Educational Administration: Theory and Practice
2024, 30(11), 2601-2609
ISSN: 2148-2403
https://kuey.net/
Research Article

A Multi-Stage Optimization Framework for Efficient Cloud Workflow Scheduling

Deepak Pathak[1*], Puneet Dwivedi[2]

[1*,2]Department of Computer Science and Engineering, RKDF University, Bhopal, India
deep_325@yahoo.com*, puneet86.dwivedi@gmail.com

Corresponding author: Deepak Pathak
deep_325@yahoo.com*,

Citation: Deepak Pathak et al. (2024). A Multi-Stage Optimization Framework for Efficient Cloud Workflow Scheduling, *Educational Administration: Theory and Practice*, 30(11) 2601-2609, Doi: 10.53555/kuey.v30i11.10730

ARTICLE INFO

ABSTRACT

Cloud Workflow scheduling is still a challenging task with increasing workload over servers. This issue is due to dynamic nature of tasks arrived and their execution dependencies over heterogeneous resources. In such condition, efficient scheduling is required to virtual machines in cloud environment to reduce high latency requirements and efficient utilization of resources. Recently, researchers have contributed in this field and developed many optimization approaches to reduce the operational costs and make span time but still there is room for improvement as growing need of resources. Motivated by this, the paper presented a multi-stage and multi-objective based workflow scheduling algorithm. In first stage of the algorithm task priotization is performed using particle swarm optimization then resource allocation matrix is generated using ensemble learning and finally meta-heuristic optimization is used for allocating the optimal number of resources to respective priority queue tasks. The entire working model is developed on MATLAB and simulation is performed with varying number of tasks as well as virtual machines (VMs). The result was compared in terms of make span time with existing works and achieved better performance.

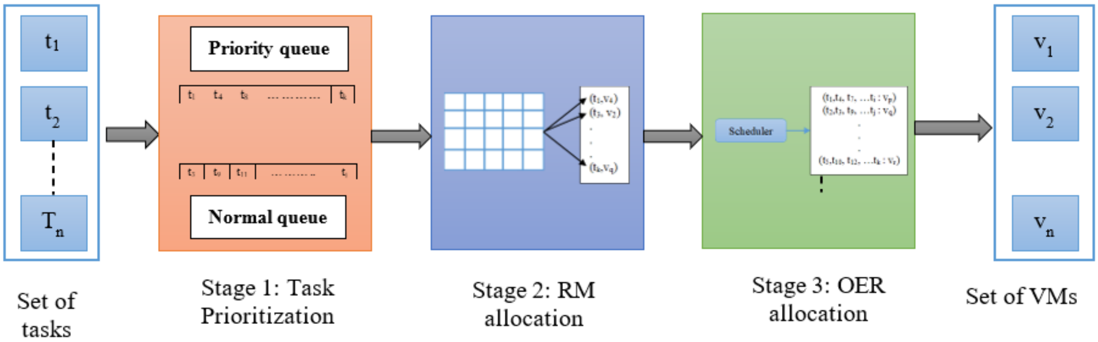

Fig.1. Multistage Workflow Scheduling in Cloud

Keywords: Cloud Computing, Workflow, Scheduling, Optimization, Multi-stage, Priority.

1. INTRODUCTION

In today's digital era, with increasing use of online applications or operations have increased the adoption of cloud computing. This has revolutionized business operations along with service delivery. For scalable and flexible computing resources that is aimed to enhance productivity, reduce infrastructure cost and adaptation with market demands [1][2]. Cloud computing is used over several domains that requires the resources for tasks such as data storage, processing, application hosting, etc. This increasing demand of resources will increase the cloud workflows that will make is quite complex task to manage and optimize the resource allocation process [3][4]. The conventional approaches for scheduling can handle workflow over cloud but face challenges to address the dynamic nature of modern workflows and therefore results in inefficient resource utilization with increased latency and ultimately increase the operational costs [5][6]. Therefore to handle such real-time dynamic and heterogeneous workflow tasks over cloud cannot be handled by conventional optimization approaches [7]-[12]. As it is known that conventional approaches are based on predefined rules and heuristics approach for resource allocation during task scheduling this will lead to suboptimal resource utilization and bottleneck the performance on occurrence of high workloads. Additionally, these approaches also needs manual tuning as well as adjustments that makes it difficult to handle high workflows over cloud and thus reduce the efficiency of scheduling [13]. This issue can be resolved by incorporating artificial intelligence (AI) for cloud workflow scheduling [14]-[18]. As AI use the data analytics and predictive modeling for deployment of autonomous decision-making. AI techniques can dynamically optimize the resource allocation by learning from past behaviors and adapting to dynamic and changing conditions [19]. There are several steps that can optimize the resource allocation [21]. By continuously monitoring available cloud resources for task allocation AI can be considered to be better solution. The process of such advance task profiling involves a systematic analysis of each task to recognize its resource demands and respective characteristics. Selection of appropriate scheduling approach is crucial as they are based on several factors such as task prioritization, deadline constraints, and optimal resource usage with dynamic data handling with heterogeneous scenarios. The advance optimization approach works in iteration phase that are based on fine-tuning parameters for enhancement of scheduling strategy. Therefore, this paper is dedicated to explore the role of optimization approach with AI to optimize resource allocation and improvement in workload distribution. The major contributions of the paper are:

- The paper presented a multi-stage workflow scheduling algorithm.
- In the proposed approach a hybridization of particle swarm optimization based task prioritization is performed. Then resource allocation matrix is generated using ensemble machine learning approach.
- Finally, the optimal resource allocation was performed using multi-objective meta-heuristic approach.
- The result was evaluated with varying number of tasks as well as varying numbers of VMs.

The rest part of the paper is organized as: section 2 presents the recent research contributions for scheduling workflows in cloud architecture. Then in section 3 proposed methodology is described with respective algorithms. Then results are presented in section 4 with comparative analysis. Finally, conclusion and future scope is presented in section 5.

2. LITERATURE REVIEW

Cheng et al. [22] proposed an optimization based scheduling approach for deep learning applications on GPU clusters. The designed approach was two-level optimization termed as DLBooster. In this approach the proposed model is composed of offloading key used for decoding tasks for FPGAs data preprocessing with back propagation computational workloads. Therefore, the proposed approach presented the better utilization for GPU allocation to respective workload. Ma et al. [23] proposed an online VM scheduling scheme (OSEC) that was based on energy and cost optimization deployed over cloud environments. The proposed approach used the Q-learning for efficient allocation of VMs. This significantly reduced the overall energy consumption, execution costs, and SLA violations. Jayanetti et al. [24] presented a workload balancing algorithm using deep reinforcement learning-based based scheduling framework for edge-cloud systems. This approach was hierarchical as well as hybrid in nature. The proposed approach used the proximal policy optimization for enhancement of performance with respect to energy consumption and execution time. Chen et al. [25] proposed deep reinforcement learning model to provide collaborative scheduling of heterogeneous workflows in cloud computing. The model reduced the make span time as well as cost. Mahmoud et al. [26] presented task scheduling algorithm that is multi-objective in nature. The model is based on decision tree for heterogeneous computing environment. Pradhan et al. [27] presented a scheduling algorithm with reinforcement learning and parallel particle swarm optimization. This algorithm addressed the load balancing with good accuracy. Zhou et al. [28] used deep-reinforcement-learning based two-stage scheduling model for IoT systems. The approach used the presorting technique with DRL for optimization of resource allocation and reduction in make span time. Mohammadzadeh et al. [30] proposed a hybrid multi-objective optimization algorithm that is composed of Seagull Optimization Algorithm (SOA) and Grasshopper Optimization Algorithm (GOA). Combination of this algorithm improved the convergence rate of optimization. This model was designed for handling scientific workflows over multi-cloud environments. The algorithm used the knee-point method for task scheduling. Dong et al. [31] proposed a hybrid algorithm with combination of evolutionary algorithm with DRL for addressing workflow scheduling challenges over cloud. The outcome of DRL was considered as input for evolutionary algorithm for optimization of tasks. Lu et al. [32] addressed the challenges of deadline constraints faced by the heterogeneous cloud resources. In this work, a market-driven workflow scheduling was used and proposed a Multi-Hierarchy Particle Swarm Optimization (MHPSO) algorithm that will use aggregation method and a hierarchical evolving process to reduce workflow load. The approach used the statistical analysis for comparison with existing approaches. Zeedan et al. [33] proposed a hybrid approach based on Enhanced Binary Artificial Bee Colony based Pareto Front (EBABC-PF). This approach introduced the Pareto Dominance strategy to optimize makespan time with cost. Reddy et al. [34] proposed an energy-efficient workflow scheduling using reinforcement learning. This approach also included the security enhancement using X-NOR Whirlpool hashing algorithm for scheduling scientific workloads. Shukla et al. [35] proposed a differential Evolution-Grey Wolf Optimization

(DE-GWO) approach in order to address slow convergence and low accuracy issue in the Grey Wolf Optimization (GWO) algorithm. The entire work is implemented for fog-cloud networks with improvement of accuracy as well as convergence speed. The technique used the weighted sum-based approach with make span time, cost, and energy consumption as objective function. The entire modeling was implemented to schedule scientific workloads. Despite the developments of ML and AI for cloud workflow load scheduling there are still some remaining research gaps areas. The one of the major problem is privacy and security concerns. Moreover issues such as minimal usage of CPU/GPUs, inefficiencies in exploring and converging to optimal solutions, latency, etc. Addressing these gaps is crucial for development of robust, scalable, and secure cloud workflow optimization strategies.

3. PROBLEM IDENTIFICATION

In cloud computing, factors such as CPU utilization, memory utilization, latency, response time, and speed are critical for virtualization. Virtual machines (VM) load depend on such critical factors and it is required to optimize these factors related with workloads at each hosting servers to avoid congestion as well as to reduce resource wastage. Each VM and host has load bounds and crossing these bounds lowers performance or wastes resources. Load balancing includes migration of workloads from overloaded to under loaded VMs. This combinatorial problem requires optimal VM selection and migration for dynamic workflow loads in cloud architecture.

4. METHODOLOGY USED

In scheduling, the workflows W are represented as directed acyclic graph (DAG) as $W = (T, E)$ where set of tasks T is represented as $\{t_1, t_2, t_3, \dots t_n\}$ with precendence dependencies as E. t_i represents individual task that is being executed on current instance of time. e_{ij} represents that j[th] task can be executed only when ith task is being completed. This represents the dependency of task over each other. In this paper, we have considered IaaS cloud for execution of multiple workflows with number of virtual machines. The optimization problem is considered to be as:

$$Objective\ function = \min\{MT, cost\} \tag{1}$$

Where, MT is makespan time and $cost$ is cost function required to allocate virtual machine.
In this paper, a multi-stage workflow scheduling is designed to resolve such issue. The working flowchart is presented in fig 1. In the first stage task priotization is performed then resource matrix (RM) generation is performed and finally optimal execution resource (OER) allocation is also performed. Each task is ranked in task priotization phase. Then based on task features, ensemble learning approach is adopted for generation of availability matrix of resources for respective task. Then finally multi-objective optimization is used to allocate the resources. All the steps are described in detailed further.

4.1 Task Prioritization

In this step, each task is assigned a rank for the given workflow. In this step, DAG of tasks are considered as $\{t_1, t_2, t_3, \dots, t_k, \dots . t_n\}$. Each task length T_{length}, task sensitivity T_s and number of child node T_{child} are considered for selecting priority of task. This priority decision is based on PSO optimization. This step selected the PSO because of its fast convergence towards optimal solution. Algorithm for task priotization is presented below:
Algorithm 1: Task Prioritization

1. Initialize the particle that represents the priority of tasks as X_i with velocity V_i (i.e., the change for priority ranking). The best known position locally and globally is represented as P_{best} and G_{best} respectively.
2. Then velocity updation is performed by $V_i(t+1) = w * V_i(t) + c_1 r_1 (P_{best} - X_i(t)) + c_2 r_2 (G_{best} - X_i(t))$. Where inertia weight is represented as w with cognitive and social coefficient of $c_1\ and\ c_2$ with random distribution of $r_1\ and\ r_2$.
3. Then position of particles are updated as $X_i(t+1) = X_i(t) + V_i(t+1)$
4. Then fitness is evaluated using function $f(x) = \sum_{k=1}^{n} (\alpha * T_{length} + \beta * T_s + \gamma * T_{child})$ with weight factors α, β, γ.
5. The entire process is repeated until the convergence is reached.

5.1 RM Allocation

In this step, the workflow tasks features are collected according to priority queue and regular queue. The collected features are: Task priority, task size, task cost, and assigned VM. These features are used generate a feature matrix as $F_m = \{f_1, f_2, \dots f_n\}$. Then ensemble model is used to train according to the presented feature vector generated out of historical data. The ensemble model predicts the availability matrix as $A_{ij} = f_{ensemble}(F_m)$.then this avilability matrix is used to allocate available resources.

5.2 OER Allocation

In this stage, optimal resources i.e., available VMs are allocated for execution of tasks i.e., priority queue tasks as well as regular queue tasks. In this step, the optimal resource allocation is performed using multi-objective meta-heuristic optimization approach is used. The multi-objective functions used here are based on make span time, total VMs availability and degree of imbalance. By combining the objectives into a single scalar value using a weighted sum approach is mathematically represented as:

$$F(a) = w_1 M - w_2 A + w_3 I \tag{2}$$

Where, w_1, w_2 *and* w_3 are respective weights.

$$M = \max\{exe_{time_i}\} \tag{3}$$

$$A = \sum_{i=1}^{T} \sum_{j=1}^{V} A_{ij} \delta\ (a_i == j) \tag{4}$$

$$I = \frac{\max\{exe_{time_i}\} - \min\{exe_{time_i}\}}{c_j} \tag{5}$$

Then chromosomes (VMs) from the current population of VMs are used to create a mating pool based on their fitness function values. Then rank-based selection is used to select the best VM. The better fitness value is used to select the best VM. Then pairs of chromosomes are combined from the mating pool to produce offspring. The crossover is then performed for new offspring's. Then, mutation operation is introduced for convergence to local optima. This process is repeated number of iterations for selection of optimal VMs for priority queue tasks as well as normal queue tasks.

6. RESULTS AND DISCUSSIONS

In this section, the performance of workflow scheduling and cloud load balancing is evaluated using the MATLAB simulation tool that is widely used for workflow scheduling simulation. MATLAB is used for designing the simulation modeling and experimentation. The evaluation is conducted on a 64-bit Windows system with an Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz (2.11 GHz) and 8GB RAM. For this following parameters are used:

Make span Time (MT): MT in cloud workflow scheduling is the total time taken to complete a set of tasks.

$$MT = \max\{exe_{time_i}\} \quad where, 0 \leq i \leq n \tag{6}$$

Average Delay Time (ADT): For execution of n tasks, the average time consumed is evaluated by ADT.

$$ADT = \frac{\sum_{i=1}^{n} exe_{time_i}}{n} \tag{7}$$

Degree of Imbalance (DI): It is represented as the difference between max and min response time with respect to ADT.

$$DI = \frac{Max(R_i) - Min(R_i)}{ADT} \tag{8}$$

Standard Deviation (SD): It is evaluated for estimation of distribution of load over the servers.

$$SD = \sqrt{\frac{\sum(R_i - ADT)2}{n}} \tag{9}$$

Fig 2 presents the make span time evaluation for the proposed algorithm with respect to variable workflow tasks. The result was evaluated for 10-500 tasks over 3 VMs. The result was compared with priority-based scheduling as well as PSO based priority scheduling. As compared to these approaches the proposed approach outperforms better. The average make span time for such condition was approx. 39ms. Similarly, fig 3 presents the average delay time (ADT) for varying tasks. The result was also presented for priority-based scheduling as well as PSO based priority scheduling. The average ADT for priority-based scheduling as well as PSO based priority scheduling was more than 150ms whereas the ADT for proposed approach was approx. 20ms. Fig 4 presents the DI representation for the proposed approach as compared to other two techniques. The average DI for both approaches was approx. 4 whereas for proposed it is near about 0.3. Then in fig 5 the SD comparison is presented. The average SD for the priority-based scheduling as well as PSO based priority scheduling was more than 80 whereas the average SD for the proposed approach was approx. 11. This shows quite improvement over the conventional approaches.

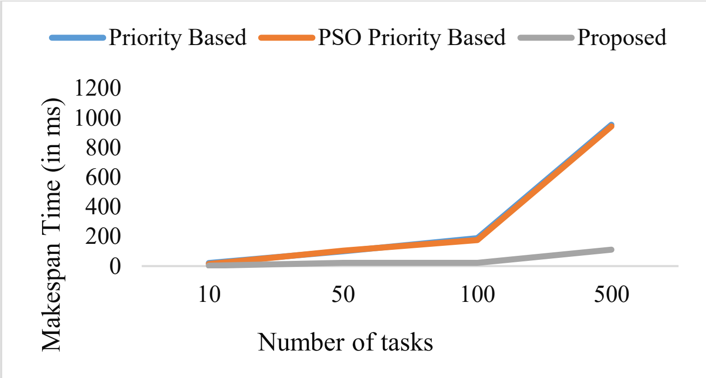

Fig. 2. Makespan Time Evaluation with respect to Workflow Tasks

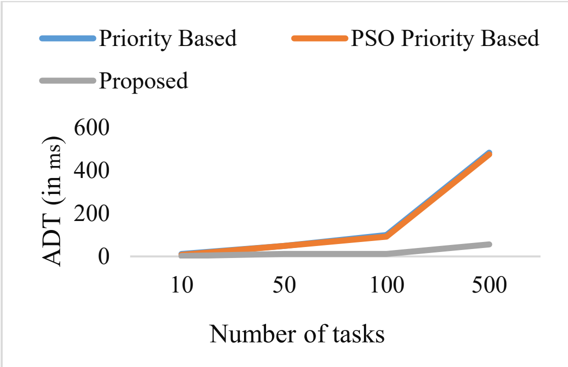

Fig. 3. ADT Evaluation with respect to Workflow Tasks

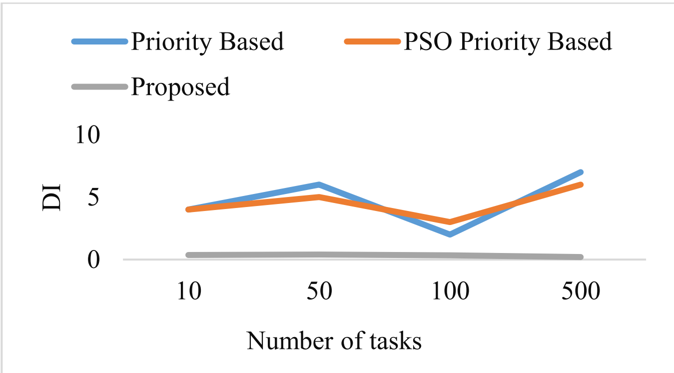

Fig. 4. DI Evaluation with respect to Workflow Tasks

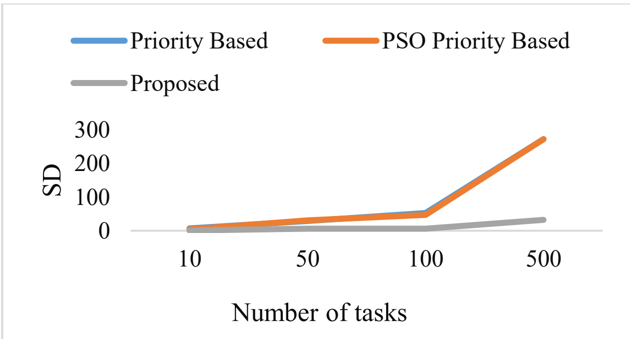

Fig. 5. SD Evaluation with respect to Workflow Tasks

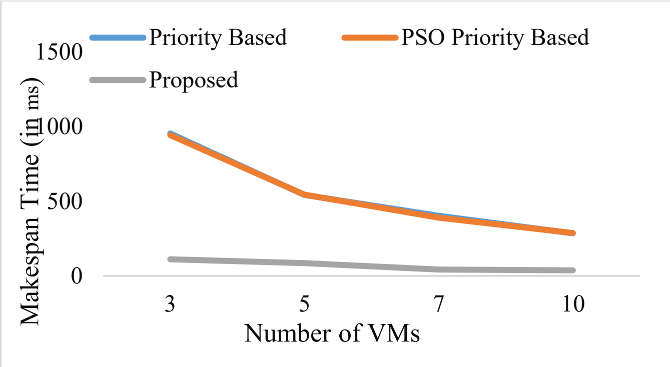

Fig. 6. Makespan Time Evaluation with respect to VMs

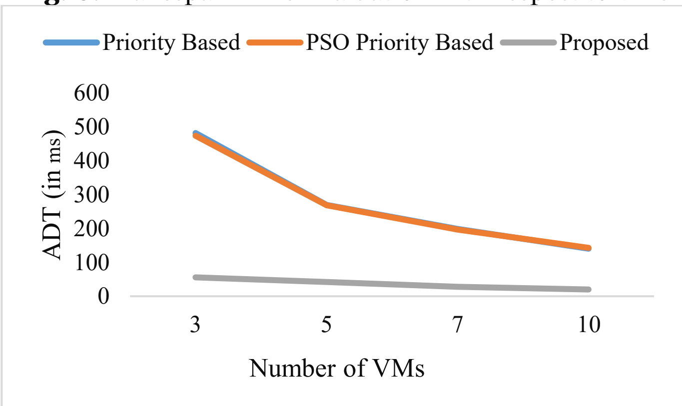

Fig. 7. ADT Evaluation with respect to VMs

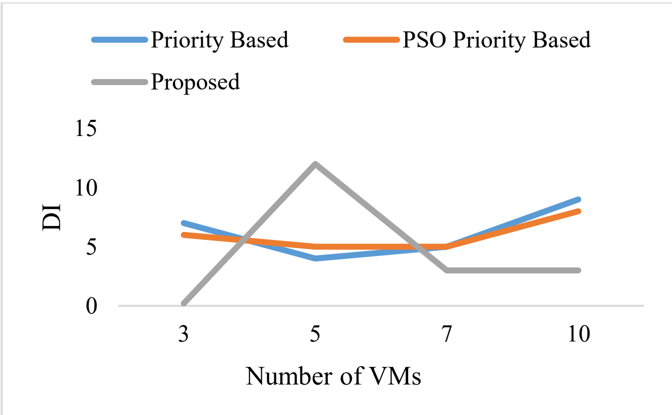

Fig. 8. DI Evaluation with respect to VMs

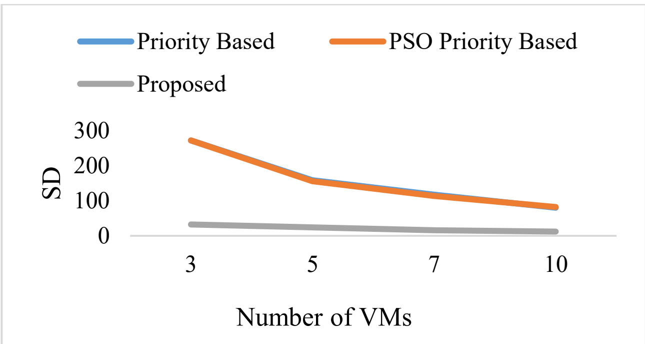

Fig. 9. SD Evaluation with respect to VMs

Fig 6 presents the make span time evaluation for the proposed algorithm with respect to variable virtual machines. The result was evaluated for 3-10 VMs for 500 tasks. The result was compared with priority based scheduling as well as PSO based priority scheduling. As compared to these approaches the proposed approach outperforms better. The average make span time for such condition was approx. 68ms. Similarly, fig 7 presents the average delay time (ADT) for varying tasks. The result was also presented for priority based scheduling as well as PSO based priority scheduling. The average ADT for priority based scheduling as well as

PSO based priority scheduling was more than 200ms whereas the ADT for proposed approach was approx. 35ms. Fig 8 presents the DI representation for the proposed approach as compared to other two techniques. The average DI for both approaches was approx. 6 whereas for proposed it is near about 4. Then in fig 9 the SD comparison is presented. The average SD for the priority based scheduling as well as PSO based priority scheduling was more than 150 whereas the average SD for the proposed approach was approx. 21. This shows quite improvement over the conventional approaches

Fig 10 presents the make span time comparison with existing work. The existing approach uses the cuckoo search optimization approach for workflow scheduling. The approach conducted the experimentation for 5 VMs and 25 tasks, 10 VMs and 25 tasks, 10 VMs 100 tasks and 15 VMs with 100 tasks and achieved make span time of 27.32ms, 29.36ms, 31.57ms and 32.46ms respectively. The proposed approach achieved the make span of 6ms, 3.6ms, 12ms and 10ms respectively. This result shows that the proposed approach have achieved better performance as compared to existing work.

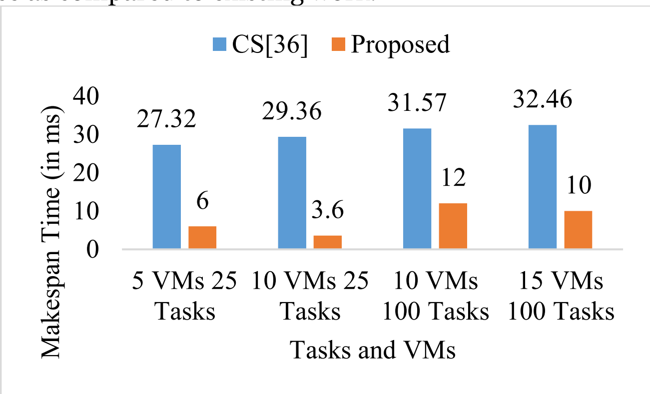

Fig 10. Makespan Comparative Analysis

7. CONCLUSION

In this paper workflow scheduling is presented for cloud. The paper explored that shifting from traditional approaches to data-driven approach such as ML/AI solutions has improved the efficiency, scalability, and cost-effectiveness of the scheduling process. The proposed approach is dedicated to design an approach for workflow task scheduling using a multi-stage framework. In the first stage of the model, task priotization is performed then second stage generates the resource matrix (RM) and finally in last stage optimal execution resource (OER) allocation is performed. The result analysis shows different operational conditions of varying task and VMs. The average makespan time for the proposed approach was in optimal range as compared to existing approach. In future the work would be extended in multi-workflows environment under hybrid cloud architecture.

REFERENCES

[1] Jiahao, W., Zhiping, P., Delong, C., Qirui, L., Jieguang, H. (2018). A Multi-object Optimization Cloud Workflow Scheduling Algorithm Based on Reinforcement Learning. In: Huang, DS., Jo, KH., Zhang, XL. (eds) Intelligent Computing Theories and Application. ICIC 2018. Lecture Notes in Computer Science(), vol 10955. Springer, Cham. https://doi.org/10.1007/978-3-319-95933-7_64

[2] Yogesh Kumar, Surabhi Kaul, Yu-Chen Hu, Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey, Sustainable Computing: Informatics and Systems, Volume 36, 2022, 100780, ISSN 2210-5379, https://doi.org/10.1016/j.suscom.2022.100780.

[3] Ji Hai Zhong, De Long Cui, Zhi Ping Peng, Qi Rui Li, Jie Guang He,Multi Workflow Fair Scheduling Scheme Research Based on Reinforcement Learning, Procedia Computer Science,Volume 154, 2019, Pages 117-123, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.06.018.

[4] Dong, T., Xue, F., Tang, H. et al. Deep reinforcement learning for fault-tolerant workflow scheduling in cloud environment. Appl Intell 53, 9916–9932 (2023). https://doi.org/10.1007/s10489-022-03963-w

[5] Cui, D., Ke, W., Peng, Z., Zuo, J. (2016). Multiple DAGs Workflow Scheduling Algorithm Based on Reinforcement Learning in Cloud Computing. In: Li, K., Li, J., Liu, Y., Castiglione, A. (eds) Computational Intelligence and Intelligent Systems. ISICA 2015. Communications in Computer and Information Science, vol 575. Springer, Singapore. https://doi.org/10.1007/978-981-10-0356-1_31

[6] Asghari, A., Sohrabi, M.K. & Yaghmaee, F. Task scheduling, resource provisioning, and load balancing on scientific workflows using parallel SARSA reinforcement learning agents and genetic algorithm. J Supercomput 77, 2800–2828 (2021). https://doi.org/10.1007/s11227-020-03364-1

[7] Srikanth, G.U., Geetha, R. Effectiveness Review of the Machine Learning Algorithms for Scheduling in Cloud Environment. Arch Computat Methods Eng 30, 3769–3789 (2023). https://doi.org/10.1007/s11831-023-09921-0

[8] Asghari, A., Sohrabi, M.K. & Yaghmaee, F. Online scheduling of dependent tasks of cloud's workflows to enhance resource utilization and reduce the makespan using multiple reinforcement learning-based agents. Soft Comput 24, 16177–16199 (2020). https://doi.org/10.1007/s00500-020-04931-7

[9] Cheng, F., Huang, Y., Tanpure, B. et al. Cost-aware job scheduling for cloud instances using deep reinforcement learning. Cluster Comput 25, 619–631 (2022). https://doi.org/10.1007/s10586-021-03436-8

[10] B. Kruekaew and W. Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning," in IEEE Access, vol. 10, pp. 17803-17818, 2022, doi: 10.1109/ACCESS.2022.3149955.

[11] F. A. Saif, R. Latip, Z. M. Hanapi and K. Shafinah, "Multi-Objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud-Fog Computing," in IEEE Access, vol. 11, pp. 20635-20646, 2023, doi: 10.1109/ACCESS.2023.3241240.

[12] S. Chen, J. Chen, Y. Miao, Q. Wang and C. Zhao, "Deep Reinforcement Learning-Based Cloud-Edge Collaborative Mobile Computation Offloading in Industrial Networks," in IEEE Transactions on Signal and Information Processing over Networks, vol. 8, pp. 364-375, 2022, doi: 10.1109/TSIPN.2022.3171336.

[13] H. Djigal, J. Xu, L. Liu and Y. Zhang, "Machine and Deep Learning for Resource Allocation in Multi-Access Edge Computing: A Survey," in IEEE Communications Surveys & Tutorials, vol. 24, no. 4, pp. 2449-2494, Fourthquarter 2022, doi: 10.1109/COMST.2022.3199544.

[14] B. Kar, W. Yahya, Y. -D. Lin and A. Ali, "Offloading Using Traditional Optimization and Machine Learning in Federated Cloud–Edge–Fog Systems: A Survey," in IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1199-1226, Secondquarter 2023, doi: 10.1109/COMST.2023.3239579.

[15] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan and M. Xiao, "Asynchronous Deep Reinforcement Learning for Collaborative Task Computing and On-Demand Resource Allocation in Vehicular Edge Computing," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2023.3249745.

[16] X. Wang, L. Ma, H. Li, Z. Yin, T. Luan and N. Cheng, "Digital Twin-Assisted Efficient Reinforcement Learning for Edge Task Scheduling," 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 2022, pp. 1-5, doi: 10.1109/VTC2022-Spring54318.2022.9860495.

[17] J. Huang, J. Wan, B. Lv, Q. Ye and Y. Chen, "Joint Computation Offloading and Resource Allocation for Edge-Cloud Collaboration in Internet of Vehicles via Deep Reinforcement Learning," in IEEE Systems Journal, vol. 17, no. 2, pp. 2500-2511, June 2023, doi: 10.1109/JSYST.2023.3249217.

[18] L. Zhang, C. Yang, Y. Yan and Y. Hu, "Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning," in IEEE Transactions on Industrial Informatics, vol. 18, no. 12, pp. 8999-9007, Dec. 2022, doi: 10.1109/TII.2022.3178410.

[19] Y. Xu, M. Z. A. Bhuiyan, T. Wang, X. Zhou and A. K. Singh, "C-FDRL: Context-Aware Privacy-Preserving Offloading Through Federated Deep Reinforcement Learning in Cloud-Enabled IoT," in IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 1155-1164, Feb. 2023, doi: 10.1109/TII.2022.3149335.

[20] T. M. Ho, K. -K. Nguyen and M. Cheriet, "Federated Deep Reinforcement Learning for Task Scheduling in Heterogeneous Autonomous Robotic System," in IEEE Transactions on Automation Science and Engineering, 2022, doi: 10.1109/TASE.2022.3221352.

[21] Genxin Chen, Jin Qi, Ying Sun, Xiaoxuan Hu, Zhenjiang Dong, Yanfei Sun,A collaborative scheduling method for cloud computing heterogeneous workflows based on deep reinforcement learning, Future Generation Computer Systems, Volume 141, 2023, Pages 284-297, ISSN 0167-739X, https://doi.org/10.1016/j.future.2022.11.032.

[22] Y. Cheng et al., "Accelerating End-to-End Deep Learning Workflow With Codesign of Data Preprocessing and Scheduling," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 7, pp. 1802-1814, 1 July 2021, doi: 10.1109/TPDS.2020.3047966.

[23] X. Ma, H. Xu, H. Gao, M. Bian and W. Hussain, "Real-Time Virtual Machine Scheduling in Industry IoT Network: A Reinforcement Learning Method," in IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 2129-2139, Feb. 2023, doi: 10.1109/TII.2022.3211622.

[24] Amanda Jayanetti, Saman Halgamuge, Rajkumar Buyya, Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge–cloud computing environments, Future Generation Computer Systems, Volume 137, 2022, Pages 14-30, ISSN 0167-739X, https://doi.org/10.1016/j.future.2022.06.012.

[25] Genxin Chen, Jin Qi, Ying Sun, Xiaoxuan Hu, Zhenjiang Dong, Yanfei Sun, A collaborative scheduling method for cloud computing heterogeneous workflows based on deep reinforcement learning, Future Generation Computer Systems, Volume 141, 2023,Pages 284-297, ISSN 0167-739X, https://doi.org/10.1016/j.future.2022.11.032.

[26] H. Mahmoud, M. Thabet, M. H. Khafagy and F. A. Omara, "Multiobjective Task Scheduling in Cloud Environment Using Decision Tree Algorithm," in IEEE Access, vol. 10, pp. 36140-36151, 2022, doi: 10.1109/ACCESS.2022.3163273.

[27] A. Pradhan, S. K. Bisoy, S. Kautish, M. B. Jasser and A. W. Mohamed, "Intelligent Decision-Making of Load Balancing Using Deep Reinforcement Learning and Parallel PSO in Cloud Environment," in IEEE Access, vol. 10, pp. 76939-76952, 2022, doi: 10.1109/ACCESS.2022.3192628.
[28] X. Ma, H. Xu, H. Gao, M. Bian and W. Hussain, "Real-Time Virtual Machine Scheduling in Industry IoT Network: A Reinforcement Learning Method," in IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 2129-2139, Feb. 2023, doi: 10.1109/TII.2022.3211622.
[29] X. Zhou et al., "Edge-Enabled Two-Stage Scheduling Based on Deep Reinforcement Learning for Internet of Everything," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3295-3304, 15 Feb.15, 2023, doi: 10.1109/JIOT.2022.3179231.
[30] Mohammadzadeh, Ali, and Mohammad Masdari. "Scientific workflow scheduling in multi-cloud computing using a hybrid multi-objective optimization algorithm." Journal of Ambient Intelligence and Humanized Computing 14.4 (2023): 3509-3529.
[31] Dong, Tingting, et al. "A hybrid algorithm for workflow scheduling in cloud environment." International Journal of Bio-Inspired Computation 21.1 (2023): 48-56.
[32] Lu, Chang, et al. "A multi-hierarchy particle swarm optimization-based algorithm for cloud workflow scheduling." Future Generation Computer Systems 153 (2024): 125-138.
[33] Zeedan, Maha, Gamal Attiya, and Nawal El-Fishawy. "Enhanced hybrid multi-objective workflow scheduling approach based artificial bee colony in cloud computing." Computing 105.1 (2023): 217-247.
[34] Reddy, Pillareddy Vamsheedhar, and Karri Ganesh Reddy. "An energy efficient RL based workflow scheduling in cloud computing." Expert Systems with Applications 234 (2023): 121038.
[35] Shukla, Prashant, and Sudhakar Pandey. "DE-GWO: a multi-objective workflow scheduling algorithm for heterogeneous fog-cloud environment." Arabian Journal for Science and Engineering 49.3 (2024): 4419-4444.
[36] Sahu, Babuli, et al. "Multiobjective Prioritized Workflow Scheduling in Cloud Computing Using Cuckoo Search Algorithm." Applied Bionics and Biomechanics 2023 (2023).