International Journal of Communication Networks and Information Security
ISSN: 2073-607X, 2076-0930
Volume 15 Issue 01 Year 2023

The Role of Information Systems in Measuring the Cost and Performance-Aware Scheduling Technique for the Cloud Computing Environment

M. A. Ashabrawy
Department of Computer Science, College of Computer Engineering and Science in AI-Kharj, Prince Sattam bin AbdulAziz University, AI-Kharj, Saudi Arabia
Reactors Department, Nuclear Research Center, Atomic Energy Authority, Egypt
ashabrawy@hotmail.com

Article History

Received: 7 April 2023
Revised: 19 May 2023
Accepted: 5 June 2023

Abstract

A lot of interest has been put forth to improve workload scheduling in the cloud platform. However, the execution of scientific workflow on a cloud platform is time-consuming and expensive. Much research has been emphasised, as users are charged based on the usage hour, minimising processing time to reduce cost. However, the processing cost can be reduced by minimising energy consumption, especially when resources are heterogeneous; Minimal work has been done considering optimising cost with energy and processing time parameters to meet task Quality of Service (QoS) requirements. This paper presents cost and performance-aware workload scheduling (CPA-WS) techniques under a heterogeneous cloud platform. This paper presents a cost optimisation model through the minimisation of processing time and energy dissipation for the execution of the task. Experiments are conducted using two widely used workflows such as Inspiral and CyberShake. The outcome shows the CPA-WS significantly reduces energy, time, and cost compared to the standard workload scheduling model.

Keywords: Information System (IS), Cloud Computing, Cost-Performance Optimisation, Workflows, Heterogeneous Server, Scheduling

1. Introduction

Cloud computing platforms are widely used for provisioning high-performance computing as a web-services for the execution of workflows [1]. Recently, a wide range of scientific areas such as bioinformatics, physics, and astronomy have leveraged cloud environments for modelling scientific workflows representing real-world problems [2]; thus, large scientific workflows can be analysed through simulation more effectively [3] with minimal time and cost [4], [5]. The scientific workflow is represented directed acyclic graph (DAG) where edges represent a set of tasks, and vertices represent its dependencies. Thus, the forthcoming task will be initiated once the primary task is completed [6]-[9]. These dependencies among tasks make scheduling in the cloud very challenging.

Recently, workflow scheduling in cloud computing platforms has gained wide attention across the research community [10]; a basic architecture of workload scheduling using the cloud is shown in Figure 1. However, designing efficient scheduling design adopting currently available heuristic models pose several difficulties such as sizeable scientific workflow prerequisite higher execution time, and execution cost. Further, it becomes even more complicated when a task demands a deadline prerequisite. Extensive work has been done to establish optimal solutions through heuristic algorithms. However, the heuristic strategy depends on job order without considering the job scheduling duration. As a result, fails to obtain optimal solutions, affecting the overall Quality of services and higher SLA violations. Thus, workflow scheduling is considered an NP-hard (non-polynomial) problem [11], [12]. Optimising cost and time together becomes extremely difficult [13]. For example, if the scheduling design tries to minimise cost, it increases the execution time because a relationship exists between them. Many existing models need to consider virtual machine selection policy in scheduling design. Thus cost-makespan optimisation problem still exists [14], [15].

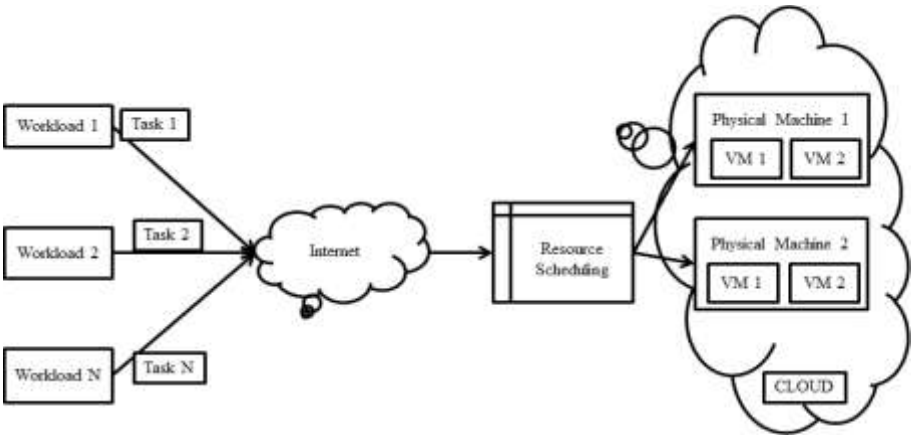

Figure 1. The Basic Architecture of Workload Scheduling Using the Cloud

In addressing research problems, this paper presents cost and performance-aware workload scheduling (CPA-WS) techniques for heterogeneous cloud computing (HCC) environments. The model optimises workload execution cost through energy and processing time minimisation constraints; Further, the CPA-WS presents an effective queuing model for ideal load balancing between already scheduled tasks concerning newly arriving tasks.

The manuscript's significance is described below:

This paper presents an effective workload scheduling technique that reduces cost.

Cost optimisation is done through the minimisation of energy and processing time constraints under heterogeneous computing platforms.

CPA-WS provides an effective load-balancing mechanism, thus, reducing buffer overhead and task waiting time.

CPA-WS achieves much better cost, energy, and processing time efficiency than EMS.

The manuscript is arranged as follows. Section 2 studies various existing workload scheduling models' advantages and limitations. Section 3 provides the mathematical representation of the proposed CPA-WS model is given. The result and discussion are given in section 4, and in chapter 5, the research is concluded with a future research direction.

2. Literature Survey

In the research, the survey is conducted to understand the benefits and limitations of using standard workload scheduling. In [16], it focused on designing and optimising energy and cost together to design workflow scheduling for heterogeneous computing platforms. Here a Min function is modelled for reducing the energy cost and meeting task deadlines considering task information is geographically distributed. Here they divided the task considering different deadlines and sorted them according to deadlines, small to high. Finally, an adaptive searching method is designed for effective optioning schedules for workflow execution. In [17] showed how energy consumption significantly increases the computing cost of service provisioning. Reliability and timeliness are a few key metrics in service provisioning. They designed a scheduling design that reduces energy dissipation and meets the reliability and timeliness requirements of workflow executions. Here a heuristic solution is obtained through a Non-linear Mixed Integer Programming problem. First, a scheduling length minimisation strategy is modelled for meeting reliability. Second, we designed a processor merging strategy to reduce energy dissipation by leveraging Dynamic Voltage Frequency Scaling (DVFS) technique. Here inefficient machines are switched off, scaling is done at both task and processor levels.

In [18], the modelled tradeoff to handle the unpredictable resource availability nature of cloud computing by adopting an evolutionary computing algorithm. Here a multi-objective parameter optimisation model of cost and makespan is considered together. Performance is studied considering various levels of interruption, and the outcome shows better performance than existing models [19]. In [20], we modelled an evolutionary computing model, namely Nested Particle Swarm Optimisation (NPSO), and a faster version of NPSO, namely FNPSO optimising execution of composite workflows. The FNPSO significantly reduces in comparison with the NPSO model. In [21], combined Q-Learning (QL) and Heterogeneous Earliest Finish Time (HEFT) to design an effective scheduling technique, namely QL-HEFT. The QL-HEFT is intended to reduce computation time. The reward function in QL is updated using the upward rank outcome of HEFT. This aid in improving the learning efficiency of the Q-Learning algorithm. The QL first obtain an optimal order of task and then finds the suitable machine for the execution of the task utilising the earliest finishing time. [22] designed a scheduling design considering contention awareness for workflow execution. A list scheduling heuristic with endpoint contention awareness is modelled to minimise makespan. A ranking mechanism is introduced to schedule a task to computational machines and modelled a rescheduling design to improve scheduling efficiency.

In [23], designed a workflow scheduling design adopted an evolutionary computing model to meet task deadlines by optimising cost, namely DCOH. Further, improved DCOH by incorporating multi-objective parameters by optimising makespan and cost together under a hybrid cloud platform. In [24], workflow application scheduling is designed to meet the application deadline and cost together. Here they improved the priority selection design for establishing the order of tasks. During the allocation of computational resources, budget and cost ratios are used to correlate between budget and deadline constraints. In improving success rate (i.e. reliability), certain decisions are discarded through the discarding mechanism.

In [25], showed scheduling model in the cloud must meet user deadline prerequisites and SLAs. They adopted a multi-cloud platform to meet stream workflow application performance requirements and reduce cost. In [26], design a fault-tolerant scheduling design for workflow execution leveraging a multi-cloud platform. Further, the model assures meeting reliability requirements and with reduced cost. Here they employed continuous probability distribution for analysing failure rate and reliability. Then, a mathematical model to measure the cost of executing using a multi-cloud platform is given, followed by defining fault-tolerant workflow scheduling design by assuring reliability and reducing cost and execution time. However, it could not guarantee meeting the cost constraints of application requirements because of the poor load-balancing mechanism. In addressing the issues above, in next section presents cost and performance-aware scheduling techniques under heterogeneous cloud environments.

3. Cost and Performance Aware Scheduling Technique for Cloud Computing Environment

This section presents cost and performance-aware workload scheduling (CPA-WS) techniques for executing scientific workflow in an HCC environment. The workload scheduling architecture of CPA-WS is shown in Figure 2. The CPA-WS technique is modelled to schedule tasks with minimal cost by optimising energy consumption and meeting task deadlines and performance prerequisites without causing a congested HCC environment. Here an effective task queueing methodology is modelled for load balancing. The task queuing methodology comprises o HCC server $\mathrm{T}_1, \mathrm{T}_2, \dots, \mathrm{T}_o$ with capacity $\mathrm{n}_1, \mathrm{n}_2, \dots,$ no, and its computational capability is $\mathrm{t}_1, \mathrm{t}_2, \dots, \mathrm{t}_o$. Let the HCC server T_j comprise n_j identical servers with computational capability t_j. The arrival load α is exponentially distributed with randomness (s) and mean average $(\bar{s})$ $1/\alpha$ considering the Poisson process with $M/M/m$ queuing model. The CPA-WS technique segment the task set into o sub-set where the j^{th} sub-set with arrival load α_j is communicated to HCC server T_j, where $1 \le j \le o$, $\alpha = \alpha_1 + \alpha_2 + \dots + \alpha_o$. An HCC server T_j retains a queue with boundless capacity for tasks in the queue, waiting to be executed when the whole server n_j is busy. The scheduling is done according to first come, first serve with exponential randomness s and mean $\bar{s}$. The servers of HCC server T_j have similar computation capacity t_j. Therefore, the computation time with exponential randomness is measured using the following equation

$$y_j = \frac{s}{t_j} \tag{1}$$

$$\bar{y}_j = \frac{\bar{s}}{t_j} \tag{2}$$

with mean

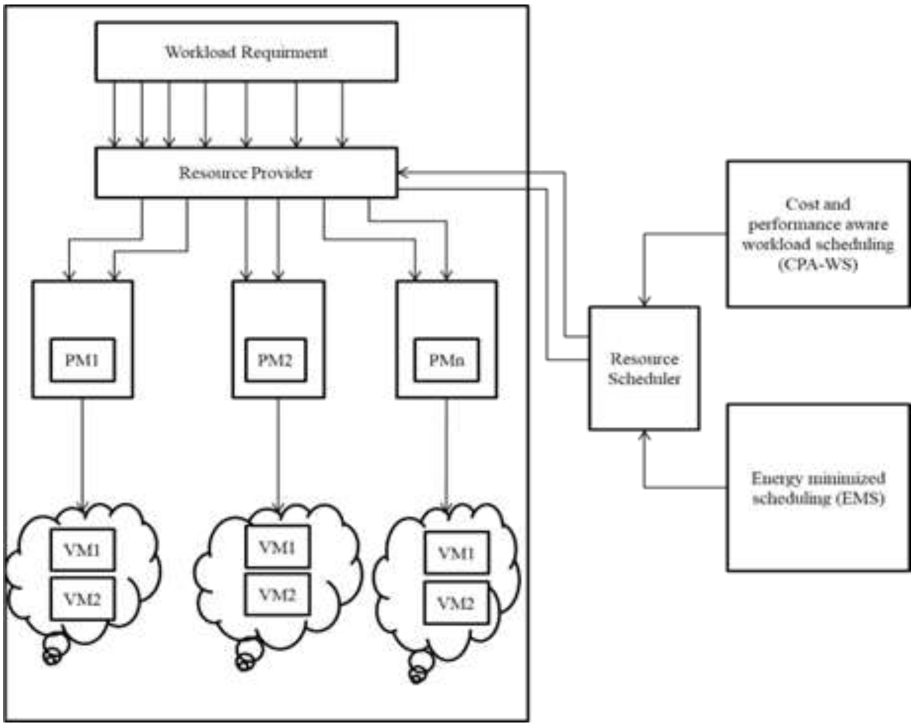

Figure 2. Workload Scheduling Architecture of CPA-WS

The mean task (i.e., the mean success rate), which is possible to be finished by the HCC server within \mathbit{T}_\mathbit{j}, is measured as follows.

$$\beta_j = \frac{1}{\bar{y}_j} \tag{3}$$

In the meantime, the server will be busy, i.e., the resource utilisation is measured as follows.

$$\gamma_j = \frac{\alpha_j}{n_j \beta_j} = \frac{\alpha_j \bar{y}_j}{n_j} = \frac{\alpha_j \bar{s}}{n_j t_j} \tag{4}$$

Let $p_{j,l}$ defines the probability that l task resides in a queue or can be handled in HCC server T_j is measured as follows.

$$p_{j,l} = \begin{cases} p_{j,0} \dfrac{(n_j \gamma_j)^l}{l!}, & l < n_j; \\ p_{j,0} \dfrac{n_j^{n_j} \gamma_j^l}{l!}, & l \geq n_j; \end{cases} \tag{5}$$

where

$$p_{j,0} = \left(\sum_{l=0}^{n_j - 1} \frac{(n_j \gamma_j)^l}{l!} + \frac{(n_j \gamma_j)^{n_j}}{n_j!} \cdot \frac{1}{1 - \gamma_j} \right)^{-1} \tag{6}$$

The probability of a newly arriving workflow task that will reside in HCC server T_j when the whole server in T_j is busy is measured as follows

$$P_{r,j} = \frac{q_j, n_j}{1 - \gamma_j} = p_{j,0} \frac{n_j^{n_j}}{n_j!} \cdot \frac{\gamma_j^{n_j}}{1 - \gamma_j} \tag{7}$$

The average workflow task currently executed/waiting in HCC server T_j is measured as follows.

$$\bar{O}_j = \sum_{l=0}^{\infty} l p_{j,l} = n_j \gamma_j + \frac{\gamma_j}{1 - \gamma_j} P_{r,j} \tag{8}$$

Similarly to Equation (8), the average workflow task completion time of HCC server T_j is measured as follows

$$U_j = \frac{\bar{O}_j}{\alpha_j} = \bar{y}_j + \frac{P_{r,j}}{n_j(1 - \gamma_j)} \bar{y}_j = \bar{y}_j \left(1 + \frac{P_{r,j}}{n_j(1 - \gamma_j)} \right) \tag{9}$$

The mean workflow task computation time of HCC server T_j is measured for ease.

$$U_j = \frac{\bar{s}}{t_j} \left(1 + p_{j,0} \frac{n_j^{n_j - 1}}{n_j!} \cdot \frac{\gamma_j^{n_j}}{(1 - \gamma_j)^2} \right) \tag{10}$$

The energy needed for completing task execution is measured as follows

$$Q = a\mathcal{C}\mathcal{V}^2\mathcal{F} = \delta t^{\mu} \tag{11}$$

A represents the task characteristics, $and\ \mathcal{V}$, $\mathcal{C}$, $\mathcal{F}$, and t, depict voltage, load capacitance, clock frequency, and processor speed, respectively. In Equation (11), the δ is measured as follows

$$\delta = \frac{a\mathscr{b}^2\mathcal{C}}{c^{2\rho + 1}} \tag{12}$$

In above Equation (12), the parameter $\mathscr{b}$ and ρ defines a constant higher than zero. The μ is measured as follows

$$\mu = 2\rho + 1 \tag{13}$$

The existing method considers both δ and μ across servers; However, in this work, it is not the case because of the HCC environment adopted; thus, I have a different value of δ and μ. Here I consider two different energy types such as static energy and dynamic energy type. The computational machine will not perform any task in the static energy type, and the energy consumed is measured as follows.

$$Q_j = n_j \left(\gamma_j \delta_j t_j^{\mu_j} + Q_j^* \right) = \alpha_j \bar{t} \delta_j t_j^{\mu_j - 1} + n_j Q_j^* \tag{14}$$

Similarly, in the dynamic energy type, the computational machine will execute the task/wait for the task's arrival, and the energy consumed is measured as follows.

$$Q_j = n_j \left(\delta_j t_j^{\mu_j} + Q_j^* \right) \tag{15}$$

This work aimed at allocating ideal resources with minimal execution cost by optimising energy and processing time for executing workload tasks under an HCC environment with varying processing speed and power consumption.

Let's consider an o HCC server with the size of $n_1, n_2, \ldots, n_o$, with dynamic energy dissipation and computation capacity for execution of workflow with prerequisite $\bar{s}$ with task arrival rate α, and have load distribution $\alpha_1, \alpha_2, \ldots, \alpha_o$ in achieving high-performance efficiency is obtained through following minimisation function

$$\min U(\alpha_1, \alpha_2, \ldots, \alpha_o) \tag{16}$$

Equation (16) is subjected to the constraint described below

$$G(\alpha_1, \alpha_2, \ldots, \alpha_o) = \alpha \tag{17}$$

where

$$G(\alpha_1, \alpha_2, \ldots, \alpha_o) = \alpha_1 + \alpha_2 + \cdots + \alpha_o \quad (18)$$

and $\gamma_j < 1, \forall\, 1 \leq j \leq o$.

Let's consider an o HCC server with the size of $n_1, n_2, \ldots, n_o$, with dynamic energy dissipation and computation capacity for execution of workflow with prerequisite $\bar{s}$ with task arrival rate α, and have load distribution $\alpha_1, \alpha_2, \ldots, \alpha_o$ in reducing energy consumption is obtained through following minimisation function

$$\min Q(\alpha_1, \alpha_2, \ldots, \alpha_o) \quad (19)$$

Equation (19) is subjected to the constraint described below

$$G(\alpha_1, \alpha_2, \ldots, \alpha_o) = \alpha \quad (20)$$

where

$$G(\alpha_1, \alpha_2, \ldots, \alpha_o) = \alpha_1 + \alpha_2 + \cdots + \alpha_o \quad (21)$$

and $\gamma_j < 1, \forall\, 1 \leq j \leq o$.

Let's consider a heterogeneous computing platform Tj; the cost outcome can be measured through the inverse proportion of execution time using the following equation.

$$C = \frac{1}{U_j} \quad (22)$$

However, the proposed design considers the energy factor Q_j into for measuring cost as defined below.

$$S_j = Q_j U_j \quad (23)$$

The mean cost-performance S considering o heterogeneous computing platform $T_1, T_2, \ldots, T_o$ is measured through the following equation

$$S(\alpha_1, \alpha_2, \ldots, \alpha_o) = \frac{\alpha_1}{\alpha} S_1 + \frac{\alpha_2}{\alpha} S_2 + \cdots + \frac{\alpha_o}{\alpha} S_o \quad (24)$$

For simplicity, the above equation is rewritten as follows

$$= \frac{\alpha_1}{\alpha} Q_1 U_1 + \frac{\alpha_2}{\alpha} Q_2 U_2 + \cdots + \frac{\alpha_o}{\alpha} Q_o U_o \quad (25)$$

Here the workload tasks are scheduled by minimising Equation (16) and Equation (19) and meeting constraints defined in Equation (17), (18), (20), and (21) to bring tradeoffs between performance and cost.

4. Simulation Results

The experiment evaluates cost and performance-aware workload scheduling CPA-WS and energy-minimised scheduling (EMS) [17]. CloudSim3 [27] is used in modelling workload scheduling algorithms [28]. The complex workload Inspiral and CyberShake is used [29], [30] because it is widely used in validating various scheduling models [31], [32], where Inspiral requires more CPU and memory; however, the CyberShake requires CPU and I/O resources [29], [30]. Time efficiency, energy consumption, and cost efficiency are metrics used to measure the performance of CPA-WS and EMS.

4.1 Time Efficiency vs Workload Size

Here the time efficiency of CPA-WS and EMS is measured by varying the Inspiral and CyberShake workload task size from 30 to 1000. Time efficiency is measured as the time taken to complete the task; lesser time indicates better performance. Figure 3 shows the time taken to complete the tasks using CPA-WS and EMS for varied Inspiral workload sizes. Similarly, Figure 4 shows the time taken to complete the tasks using CPA-WS and EMS for varied CyberShake workload sizes. Experiments show that the CPA-WS is very efficient for smaller and larger workloads; however, EMS achieves inferior results for larger workloads considering both Inspiral and CyberShake workloads.

The CPA-WS improves time efficiency by 83.32% over EMS for Inspiral Workload. Similarly, The CPA-WS improves time efficiency by 79.16% over EMS for CyberShake Workload.

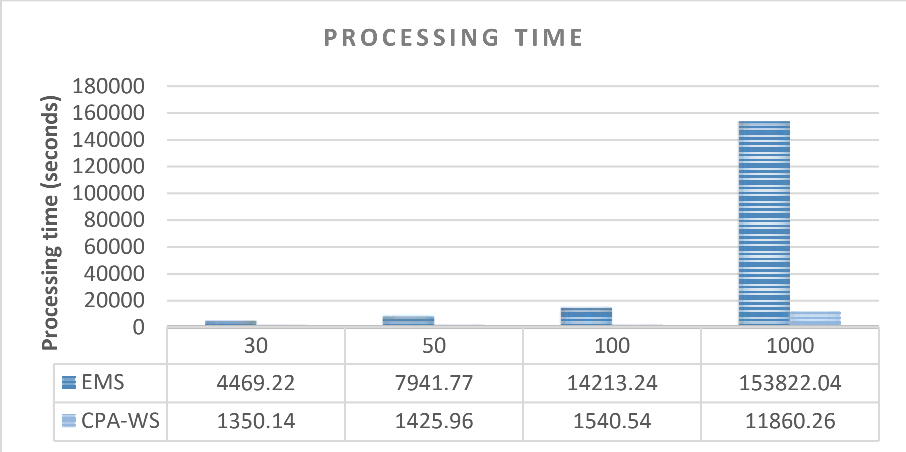

Figure 3. Time Efficiency with Different Inspiral Workload Sizes

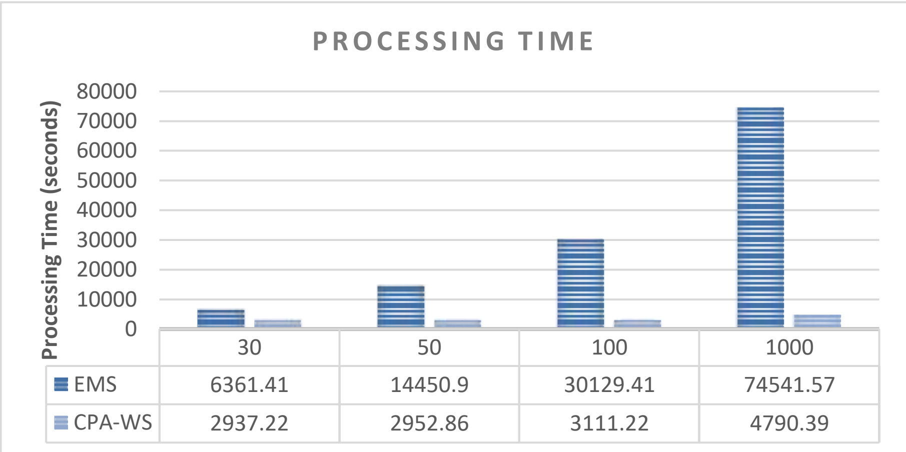

Figure 4. Time Efficiency with Different CyberShake Workload Sizes

4.2 Energy Consumption vs Workload Size

Here the energy consumption of CPA-WS is measured, and EMS is measured by varying the Inspiral and CyberShake workload task size from 30 to 1000. The energy consumption is measured as the amount of power consumed in a watt to complete the task; a lesser watt indicates better performance. Figure 5 shows the energy consumed to complete the tasks using CPA-WS and EMS for varied Inspiral workload sizes. Similarly, Figure 6 shows the energy consumed to complete the tasks using CPA-WS and EMS for varied CyberShake workload sizes. Experiments show that the CPA-WS is very energy efficient for smaller and larger workloads; however, EMS achieves significantly higher energy for smaller and larger workloads, considering both Inspiral and CyberShake workloads. The CPA-WS improves energy efficiency by 44.85% over EMS for Inspiral Workload. Similarly, The CPA-WS improves energy efficiency by 24.35% over EMS for CyberShake Workload.

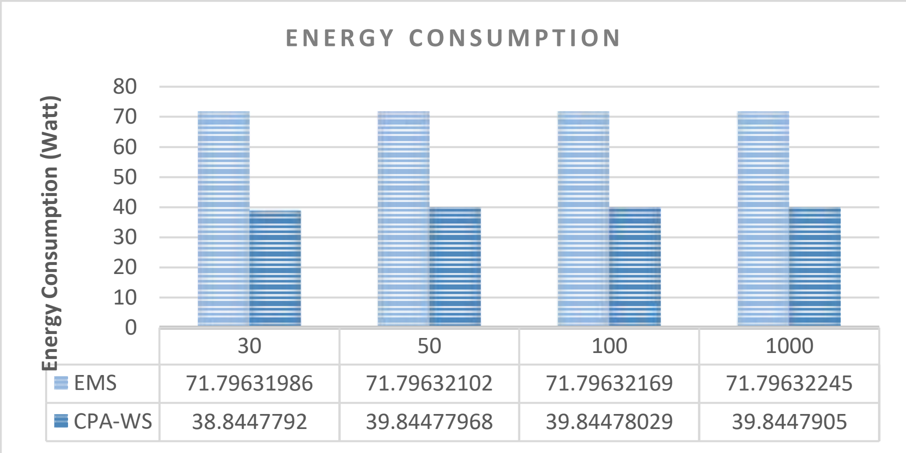

Figure 5. Energy Efficiency with Different Inspiral Workload Sizes

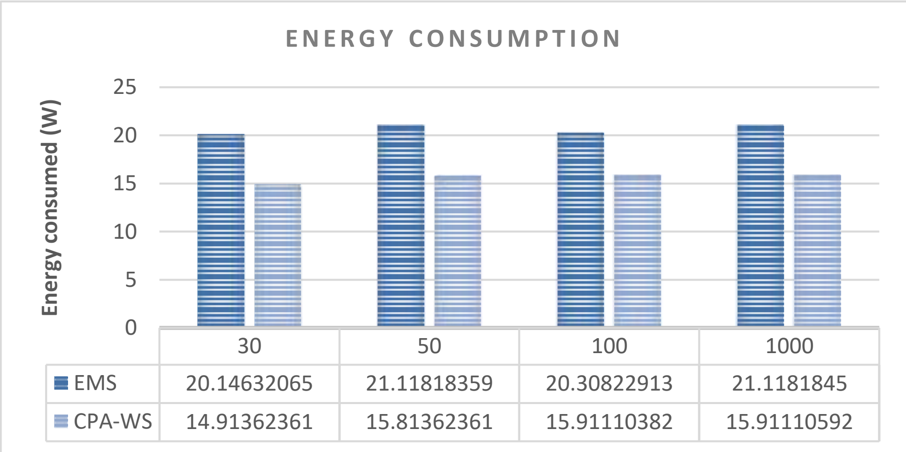

Figure 6. Energy Efficiency with Different CyberShake Workload Sizes

4.3 Cost Efficiency vs Workload Size

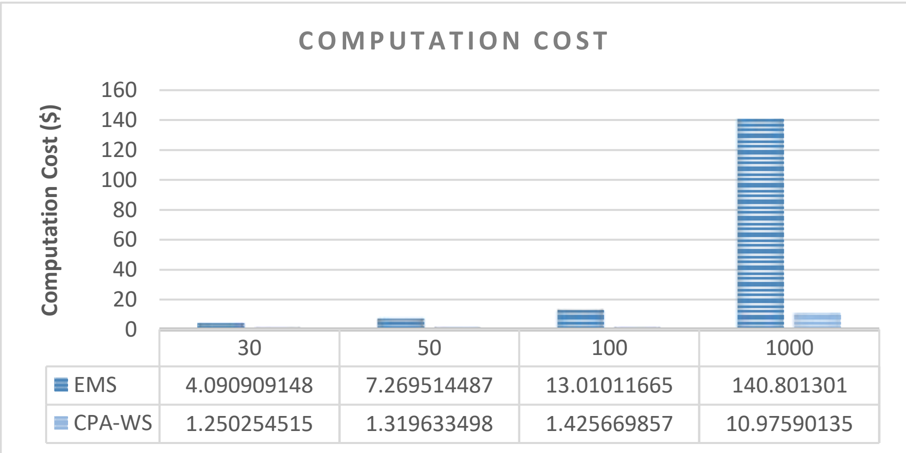

Figure 7. Cost Efficiency with Different Inspiral Workload Sizes

Here the cost efficiency of CPA-WS and EMS is measured by varying the Inspiral and CyberShake workload task size from 30 to 1000. Cost efficiency is measured as energy consumed and time taken to complete the task; a lesser value indicates better performance. Figure 7 shows the cost incurred to complete the tasks using CPA-WS and EMS for varied Inspiral workload sizes. Similarly, Figure 8 shows the cost incurred to complete tasks using CPA-WS and EMS for varied CyberShake workload sizes. Experiments show that the CPA-WS is very efficient for smaller and larger workloads; however, EMS achieves inferior results for larger workloads considering both Inspiral and CyberShake workloads. The CPA-WS reduces computation cost by 83.13% over EMS for Inspiral

Workload. Similarly, The CPA-WS reduces computation cost by 78.851% over EMS for CyberShake Workload.

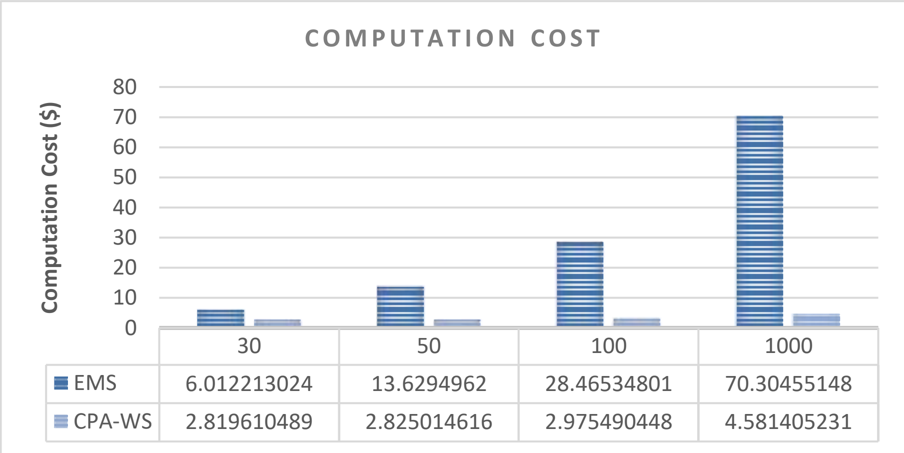

Figure 8. Cost Efficiency with Different CyberShake Workload Sizes

5. Conclusions

After studting different workload scheduling techniques for the execution of real-time workloads employing cloud computing platforms. The study identified that most existing workload scheduling focused on reducing cost by minimising processing time, energy, and delay; however, very few have focused on addressing cost minimisation considering both energy and processing time together under a heterogeneous cloud platform. This paper designed a workload scheduling technique by presenting energy and processing time optimisation constraint for reducing computation costs. Further, an effective load-balancing technique is presented for reducing the waiting time; adopting such a strategy significantly aid in utilising resource more efficiently. Experiment outcome shows the CPA-WS significantly improves time, energy, and cost efficiency by 83.32%, 44.85%, and 83.13% over EMS for executing Inspiral workload, respectively.

Similarly, CPA-WS significantly improves time, energy, and cost efficiency by 79.16%, 24.35%, and 78.851% over EMS for executing CyberShake workload. From the result, it can be stated that CPA-WS computation cost performance gets profitable with increasing workload size compared to EMS. Thus, they are suitable for provisioning smaller and larger workloads with high profitability. Future work would consider improving resource usage efficiency and provisioning security for workload execution for performing different kinds of tasks.

References

[1] A. A. Nasr, N. A. El-Bahnasawy, G. Attiya, A. El-Sayed, "Cost-effective algorithm for workflow scheduling in cloud computing under deadline constraint," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, 2019, pp.3765-3780.

[2] J. Sahni, D. P. Vidyarthi, "A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment," *IEEE Transactions on Cloud Computing*, vol. 6, no.1, 2015, pp.2-18.

[3] L. InfoTech, "What is cloud computing," *IBM Journal of Research and Development*, vol. 60, no. 4, 2012, pp.41-44.

[4] M. A. Sossa, "Resource provisioning and scheduling algorithms for scientific workflows in cloud computing environments," Doctoral dissertation, University of Melbourne, Department of Computing and Information Systems, 2016.

[5] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," *IEEE Transactions on Services Computing*, vol. 11, no.4, 2015, pp.713-726.

[6] M. Masdari, S. ValiKardan, Z. Shahi, S. I. Azar, "Towards workflow scheduling in cloud computing: a comprehensive analysis," *Journal of Network and Computer Applications*, vol. 66, 2016, pp.64-82.

[7] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei, S. Hu, "Minimising cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based heft," *Future Generation Computer Systems*, vol. 93, 2019, pp.278‑289.

[8] M. A. Rodriguez, R. Buyya, "Budget-driven scheduling of scientific workflows in IaaS clouds with fine-grained billing periods," *ACM Transactions on Autonomous and Adaptive Systems*, vol.12, no.2, pp.1‑22, 2017.

[9] N. Anwar, H. Deng, "Elastic scheduling of scientific workflows under deadline constraints in cloud computing environments," *Future Internet*, vol. 10, no. 1, 2018, pp. 5.

[10]Y. Zhao, X.Fei, I.Raicu, S.Lu, "Opportunities and challenges in running scientific workflows on the cloud," In International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. *IEEE*, 2011. pp 455‑462.

[11]G. Ismayilov, H.R.Topcuoglu, "Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing," *Future Generation Computer Systems*, vol. 102, 2020, pp. 307‑322.

[12]A. M. Manasrah, H. Ba Ali, "Workflow scheduling using hybrid ga-pso algorithm in cloud computing," *Wireless Communications and Mobile Computing*, pp. 1-16, 2018.

[13]S. Yassir, Z.Mostapha, T.Claude, "Workflow scheduling issues and techniques in cloud computing: A systematic literature review," In *International Conference of Cloud Computing Technologies and Applications*, Springer. pp 241‑263, 2017.

[14]Y. Qin, H. Wang, S. Yi, X. Li, L. Zhai, "An energy-aware scheduling algorithm for budget-constrained scientific workflows based on multi-objective reinforcement learning," *Journal of Supercomputing*, vol. 76, no.1, 2020, pp.455‑480.

[15]J. K. Konjaang, L. Xu, "Cost optimised heuristic algorithm (coha) for scientific workflow scheduling in iaas cloud environment," In 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), *IEEE Computer Society*, pp.162‑168.

[16]X. Li, W. Yu, R. Ruiz and J. Zhu, "Energy-aware cloud workflow applications scheduling with geo-distributed data," In *IEEE Transactions on Services Computing*, vol. 15, no. 2, 2020, pp. 891-903.

[17]B. Hu, Z. Cao and M. Zhou, "Energy-Minimised Scheduling of Real-Time Parallel Workflows on Heterogeneous Distributed Computing Systems," *IEEE Transactions on Services Computing*, vol.15, no.15, 2020, pp. 2766-2779.

[18]T. Pham and T. Fahringer, "Evolutionary Multi-objective Workflow Scheduling for Volatile Resources in the Cloud," *IEEE Transactions on Cloud Computing*, vol.10, no.3, 2020., pp. 1780-1791

[19]H. Li, B. Wang, Y. Yuan, M. Zhou, Y. Fan and Y. Xia, "Scoring and Dynamic Hierarchy-Based NSGA-II for Multi-objective Workflow Scheduling in the Cloud," *IEEE Transactions on Automation Science and Engineering*, vol.19, no.2, 2021, pp. 982-993.

[20]A. Song, W. Chen, X. Luo, Z. Zhan and J. Zhang, "Scheduling Workflows with Composite Tasks: A Nested Particle Swarm Optimisation Approach," *IEEE Transactions on Services Computing*, vol. 15, no. 2, 2020, pp.1074‑1088.

[21][21]Z.Tong, X. Deng, J.Mei, H. Liu, "QL-HEFT: a novel machine learning scheduling scheme base on cloud computing environment," *Neural Computing and Applications*, vol.32, 2020, pp. 5553-5570.

[22]Q. Wu, M. Zhou and J. Wen, "Endpoint Communication Contention-Aware Cloud Workflow Scheduling," *IEEE Transactions on Automation Science and Engineering*, vol.19, no.2, 2021, pp. 1137-1150.

[23]J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, M. Chen, "Cost and Makespan-Aware Workflow Scheduling in Hybrid Clouds," *Journal of Systems Architecture*, vol.100, 2019, pp.101631.

[24]G. Wang, Y. Wang, M. S. Obaidat, C. Lin and H. Guo, "Dynamic Multiworkflow Deadline and Budget Constrained Scheduling in Heterogeneous Distributed Systems," *IEEE Systems Journal*, vol.15, no.4, 2021, pp.4939‑49.

[25]M. Barika, S. Garg, A. Chan and R. Calheiros, “Scheduling Algorithms for Efficient Execution of Stream Workflow Applications in Multicloud Environments,” *IEEE Transactions on Services Computing*, vol.15, no.4, 2019, pp. 860‑875.

[26]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, 2021.

[27]R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, R. Buyya, “Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software: Practice and Experience*, vol. 41, no. 1, pp. 23‑50, 2011.

[28]W. Chen, E. Deelman, “Workflowsim: A toolkit for simulating scientific workflows in distributed environments,” In 2012 IEEE 8th International Conference on E-Science. *IEEE*. pp 1‑8, 2012.

[29]S. Bharathi et al., “Characterisation of scientific workflows,” In Workflows in Support of Large-Scale Science, 2008 third workshop on workflows in support of large-scale science, *IEEE*, pp. 1‑10, 2008.

[30]G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, “Characterising and profiling scientific workflows,” *Future Generation Computer Systems*, vol. 29, no. 3, pp. 682‑692, 2013.

[31]A. Choudhary, M. C. Govil, G. Singh, L. K. Awasthi and E. S. Pilli, “Task Clustering-Based Energy-Aware Workflow Scheduling in Cloud Environment,” 2018 IEEE 20th International Conference on High-Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018, pp. 968 ‑ 973.

[32]Y. Xie, Y. Zhu, Y. Wang, Y. Cheng, R. Xu, A. S. Sani, D. Yuan, Y. Yang, “A novel directional and non-local-convergent particle swarm optimisation based workflow scheduling in cloud‑edge environment,” *Future Generation Computer Systems*, vol. 97, 2019, pp. 361‑378.