International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 13, No. 1, March 2024, pp. 9~19
ISSN: 2089-4864, DOI: 10.11591/ijres.v13.i1.pp9-19

Cost and performance aware scheduling technique for cloud computing environment

Santhosh Kumar Gorva[1], Latha C. Anandachar[2]
[1]Department of Computer Science and Engineering, PES University, Bengaluru, India
[2]Department of Information Science Engineering, RV Institute of Technology and Management, Bengaluru, India

Article Info

Article history:

Received Feb 6, 2023
Revised Apr 28, 2023
Accepted May 10, 2023

Keywords:

Cloud computing
Cost-performance optimization
Heterogeneous server
Scheduling
Workflows

ABSTRACT

Recently, lot of interest have been put forth by researchers to improve workload scheduling in cloud platform. However, execution of scientific workflow on cloud platform is time consuming and expensive. As users are charged based on hour of usage, lot of research work have been emphasized in minimizing processing time for reduction of cost. However, the processing cost can be reduced by minimizing energy consumption especially when resources are heterogeneous in nature; very limited work have been done considering optimizing cost with energy and processing time parameters together in meeting task quality of service (QoS) requirement. This paper presents cost and performance aware workload scheduling (CPA-WS) technique under heterogeneous cloud platform. This paper presents a cost optimization model through minimization of processing time and energy dissipation for execution of task. Experiments are conducted using two widely used workflow such as Inspiral and CyberShake. The outcome shows the CPA-WS significantly reduces energy, time, and cost in comparison with standard workload scheduling model.

Corresponding Author:

Santhosh Kumar Gorva
Department of Computer Science and Engineering, PES University
Banashankari, Bengaluru, Karnataka, India
Email: gsantosh905@gmail.com

1. INTRODUCTION

Cloud computing platforms are widely used for provisioning high-performance computing as web-services for execution of workflows [1]. Recently, wide-range of scientific areas such as bioinformatics, physics, and astronomy have leveraged cloud environment for modeling scientific workflows representing real-world problems [2]; thus, large scientific workflows can be analyzed through simulation in more effective manner [3] with minimal time and cost [4], [5]. The scientific workflow is represented directed acyclic graph (DAG) where edges represent set of task and vertices represent its dependencies. Thus, the forthcoming task will not be initiated until the preceding task is completed [6]–[9]. These dependencies among task makes scheduling in cloud very challenging.

Recently, workflow scheduling in cloud computing platform have gained wide attention across research community [10], a basic architecture of workload scheduling using cloud is shown in Figure 1. However, designing efficient scheduling design adopting currently available heuristic models pose several difficulties such as large scientific workflow prerequisite higher execution time and execution cost. Further, it becomes even more difficult when task demands deadline prerequisite. Extensive work have been done for establishing optimal solution through heuristic algorithm. However, heuristic strategy depends on job order without considering the job scheduling duration. As a result, fails to obtain optimal solution, affecting overall quality of services (QoS) and higher service level agreement (SLA) violation. Thus, the workflow scheduling is considered to non-polynomial (NP-hard) problem [10], [11]. Optimizing cost and time together becomes extremely difficult [12]. For example, if the scheduling design try to minimize cost it increase the execution time; this is because there exist relationship among each other. Many existing model fails to consider virtual machine selection policy in scheduling design, thus cost-makespan optimization problem still exist [13], [14].

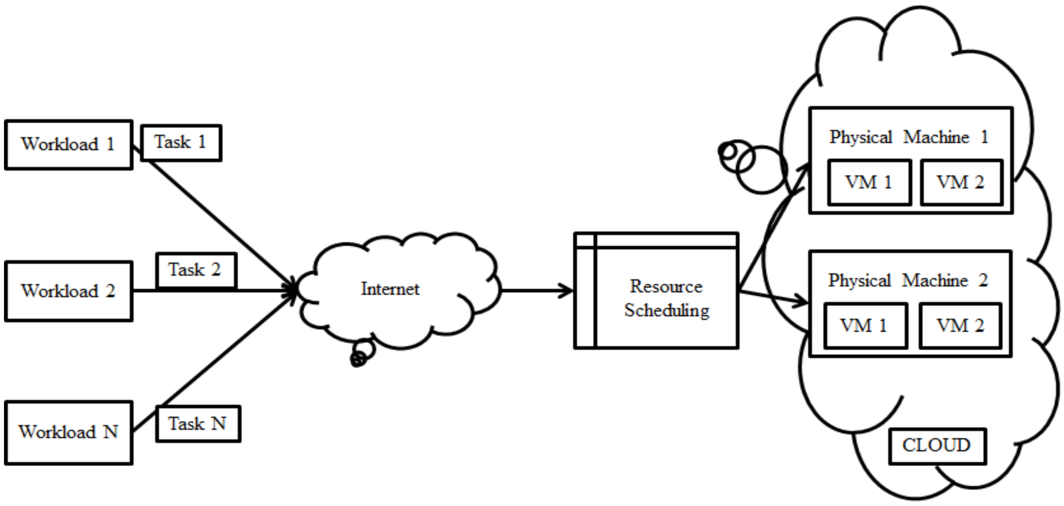


Figure 1. Basic architecture of workload scheduling using cloud

In addressing research problems this paper presents cost and performance aware workload scheduling (CPA-WS) technique for heterogeneous cloud computing (HCC) environment. The model optimizes workload execution cost through energy and processing time minimization constraint; further, the CPA-WS presents an effective queuing model for ideal load balancing between already scheduled tasks with respect to newly arriving task. The manuscript significance is described as:

- This paper presets an effective workload scheduling technique that reduces cost.
- Cost optimization is done through minimization of energy and processing time constraint under heterogeneous computing platform.
- CPA-WS provide an effective load balancing mechanism; thus, reduces buffer overhead and task waiting time.
- CPA-WS achieves much better cost, energy, and processing time efficiency in comparison with energy minimized scheduling (EMS).

The manuscript is arranged as: in section 2, various existing workload scheduling models advantages and limitation is studied. The section 3, provides the mathematical representation of proposed CPA-WS model is given. The result and discussion is given in section 4 and in section 5 the research is concluded with future research direction

2. LITERATURE SURVEY

In the research, survey is conducted for understanding the benefits and limitation of using standard workload scheduling. They focused on designing optimizing energy and cost together to design workflow scheduling for heterogeneous computing platform [15]. Here a main function is modelled for reducing the energy cost and meeting task deadlines considering task information are geographically distributed. Here they divided the task considering different deadline and sorted according to deadline small to high. Finally, an adaptive searching method is designed for optioning effective schedule for workflow execution. It is showed how energy consumption play a significant role on increasing computing cost of service provisioning [16]. Reliability and timeliness are few key metrics in service provisioning. They designed a scheduling design that reduce energy dissipation and meets reliability and timeliness requirement of workflow executions. Here a heuristic solution is obtained through non-linear mixed integer programming problem. First, a scheduling length minimization strategy is modelled for meeting reliability. Second, in reducing energy dissipation designed processor merging strategy by leveraging dynamic voltage frequency scaling (DVFS) technique. Here inefficient machine are switched off and scaling is done at both task and processor level.

Modeled tradeoff to handle unpredictable resource availability nature of cloud computing through adoption of evolutionary computing algorithm [17]. Here a multi-objective parameter optimization model of cost and makespan is considered together. Performance is studied considering various level of interruption and outcome shows better performance than existing models [18]. Modeled an evolutionary computing model namely nested particle swarm optimization (NPSO) and faster version of NPSO namely (FNPSO) optimizing execution of composite workflows [19]. The FNPSO significantly reduce in comparison with NPSO model. Combined Q-learning (QL) and heterogeneous earliest finish time (HEFT) together for designing an effective scheduling technique namely QL-HEFT [20]. The QL-HEFT is intended to reduce computation time. The reward function in QL is updated using upward rank outcome of HEFT. This aid in improving learning efficiency of QL algorithm. The QL first obtain an optimal order of task and then finds the suitable machine for execution of task utilizing earliest finishing time. Designed a scheduling design considering contention awareness for workflow execution [21]. In minimizing makespan a list scheduling heuristic with endpoint contention awareness is modelled. A ranking mechanism is introduced to schedule task to computational machines and modeled a rescheduling design to improve scheduling efficiency.

They designed a workflow scheduling design adopting evolutionary computing model to meet task deadline by optimizing cost namely deadline-constrained cost optimization for hybrid clouds (DCOH) [22]. Further, improved DCOH by incorporating multi-objective parameter by optimizing makespan and cost together under hybrid cloud platform. Workflow applications scheduling designed meeting application deadline and cost together [23]. Here they improved priority selection design for establishing the order of task and during allocation of computational resource, budget and cost ratio are used correlate among budget, and deadline constraint. In improving success rate (i.e. reliability) certain decision are discarded through discarding mechanism.

They showed scheduling model in cloud must meet user deadline prerequisite and SLA's. Here they adopted a multi-cloud platform for meeting stream workflow application performance requirement and cost reduction [24]–[26]. They design a fault-tolerant scheduling design for workflow execution leveraging multi-cloud platform. Further, the model assures in meeting reliability requirement and with reduced cost [26]. Here they employed continuous probability distribution for analyzing failure rate and reliability. Then, a mathematical model to measure cost of executing using multi-cloud platform is given followed by defining fault-tolerant workflow scheduling design by assuring reliability, reducing cost and execution time. However, could not guarantee in meeting cost constraint of application requirement because of poor load balancing mechanism. In addressing aforementioned issues, tin next section presents cost, and performance aware scheduling technique under heterogeneous cloud environment.

3. COST AND PERFORMANCE AWARE SCHEDULING TECHNIQUE FOR CLOUD COMPUTING ENVIRONMENT

This section present CPA-WS technique for executing scientific workflow in HCC environment. The workload scheduling architecture of CPA-WS is shown in Figure 2. The CPA-WS technique is modelled to schedule task with minimal cost by optimizing energy consumption meeting task deadlines performance prerequisite without causing any congestion in HCC environment. Here an effective task queueing methodology is modelled for load balancing. The task queuing methodology is composed of o HCC server $T_1, T_2, \dots, T_o$ with capacity $n_1, n_2, \dots, n_o$ and its computational capability is $t_1, t_2, \dots, t_o$. Let the HCC server T_j is composed of n_j identical servers with computational capability t_j. The arrival load α is exponentially distributed with randomness (s) and mean average $(\underline{s})$ $1/\alpha$ considering 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 $1 \le j \le o, \alpha = \alpha_1 + \alpha_2 + \dots + \alpha_o$. A HCC server T_j retains a queue with boundless capacity for task in queue which is waiting to be executed when whole server n_j is busy. The scheduling is done according to first come first serve with exponential randomness s and mean $\underline{s}$. The n_j servers of HCC server T_j have similar computation capacity t_j. Therefore, the computation time with exponential randomness is measure using (1):

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

with mean:

$$\underline{y_j} = \frac{\underline{s}}{t_j} \tag{2}$$

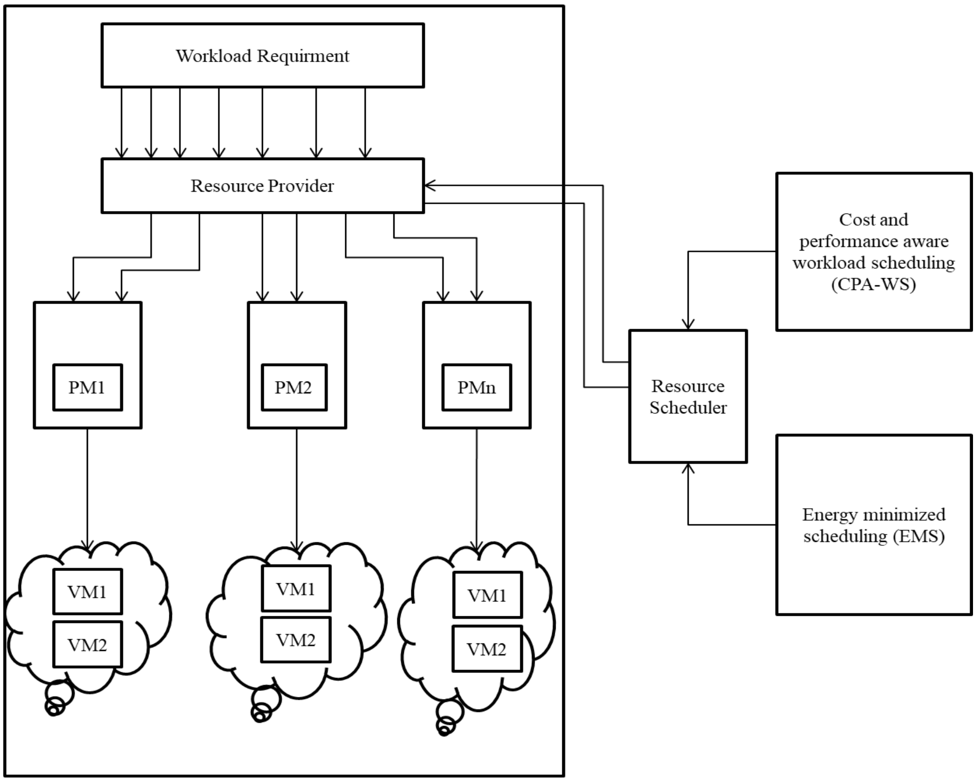


Figure 2. Workload scheduling architecture of CPA-WS

The mean task (i.e., the mean success rate) which are possibly be finished by HCC server within T_j is measured as (3).

$$\beta_j = \frac{1}{y_j} \tag{3}$$

The mean amount of time the server will be busy i.e., the resource utilization is measured as (4).

$$\gamma_j = \frac{\alpha_j}{n_j \beta_j} = \frac{\alpha_j y_j}{n_j} = \frac{\alpha_j s}{n_j t_j} \tag{4}$$

Let $p_{j,l}$ defines the probability that l task resides in queue or can be handled in HCC server T_j is measured as (5):

$$p_{j,l} = \{p_{j,0} \frac{(n_j \gamma_j)^l}{l!},\ l < n_j;\ p_{j,0} \frac{n_j^{n_j} \gamma_j^l}{l!},\ l \geq n_j; \tag{5}$$

where:

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

the probability of newly arriving workflow task that will resides in HCC server T_j when whole server in T_j is busy is measured as (7).

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

The average workflow task that are currently being executed/waiting in HCC server T_j is measured as (8).

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

In similar manner to (8), the average workflow task completion time of HCC server T_j is measured as (9).

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

For easiness the mean workflow task computation time of HCC server T_j is measured as (10).

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

The energy need for completing task execution is measured as (11):

$$Q = aCV^2F = \delta t^{\mu} \tag{11}$$

where a represent the task characteristics, V, C, F, and t depicts voltage, load capacitance, clock frequency, processor speed, respectively. In (11), the δ is measured as (12):

$$\delta = \frac{ab^2C}{c^{2\rho+1}} \tag{12}$$

in (12), the parameter b and ρ defines constant higher than zero. The μ is measured as (13).

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

Existing method consider both δ and μ across server; however, in this work it is not the case because of HCC environment adopted; thus, we have different value of δ and μ. Here we consider two different energy type such as static energy and dynamic energy type. In static energy type, the computational machine will not perform any task and energy consumed is measured as (14).

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

Similarly, in dynamic energy type the computational machine will execute task/will be waiting for task arrival and the energy consumed is measured as (15):

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

this work aimed at allocating ideal resource with minimal execution cost by optimizing energy and processing time for executing workload task under HCC environment with varying processing speed and power consumption.

Let consider a o HCC server with size of $n_1, n_2, \dots, n_o$, with dynamic energy dissipation and computation capacity for execution of workflow with prerequisite $\underline{s}$ with task arrival rate α, and have load distribution $\alpha_1, \alpha_2, \dots, \alpha_o$ in achieving high performance efficiency is obtained through following minimization function:

$$minU(\alpha_1, \alpha_2, \dots, \alpha_o) \tag{16}$$

the (16) is subjected to constraint described in (17) and (18):

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

where,

$$G(\alpha_1, \alpha_2, \dots, \alpha_o) = \alpha_1 + \alpha_2 + \cdots + \alpha_o \tag{18}$$

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

Let consider a o HCC server with size of $n_1, n_2, \dots, n_o$, with dynamic energy dissipation and computation capacity for execution of workflow with prerequisite $\underline{s}$ with task arrival rate α, and have load distribution $\alpha_1, \alpha_2, \dots, \alpha_o$ in reducing energy consumption is obtained through following minimization function as (19).

$$minQ(\alpha_1, \alpha_2, \dots, \alpha_o) \tag{19}$$

The (19) is subjected to constraint described in (20) and (21):

$$G(\alpha_1, \alpha_2, \dots, \alpha_o) = \alpha \tag{20}$$

where,

$$G(\alpha_1, \alpha_2, \dots, \alpha_o) = \alpha_1 + \alpha_2 + \dots + \alpha_o \tag{21}$$

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

Let's consider heterogeneous computing platform T_j, the cost outcome can be measured through inverse proportion of execution time using (22).

$$C = \frac{1}{U_j} \tag{22}$$

However, the proposed design considers energy factor Q_j into consideration for measuring cost as defined in (23).

$$S_j = Q_j U_j \tag{23}$$

The mean cost-performance S considering o heterogeneous computing platform $T_1, T_2, \dots, T_o$ is measured through (24).

$$S(\alpha_1, \alpha_2, \dots, \alpha_o) = \frac{\alpha_1}{\alpha} S_1 + \frac{\alpha_2}{\alpha} S_2 + \dots + \frac{\alpha_o}{\alpha} S_o \tag{24}$$

For simplicity the (24) is rewritten as (25).

$$= \frac{\alpha_1}{\alpha} Q_1 U_1 + \frac{\alpha_2}{\alpha} Q_2 U_2 + \dots + \frac{\alpha_o}{\alpha} Q_o U_o \tag{25}$$

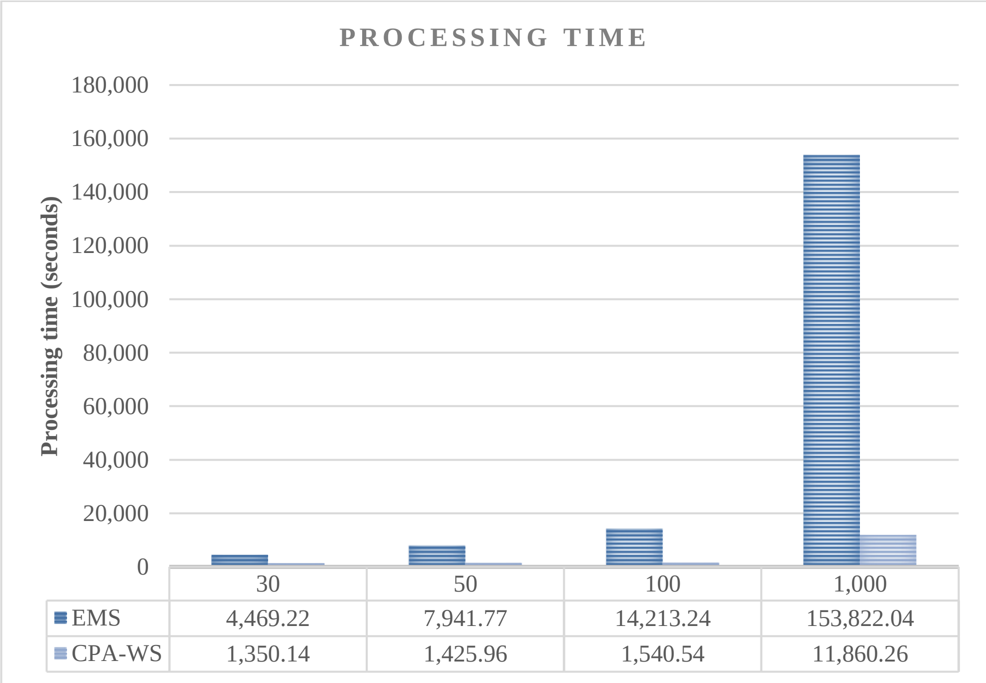
Here the workload tasks are scheduled by minimizing (16) and (19), and meeting constraint defined in (17), (18), (20), and (21) in order to bring tradeoffs between performance and cost.

4. SIMULATION RESULTS

Experiment is conducted for evaluating CPA-WS and EMS [16]. CloudSim3 [26] is used modelling workload scheduling algorithm [27]. The complex workload Inspiral and CyberShake is used [28], [29] because it is widely used validating various scheduling model [30]–[34], where Inspiral requires more central processing unit (CPU) and memory; however, the CyberShake requires CPU and I/O resources [28], [29]. Time efficiency, energy consumption and cost efficiency is metrics used measuring performance of CPA-WS and EMS.

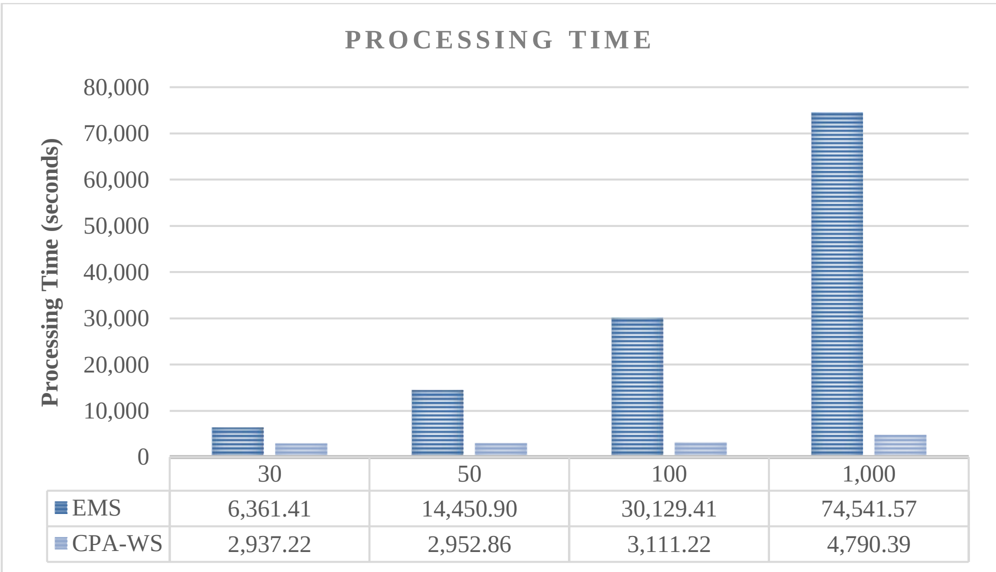
4.1. Time efficiency vs workload size

Here the time efficiency of CPA-WS and EMS is measured by varying the Inspiral and CyberShake workload task size from 30 to 1,000. The time efficiency is measured as time taken to complete the task, lesser time indicates better performance. The Figure 3 shows time taken to complete task using CPA-WS and EMS for varied Inspiral workload size. Similarly, the Figure 4 shows time taken to complete task using CPA-WS and EMS for varied CyberShake workload size. From experiments, it can be seen the CPA-WS is very efficient for both smaller and larger workload; however, EMS achieves very poor result for larger workload considering both Inspiral and CyberShake workload. 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.



	30	50	100	1,000
EMS	4,469.22	7,941.77	14,213.24	153,822.04
CPA-WS	1,350.14	1,425.96	1,540.54	11,860.26

Figure 3. Time efficiency with different Inspiral workload size



	30	50	100	1,000
EMS	6,361.41	14,450.90	30,129.41	74,541.57
CPA-WS	2,937.22	2,952.86	3,111.22	4,790.39

Figure 4. Time efficiency with different CyberShake workload size

4.2. Energy consumption vs workload size

Here the energy consumption is measured of CPA-WS and EMS is measured by varying the Inspiral and CyberShake workload task size from 30 to 1,000. The energy consumption is measured as amount of power consumed in watt to complete the task, lesser watt indicates better performance. The Figure 5 shows energy consumed to complete task using CPA-WS and EMS for varied Inspiral workload size. Similarly, the Figure 6 shows energy consumed to complete task using CPA-WS and EMS for varied CyberShake workload size. From experiments, it can be seen the CPA-WS is very energy efficient for both smaller and larger workload; however, EMS achieves significantly higher energy for both smaller and larger workload considering both Inspiral and CyberShake workload. 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.

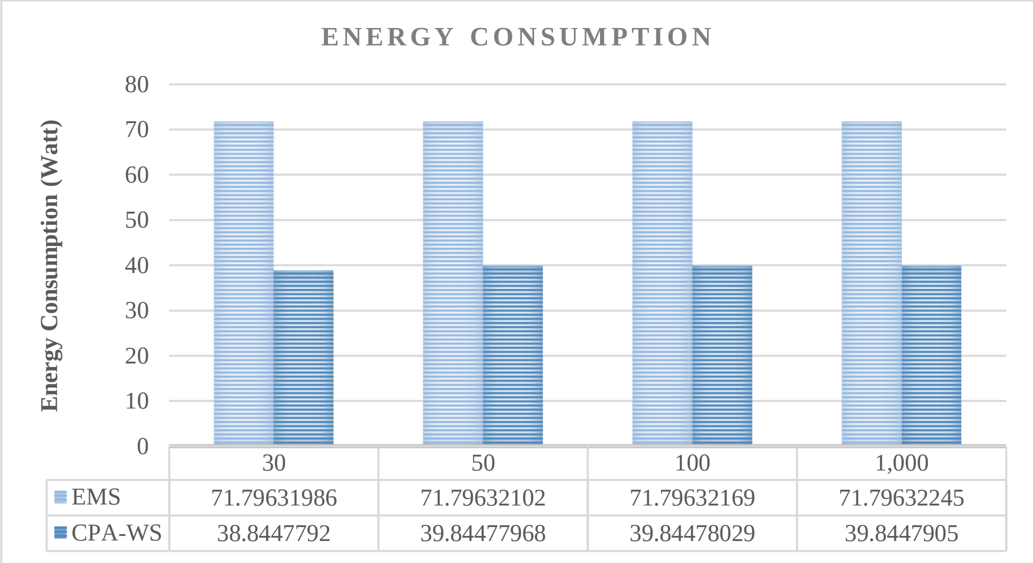


Figure 5. Energy efficiency with different Inspiral workload size

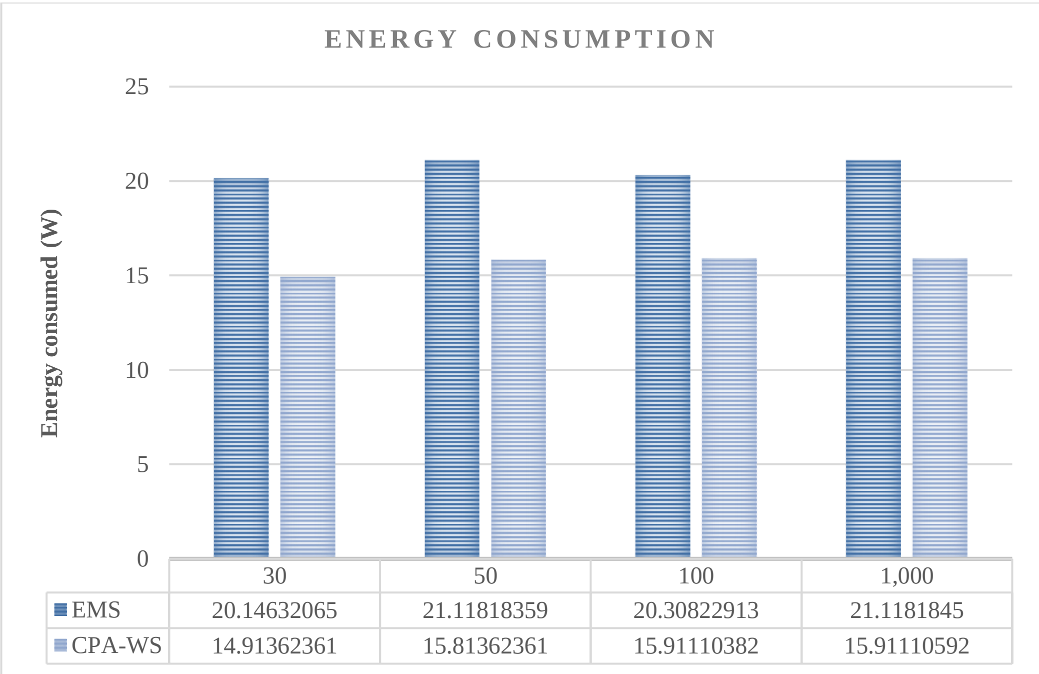


Figure 6. Energy efficiency with different CyberShake workload size

4.3. Cost efficiency vs workload size

Here the cost efficiency of CPA-WS and EMS is measured by varying the Inspiral and CyberShake workload task size from 30 to 1,000. The cost efficiency is measured as energy consumed and time taken to complete the task, lesser value indicates better performance. The Figure 7 shows cost incurred to complete task using CPA-WS and EMS for varied Inspiral workload size. Similarly, the Figure 8 shows cost incurred to complete task using CPA-WS and EMS for varied CyberShake workload size. From experiments it can be seen the CPA-WS is very efficient for both smaller and larger workload; however, EMS achieves very poor result for larger workload considering both Inspiral and CyberShake workload. The CPA-WS reduce computation cost by 83.13% over EMS for Inspiral workload. Similarly, the CPA-WS reduce computation cost by 78.851% over EMS for CyberShake workload.

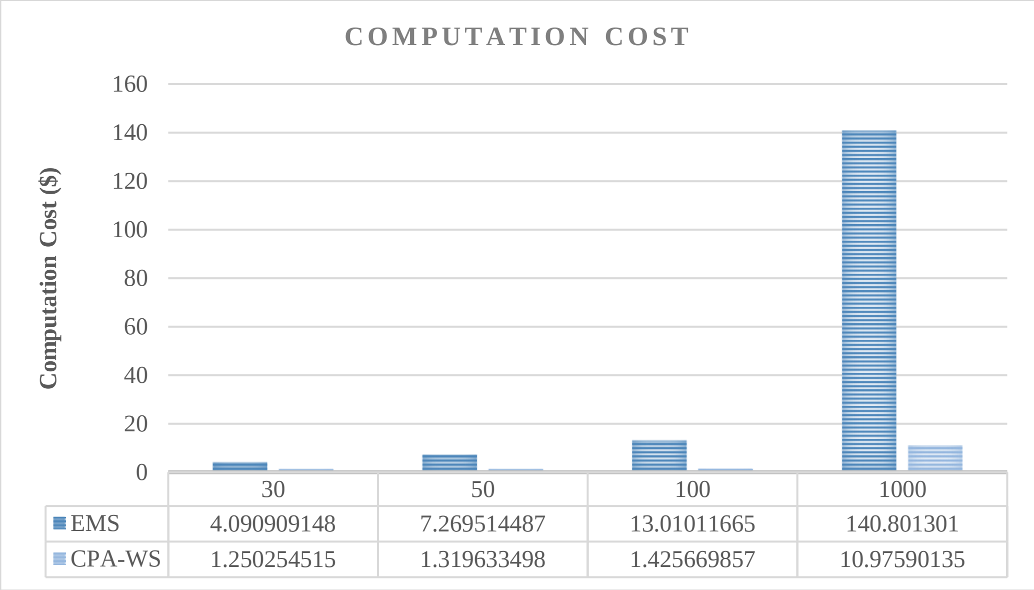


Figure 7. Cost efficiency with different Inspiral workload size

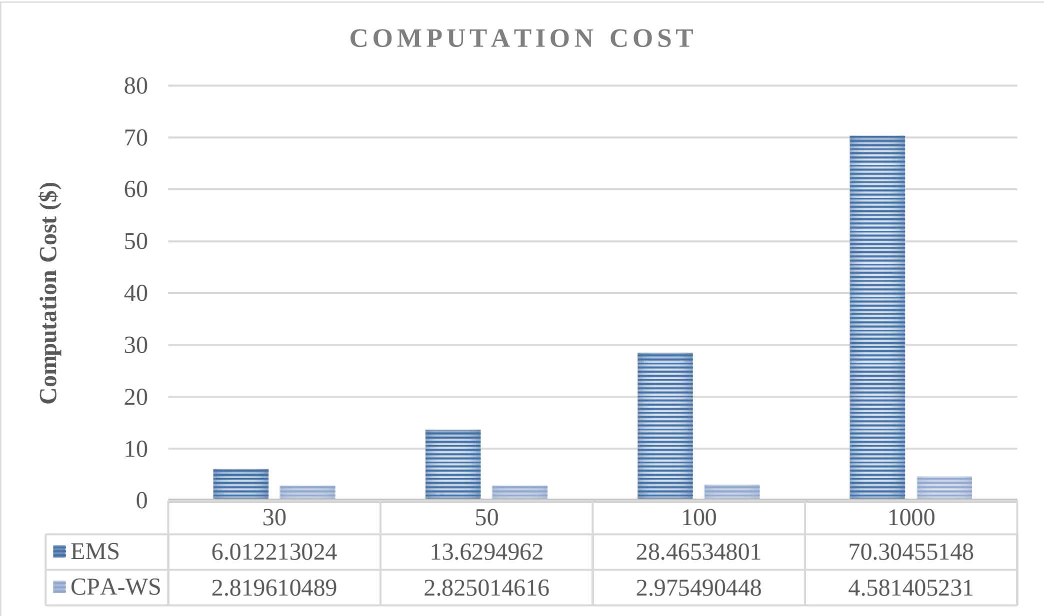


Figure 8. Cost efficiency with different CyberShake workload size

5. CONCLUSION

Here we studied different workload scheduling technique for execution of real-time workload employing cloud-computing platform. The study identified majority of existing workload scheduling focused on reducing cost through minimization of processing time, energy, and delay; however, very limited have focused on addressing cost minimization considering both energy and processing time together under heterogeneous cloud platform. This paper designed a workload scheduling technique by presenting energy and processing time optimization constraint for reducing computation cost. Further, an effective load balancing technique is presented for reducing the waiting time; the adoption of such strategy significantly aid in utilizing 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 result, it can be stated CPA-WS computation cost performance gets profitable with increasing in workload size in comparison with EMS. Thus, are suitable for provisioning both smaller and larger workload with high profitability. Future work would consider improving resource usage efficiency and consider provisioning security for workload execution for performing different kind of task.

REFERENCES

[1] A. A. Nasr, N. A. El-Bahnasawy, G. Attiya, and 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, pp. 3765–3780, Apr. 2019, doi: 10.1007/s13369-018-3664-6.

[2] J. Sahni and 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, pp. 2–18, Jan. 2018, doi: 10.1109/TCC.2015.2451649.

[3] S. Tuli, S. Ilager, K. Ramamohanarao, and R. Buyya, "Dynamic scheduling for stochastic edge-cloud computing environments using A3C learning and residual recurrent neural networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 940–954, Mar. 2022, doi: 10.1109/TMC.2020.3017079.

[4] Q. Jiang, V. C. M. Leung, H. Tang, and H.-S. Xi, "Adaptive scheduling of stochastic task sequence for energy-efficient mobile cloud computing," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3022–3025, Sep. 2019, doi: 10.1109/JSYST.2019.2922436.

[5] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and 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, pp. 713–726, Jul. 2018, doi: 10.1109/TSC.2015.2466545.

[6] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: a comprehensive analysis," *Journal of Network and Computer Applications*, vol. 66, pp. 64–82, May 2016, doi: 10.1016/j.jnca.2016.01.018.

[7] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei, and S. Hu, "Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT," *Future Generation Computer Systems*, vol. 93, pp. 278–289, Apr. 2019, doi: 10.1016/j.future.2018.10.046.

[8] M. A. Rodriguez and 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, Jun. 2017, doi: 10.1145/3041036.

[9] N. Anwar and H. Deng, "Elastic scheduling of scientific workflows under deadline constraints in cloud computing environments," *Future Internet*, vol. 10, no. 1, p. 5, Jan. 2018, doi: 10.3390/fi10010005.

[10] Y. Zhao, X. Fei, I. Raicu, and S. Lu, "Opportunities and challenges in running scientific workflows on the cloud," in *2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Oct. 2011, pp. 455–462, doi: 10.1109/CyberC.2011.80.

[11] A. M. Manasrah and H. Ba Ali, "Workflow scheduling using hybrid GA-PSO algorithm in cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–16, 2018, doi: 10.1155/2018/1934784.

[12] S. Yassir, Z. Mostapha, and T. Claude, "Workflow scheduling issues and techniques in cloud computing: a systematic literature review," in *Cloud Computing and Big Data: Technologies, Applications and Security*, M. Zbakh, M. Essaaidi, P. Manneback, and C. Rong, Eds., in Lecture Notes in Networks and Systems, vol. 49. Cham: Springer International Publishing, 2019, pp. 241–263, doi: 10.1007/978-3-319-97719-5_16.

[13] Y. Qin, H. Wang, S. Yi, X. Li, and L. Zhai, "An energy-aware scheduling algorithm for budget-constrained scientific workflows based on multi-objective reinforcement learning," *J Supercomput The Journal of Supercomputing*, vol. 76, no. 1, pp. 455–480, Jan. 2020, doi: 10.1007/s11227-019-03033-y.

[14] J. K. Konjaang and 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)*, Baltimore, MD, USA: IEEE, May 2020, pp. 162–168, doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00038.

[15] X. Li, W. Yu, R. Ruiz, and J. Zhu, "Energy-aware cloud workflow applications scheduling with geo-distributed data," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 891–903, Mar. 2022, doi: 10.1109/TSC.2020.2965106.

[16] B. Hu, Z. Cao, and M. Zhou, "Energy-minimized scheduling of real-time parallel workflows on heterogeneous distributed computing systems," *IEEE Transactions on Services Computing.*, vol. 15, no. 5, pp. 2766–2779, Sep. 2022, doi: 10.1109/TSC.2021.3054754.

[17] T.-P. Pham and T. Fahringer, "Evolutionary multi-objective workflow scheduling for volatile resources in the cloud," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1780–1791, Jul. 2022, doi: 10.1109/TCC.2020.2993250.

[18] H. Li, B. Wang, Y. Yuan, M. Zhou, Y. Fan, and Y. Xia, "Scoring and dynamic hierarchy-based NSGA-II for multiobjective workflow scheduling in the cloud," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 2, pp. 982–993, Apr. 2022, doi: 10.1109/TASE.2021.3054501.

[19] A. Song, W.-N. Chen, X. Luo, Z.-H. Zhan, and J. Zhang, "Scheduling workflows with composite tasks: a nested particle swarm optimization approach," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 1074–1088, Mar. 2022, doi: 10.1109/TSC.2020.2975774.

[20] Z. Tong, X. Deng, H. Chen, J. Mei, and H. Liu, "QL-HEFT: a novel machine learning scheduling scheme base on cloud computing environment," *Neural Computing and Applications*, vol. 32, no. 10, pp. 5553–5570, 2020, doi: 10.1007/s00521-019-04118-8.

[21] 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, pp. 1137–1150, Apr. 2022, doi: 10.1109/TASE.2020.3046673.

[22] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, and M. Chen, "Cost and makespan-aware workflow scheduling in hybrid clouds," *Journal of Systems Architecture*, vol. 100, p. 101631, Nov. 2019, doi: 10.1016/j.sysarc.2019.08.004.

[23] 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, pp. 4939–4949, Dec. 2021, doi: 10.1109/JSYST.2021.3087527.

[24] M. Barika, S. Garg, A. Chan, and R. N. Calheiros, "Scheduling algorithms for efficient execution of stream workflow applications in multicloud environments," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 860–875, Mar. 2022, doi: 10.1109/TSC.2019.2963382.

[25] X. Tang, "Reliability-aware cost-efficient scientific workflows scheduling strategy on multi-cloud systems," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2909–2919, Oct. 2022, doi: 10.1109/TCC.2021.3057422.

[26] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and 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, Jan. 2011, doi: 10.1002/spe.995.

[27] W. Chen and E. Deelman, "Workflowsim: a toolkit for simulating scientific workflows in distributed environments," in *2012 IEEE 8th International Conference on E-Science*, Chicago, IL, USA: IEEE, Oct. 2012, pp. 1–8, doi: 10.1109/eScience.2012.6404430.

[28] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, and K. Vahi, "Characterization of scientific workflows," in *2008 Third Workshop on Workflows in Support of Large-Scale Science*, Austin, TX, USA: IEEE, Nov. 2008, pp. 1–10, doi: 10.1109/WORKS.2008.4723958.

[29] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 682–692, Mar. 2013, doi: 10.1016/j.future.2012.08.015.

[30] A. Choudhary, M. C. Govil, G. Singh, L. K. Awasthi, and E. S. Pilli, "Task clustering-based energy-aware workflow scheduling in cloud environment," in *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)*, Exeter, United Kingdom: IEEE, Jun. 2018, pp. 968–973, doi: 10.1109/HPCC/SmartCity/DSS.2018.00160.

[31] V. Singh, I. Gupta, and P. K. Jana, "An energy efficient algorithm for workflow scheduling in IaaS cloud," *Journal of Grid Computing*, vol. 18, no. 3, pp. 357–376, Sep. 2020, doi: 10.1007/s10723-019-09490-2.

[32] Y. Gao, S. Zhang, and J. Zhou, "A hybrid algorithm for multi-objective scientific workflow scheduling in IaaS cloud," *IEEE Access*, vol. 7, pp. 125783–125795, 2019, doi: 10.1109/ACCESS.2019.2939294.

[33] K. Dubey, M. Y. Shams, S. C. Sharma, A. Alarifi, M. Amoon, and A. A. Nasr, "A management system for servicing multi-organizations on community cloud model in secure cloud environment," *IEEE Access*, vol. 7, pp. 159535–159546, 2019, doi: 10.1109/ACCESS.2019.2950110.

[34] Y. Xie *et al.*, "A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment," *Future Generation Computer Systems*, vol. 97, pp. 361–378, Aug. 2019, doi: 10.1016/j.future.2019.03.005.

BIOGRAPHIES OF AUTHOR

Santhosh Kumar Gorva earned his bachelor's of engineering B.E. degree in computer science and engineering from VTU, Belagavi in 2005. He has obtained his master's degree in M.Tech. (networking and internet engineering) from VTU, Belgavi in 2010. Currently, he is a research scholar at VTU, Belgavi doing his Ph.D. in computer science and engineering; also, he working as an assistant professor at PES University. He has attended many workshops and induction programs conducted by various universities. His areas of interest are cloud computing and internet of things. He can be contacted at email: gsantosh905@gmail.com.

Dr. Latha C. Anandachar is a professor and head in the information science and engineering, RVITM, Bengaluru with an experience of 25 years in teaching. She is qualified in bachelor's and master's degrees in computer science and engineering, and a Ph.D. in computer science and engineering in the area of cloud computing and data mining. Her areas of interest are data mining, computer vision, machine learning, artificial intelligence, and internet of things. She can be contacted at email: lathacadr@gmail.com.