter is optimized using an enhanced DRL model contributing to a significant reduction of cost using ECAWS in comparison with EMS and MODRLWS. This cost reduction is a result of the effective optimization strategies incorporated into ECAWS.

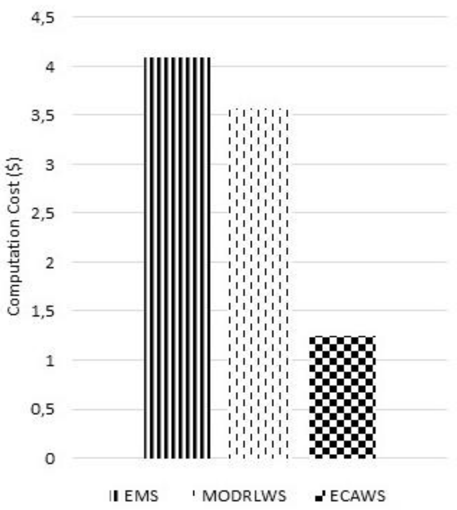

Figure 7. Computation cost for inspiral 30

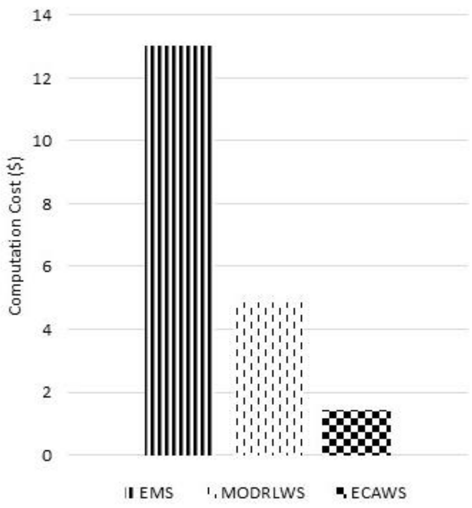

Figure 8. Computation cost for inspiral 100

5. CONCLUSION

In summary, the ECAWS model shows superior performance in reducing makespan, energy consumption, and cost compared to EMS and MODRLWS. EMS, while focusing on energy and cost reduction, did not effectively address makespan reduction. The MODRLWS model provided improvements but fell short in overall cost reduction across different workload sizes. The ECAWS model successfully tackles these issues, offering better overall performance in makespan, energy, and cost reduction. Looking ahead, this research can be extended to other scientific workloads such as Montage and Sipht. Since parallel scientific workloads, which are typically represented as DAGs, involve significant data exchanges and are executed across distributed clusters, optimizing communication and computation remains a key area for future exploration. The ECAWS model, tailored for IoT-generated parallel scientific workloads in heterogeneous cloud platforms, has demonstrated its effectiveness through the Inspiral workload, highlighting its potential for broader applications.

REFERENCES

[1] S. Qin, D. Pi, Z. Shao, Y. Xu and Y. Chen, "Reliability-aware multi-objective memetic algorithm for workload scheduling problem in multi-cloud system," *in IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1343-1361, 2023, doi: 10.1109/TPDS.2023.3245089.

[2] H. Lahza, B. R. Sreenivasa. H. F. M. Lahza, and J. S. Shreyas, "Adaptive multi-objective resource allocation for edge-cloud workflow optimization using deep reinforcement learning," *Modelling*, vol. 5, pp. 1298-1313, 2024, doi: 10.3390/modelling5030067.

[3] M. Menaka and K. Sendhil, "Workload scheduling in cloud environment – challenges, tools, limitations and methodologies: a review," *Measurement: Sensors*, 24, p. 100436, 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," *in IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4322-4335, 2022, doi: 10.1109/TVT.2022.3147027.

[5] J. Perez-Valero, A. Banchs, P. Serrano, J. Ortín, J. Garcia-Reinoso, and X. Costa-Pérez, "Energy-aware adaptive scaling of server farms for NFV with reliability requirements," *in IEEE Transactions on Mobile Computing*, vol. 1, pp. 1, 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, 2023, doi: 10.1109/tpds.2023.3287633.

[7] X. Jia, D. Ran, L. Xiao, L. Xuejun, G. John, and Y. Yun, "EdgeWorkload: One click to test and deploy your workload applications to the edge," *Journal of Systems and Software*, vol. 193, pp. 111456, 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," *ICC 2023 - IEEE International Conference on Communications, Rome, Italy*, pp. 864-870, 2023, doi: 10.1109/ICC45041.2023.10279262.

[9] 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, 2370-2382, 2023, doi: 10.1016/j.jksuci.2020.11.002.

[10] L. Ye, L. Yang, Y. Xia, and X. Zhao, "A cost-driven intelligence scheduling approach for deadline-constrained IoT workload applications in cloud computing," *in IEEE Internet of Things Journal*, vol. 1, pp. 1, 2024, doi: 10.1109/JIOT.2024.3351630.
[11] B. Hu, Z. Cao, and M. Zhou, "Energy-minimized scheduling of real-time parallel workloads on heterogeneous distributed computing systems," *in IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2766-2779, 2022, doi: 10.1109/TSC.2021.3054754.
[12] 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," *in IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1491-1507, 2021, doi: 10.1109/TNSM.2021.3056917.
[13] R. Lin, H. Liu, 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, 2023, doi: 10.1109/tsc.2022.3162328.
[14] Z. Ahmad, A. Jehangiri, M. Ala'anzy, M. Othman, R. Latip, and A. Umar, "Scientific workflows management and scheduling in cloud computing: taxonomy, prospects, and challenges," *IEEE Access*, vol. 9, pp. 53491-53508, 2022, 10.1109/ACCESS.2021.3070785.
[15] B. Nidhi and S. Ajay, "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, pp. 550, 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 workloads in cloud computing," *in IEEE Access*, vol. 9, pp. 37262-37272, 2021, doi: 10.1109/ACCESS.2021.3063456.
[17] V. Sindhu, M. Prakash, and P. Mohan, "Energy-efficient task scheduling and resource allocation for improving the performance of a cloud–fog environment," *Symmetry*, vol. 14, pp. 2340, 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, pp. 54, 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, pp. 53, 2021, doi: 10.1186/s13677-021-00264-4.
[20] S. Shashank, M. S. Elhadi, 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," *2022 Global Conference on Robotics, Artificial Intelligence and Information Technology (GCRAIT)*, pp. 256-260, 2022, doi: 10.1109/GCRAIT55928.2022.00061.
[22] C. Naveen and D. Annapurna, "Resource provisioning model for executing realistic workload in heterogenous internet of things environment," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, pp. 318, 2023, doi: 10.11591/ijeecs.v32.i1.pp318-327.
[23] J. K. Konjaang and L. Xu, "Multi-objective workload optimization strategy (MOWOS) for cloud computing," *Journal of Cloud Computing*, vol. 10, pp. 11, 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," *in IEEE Access*, vol. 10, pp. 3617-3630, 2022, doi: 10.1109/ACCESS.2021.3136827.
[25] L. Rui, S. Chen, S. Wang, Z. Gao, X. Qiu, W. Li, and S. Guo, "SFC orchestration method for edge cloud and central cloud collaboration: QoS and energy consumption joint optimization combined with reputation assessment," *in IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 10, pp. 2735-2748, 2023, doi: 10.1109/TPDS.2023.3301670.
[26] N. Bacanin, M. Zivkovic, T. Bezdan, K. Venkatachalam, and M. Abouhawwash, "Modified firefly algorithm for workload scheduling in cloud-edge environment," *Neural Computing and Applications*, vol. 34, no. 11, pp. 9043–9068, 2022, doi: 10.1007/s00521-022-06925-y.
[27] S. Mangalampalli, S. S. Hashmi, A. Gupta, G. R. Karri, T. Chakrabarti, P. Chakrabarti, K. V. Rajkumar, and M. Margala, "Multi objective prioritized workload scheduling using deep reinforcement based learning in cloud computing," *in IEEE Access*, vol. 12, pp. 5373-5392, 2024, doi: 10.1109/ACCESS.2024.3350741.
[28] S. Manjunath, P. Malini, M. D. Swetha, and S. S. P. Vijay, "Tampering detection and segmentation model for multimedia forensic," *International Journal of Advanced Computer Science and Applications*, vol. 14, pp. 878-887, 2023, doi: 10.14569/IJACSA.2023.0140992.
[29] M. D. Swetha and C. R. Aditya, "Noise invariant convolution neural network for segmentation of multiple sclerosis lesions from brain magnetic resonance imaging," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 18, no. 13, pp. 38–55, 2022, doi: 10.3991/ijoe.v18i13.34273.
[30] S. Manjunath and P. Malini, "Efficient resampling features and convolution neural network model for image forgery detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, pp. 183, 2022, doi: 10.11591/ijeecs.v25.i1.pp183-190.
[31] S. Manjunath and P. Malini, "Extraction of image resampling using correlation aware convolution neural networks for image tampering detection," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, pp. 3033, 2022, doi: 10.11591/ijece.v12i3.pp3033-3043.

BIOGRAPHIES OF AUTHORS

Dr. Manjunatha Shivanandappa has 21 years of academic experience. Presently he is working as Professor in the Department of Computer Science and Engineering, BNM Institute of Technology, Bengaluru, Karnataka, India. He obtained his Ph.D. from VTU, Belagavi, Karnataka, India. His research area is Image Processing, Data Analytics. He has published more than 15 research articles in reputed international journals and conferences in India. He is a member of professional societies. He can be contacted at email: manjunaths@bnmit.in, manjunaths.dvg@gmail.com.

Dr. Naveen Kumar Chowdaiah completed Ph.D. from Visveswaraya Technological University (VTU) in the area of Internet of Things in 2023, received his MTech degree in 2011 from VTU and currently working as Associate 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, naveenkc31@gmail.com.

Dr. Swetha Mysore Devaraje Gowda has 12 years of Academic Experience. Awarded Ph.D. in 2023 from VTU, Belagavi, Karnataka. Her research interests in Image Processing, Machine Learning. She is working as an Associate Professor in the Department of Computer Science and Engineering at BNM Institute of Technology, Bengaluru. She has published 10 + research articles in referred journals and also a member of professional society (ISTE). She can be contacted at email: swetha.md@bnmit.in.

Dr. Rashmi Shivaswamy is an Incharge Head in Department of Computer Science and Engineering (Data Science) at Dayananda Sagar College of Engineering, Bengaluru. Her main research interests include Cloud Computing, Blockchain Technology, IOT, Machine Learning and Cyber security. She has earned Ph.D. in Computer Science and Engineering from VTU, Belgaum for her work in the area of Cloud Computing, in Jan 2019.She is an IEEE Senior member and has worked as a Managing Committee Member of Computer Society of India, Bangalore chapter for the year 2019-2020. She has published 2 Indian Patents and 20+ referred publications. email: rashmineha.s@gmail.com.

Dr. Vadivel Ramasamy is working as an Associate Professor in the Department of Artificial Intelligence and Data Science at Nitte Meenakshi Institute of Technology, Bangalore, Karnataka, India. He received Bachelor of Engineering Degree in Anna University, Chennai, India. He received Master of Engineering Degree in Sathyabama University, Chennai, India. He received Ph.D. Degree in Hindustan Institute of Technology and Science, Chennai, India. He has published more than 15 publications in the reputed Indexed Journals and International Conferences. His research areas are cloud computing, networking and advanced computing. He can be contacted at email: vadivel.r@nmit.ac.in, vadivelcse@gmail.com.

Subramani Suryakumar Prabhu Vijay is the research analyst and senior software developer at navshya technologies. Areas of interest includes wireless sensor networks, internet-of-things, cloud computing, network security, wireless communication network, image processing, image forensic. He has assisted in technical content writeup and software development various academics and industrial projects. Attented as as a guest lecturer and trainer in various workshops. He can be contacted at email: navshyatechnologies@gmail.com and prabhu.vijay23@gmail.com.