**International Journal of Reconfigurable and Embedded Systems (IJRES)**
Vol. 13, No. 1, March 2024, pp. 210~216
ISSN: 2089-4864, DOI: 10.11591/ijres.v13.i1.pp210-216 

# Task level energy and performance assurance workload scheduling model in distributed computing environment

**Jagadevi Bakka[1], Sanjeev C. Lingareddy[2]**
[1]Department of Computer Science and Engineering, East Point College of Engineering and Technology, Bangalore, India
[2]Department of Computer Science and Engineering, Sri Venkateshwara College of Engineering, Bangalore, India

**Article Info**

***Article history:***

Received Jul 22, 2022
Revised Jun 30, 2023
Accepted Aug 5, 2023

***Keywords:***

Cloud computing
Direct acyclic graph
MapReduce
Service level agreement
Task level energy and performance assurance

**ABSTRACT**

Scientific workload execution on a distributed computing platform such as a cloud environment is time-consuming and expensive. The scientific workload has task dependencies with different service level agreement (SLA) prerequisites at different levels. Existing workload scheduling (WS) designs are not efficient in assuring SLA at the task level. Alongside, induces higher costs as the majority of scheduling mechanisms reduce either time or energy. In reducing, cost both energy and makespan must be optimized together for allocating resources. No prior work has considered optimizing energy and processing time together in meeting task level SLA requirements. This paper presents task level energy and performance assurance-workload scheduling (TLEPA-WS) algorithm for the distributed computing environment. The TLEPA-WS guarantees energy minimization with the performance requirement of the parallel application under a distributed computational environment. Experiment results show a significant reduction in using energy and makespan; thereby reducing the cost of workload execution in comparison with various standard workload execution models.



***Corresponding Author:***

Jagadevi Bakka
Department of Computer Science and Engineering, East Point College of Engineering and Technology
Jnana Prabha, East Point Campus, Virgo Nagar Post, Avalahalli, Bengaluru, Karnataka, India
Email: jagdevi198@rediffmail.com

## 1. INTRODUCTION

Recently, the big data framework is being emphasized in wide-range of complex scientific applications. For example, scientific complex workloads such as Montage (Figure 1) which are widely used for the scientific and business investigation [1]. The significant growth has led to some serious challenges such as low-latency and cost effectiveness for storing, communication, and processing [2]. In addressing such problem, cloud computing such as Amazon EC2, Pegasus, and MapReduce [3], [4] are being used. Cloud platforms provide high quality storage and computing resources like networks, services, and applications. for the execution of the scientific complex workloads [5]. The scientific workload generally represented as a directed acyclic graph (DAG) where there exist dependencies among task as shown in Figure 1. This makes scheduling of cloud resource for workload execution a challenges task [6].

Recently, the researchers are using the cloud services to schedule the workload [7]–[10]. The Figure 2 depicts a simple architecture for the workload scheduling (WS) in a cloud environment. However, to design an efficient workload-scheduling model by reviewing the existing models presents various challenges like executing larger and more complex scientific workloads, which requires more execution time and increases the energy usage for the execution. It becomes more challenging when the tasks have to be executed in a given deadline. Moreover, the scheduling of workload is deemed an NP-hard problem [11]–[14]. In fact, optimizing both time and cost is a very challenging task in WS [15]. For an instance, if a scheduling model seeks to reduce the energy usage, the time required to complete a given task increases. This is because both the energy and time are linked. Since many existing models do not consider task level service level agreement (SLA) into account while creating schedules, the energy and makespan optimization problem persists [16]. To address the above problem, this paper presents a task level energy and performance assurance-workload scheduling (TLEPA-WS) technique. The TLEPA-WS assures SLA at task level [17], [18]. Research contribution for this research is mentioned as: i) the TLEPA-WS is efficient in minimizing energy with performance assurance of workload task level SLA prerequisite. No prior work has considered such scheduling mechanism and ii) the TLEPA-WS model reduces energy consumption, makespan, and computational cost in comparison with energy-min scheduling.

The paper organization: in section 2, study various existing WS design for distributed computational environment. In section 3, the TLEPA workload-scheduling model is explained. The simulation study is given in section 4. In last section, the research is concluded and future research enhancement is given.

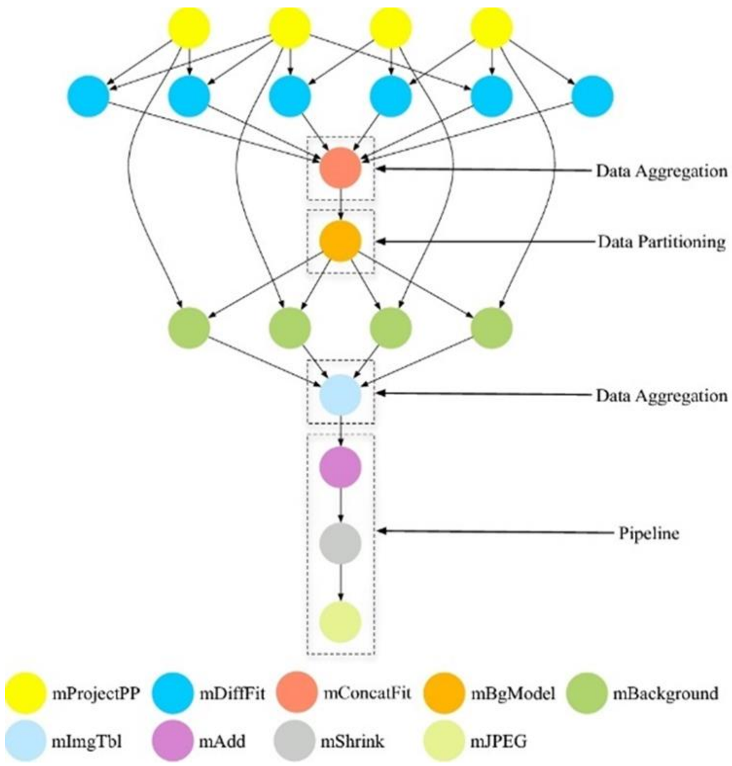


Figure 1. Sample representation of Montage workload [2]

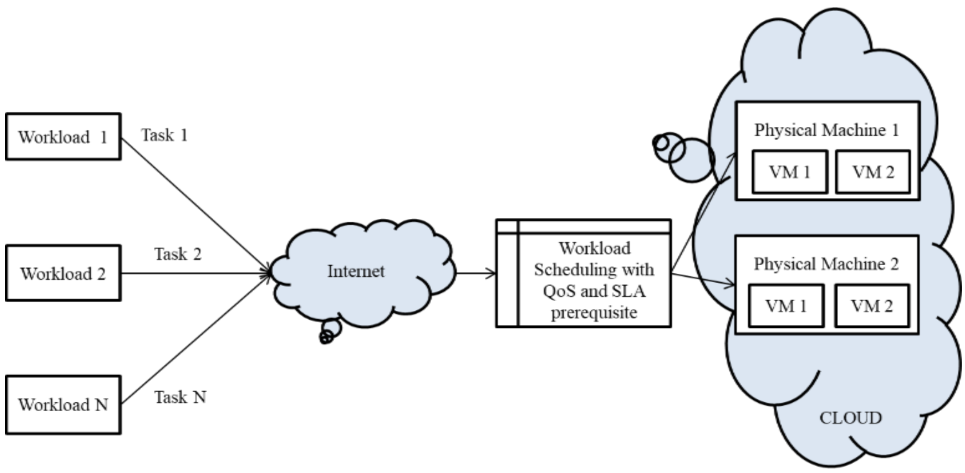


Figure 2. The standard resource-scheduling framework

## 2. LITERATURE SURVEY

This section studies various workload-scheduling model to leveraging distributed computing platform such as cloud environment. Table 1 comprises of several survey regarding this research. Here the advantage as well as limitation is mentioned of all the survey. In addressing aforementioned limitation in next section, presents task level performance and energy assurance WS technique for cloud environment.

Table 1. Survey table

| Method | Advantage and limitation | Metrics and workload used |
|---|---|---|
| Hu *et al.* [19] proposed an algorithm, which reduces the scheduling length by providing proper reliability and also another algorithm for the processor has been proposed which reduces the execution time using the dynamic voltage and frequency scaling (DVFS) method to minimize the energy consumption. | This model has performed better in terms of energy and computation time. Furthermore, this model has also addressed the issue of reliability during the execution of the tasks. However, the given model consumes more cost for the execution of the task of the workflow, hence is not cost efficient. | Energy consumption and computation time are metric used. Montage, cybershake, laser interferometer gravitational wave observatory (LIGO) and some synthetic workflows are used. |
| Pham and Fahringer [20] proposed a multi-objective workflow scheduling method which provides a trade-off solution to reduce the cost and increase the makespan. | This method performs better in terms of makespan and cost. However, this model has not addressed the issue of energy consumption during the execution of the workflow in the clouds. | Makespan, cost, deadline, and budget are metric used. cybershake, epigenomics, inspiral, Montage and sipht are workload used. |
| Zhou *et al.* [21] used two method deadline-constrained cost optimization for hybrid clouds (DCOH) and multi-objective optimization for hybrid clouds (MOH) which optimizes the makespan and reduces the cost. | This method has reduced the execution cost by 100% when compared with the deadline constraint models. However, in order to reduce the cost this model has consumed more energy and this model is good for hybrid clouds, not good for edge cloud platforms. | Energy, cost, and makespan are metric used. Cybershake, epigenomics, inspiral, Montage, and sipht are workload used. |
| Wang *et al.* [22] presented an algorithm, deadline and budget constrained heterogenous scheduling (DMW-HDBS) to select the task, check the priority of the task so that the resources should be allocated to the most priority task first and then to the least priority tasks. | This model has showed good performance and has increased the success rate for the execution of the task in the multi-workflow scheduling platforms using their method. However, this model has not addressed the issue of energy consumption by each of the task during the execution of the workflow. | Cost, deadline, and makespan are metric used. Lille, Sophia, and Rennes are workflows used. |
| Tang [23] proposed a strategy to provide a cost-efficient and reliability method for the multi-cloud network. In this model, they have developed a fault-tolerant workflow-scheduling framework, which increase the reliability and reduces the cost during the execution of the task. | The results show that the model has outperformed when compared with the existing methods in terms of reliability and cost. However, the model has failed to address the issue of energy consumption in their model. | Execution time, cost and reliability are metrics used. LIGO and epigenomics are workload used. |
| Malik *et al.* [24] proposed an algorithm based on the classification of the task and the threshold in order to provide better resources for the execution of the task. | This model has used particle swarm optimization (PSO) to increase the performance of the model and reduce the energy consumption, makespan and load balancing. However, this model consume more time to execute as it has to run the algorithm first and then run the PSO algorithm to get its result, hence, is not time-efficient method. | Makespan, energy consumption, load balance are metric used. Cybershake, epigenomics, LIGO, Montage, and sipht are workflows used. |

## 3. ENERGY EFFICIENT MULTI-SENSORY TARGET TRACKING METHOD FOR WIRELESS SENSOR NETWORK

This section present TLEPA-WS algorithm in cloud computing environment as shown in Figure 3. The TLEPA provide ideal performance considering varying SLA condition at task level in parallel computational framework. In this section, we design and task-level SLA-based WS algorithm for monitoring the process and solves the various attributes and constraints; in here various constraint is considered such as priority, task dependency, parallel computation and various attribute such as energy and makespan is considered for minimization.

### 3.1. Algorithm

This section presents the TLEPA algorithm for execution of complex workload in distribute computing platform in Algorithm 1. In this section, we design and task-level SLA-based WS algorithm for monitoring the process and solves the various attributes and constraints; in here various constraint is considered such as priority, task dependency, parallel computation and various attribute such as energy and makespan is considered for minimization.

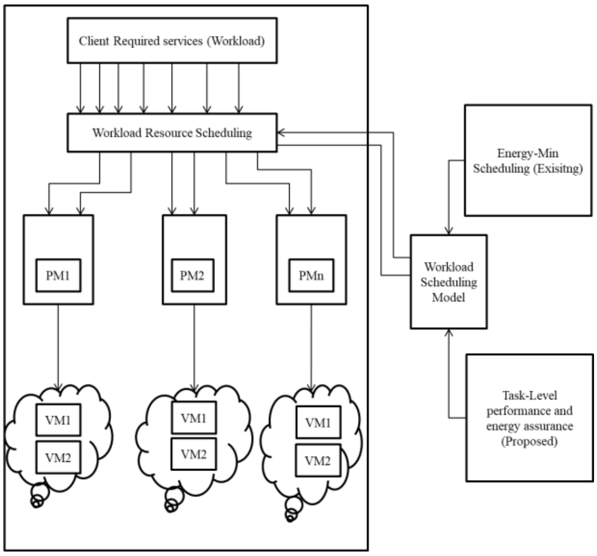


Figure 3. The workload-scheduling framework

Algorithm 1. The algorithm of adaptive workload scheduling assuring SLA at task level
Step 1. Design a task and power model.
Step 2. Compute the given lower performance constraint limits.
Step 3. Mechanism for adaptive task scheduling in which tasks that have not been scheduled are tested to see if the necessary resources are available for their execution.
Step 4. Design the tasks in DAG. The number of levels in the DAG is w, where w is the number of levels with varying energy and makespan SLA constraint in the DAG.
Step 5. Non-urgent tasks fall under the category of level 1.
Step 6. TLEPA organizes tasks into levels 1, 2, 3, 4, etc., and keeps track of all of them.
Step 7. Furthermore, unless the task in level m is finished, m+1 level cannot be executed. The monitoring process is the same for each level m.
Step 8. As tasks are independent and have different levels as a result, they are scheduled using an adaptive task scheduling technique.

### 3.2. Service level agreement optimization

In this given section, we calculate the lower bound. Assume a parameter $\mathrm{X}$ which represents the amount of work done for a given parallel task o, this can be represented using as (1).

$$\mathrm{X} = \mathrm{x}_1 + \mathrm{x}_2 + \cdots + \mathrm{x}_\mathrm{o} = \mathbb{P}_1\, \mathrm{s}_1 + \mathbb{P}_1\, \mathrm{s}_2 + \cdots + \mathbb{P}_\mathrm{o} \mathrm{s}_\mathrm{o} \tag{1}$$

The minimal energy and optimal length required for an optimal scheduling is represented using $\mathrm{F}'$ and $\mathrm{U}'$ respectively. By considering, all these parameters, the lower bound can be evaluated which help to reduce the makespan in a multi-level workload-scheduling model. In (2) represent that the lower bound reduces the makespan and the given (3) represents the lower bound to reduce the consumption of energy.

$$\mathrm{U}' \geq \left(\frac{\mathrm{n}}{\tilde{\mathrm{F}}}\left(\frac{\mathrm{X}}{\mathrm{n}}\right)^{\mathrm{b}}\right)^{1/(\mathrm{b}-1)} \tag{2}$$

$$\mathrm{F}' \geq \mathrm{n}\left(\frac{\mathrm{X}}{\mathrm{n}}\right)^{\mathrm{b}} \left(\tilde{\mathrm{U}}^{\mathrm{b}-1}\right)^{-1} \tag{3}$$

Both the (2) and (3) can be used for any dependent, independent and parallel task.

Method first considers $\mathrm{u}_\mathrm{c}$ as the time required for the execution of task $\mathrm{O}_{\mathrm{kd}}$. In this section the tasks which are not scheduled are tested and monitored just to understand whether the task is free for the execution or not, i.e., it means to check whether the resources required for the execution of the task are available or not. Hence, this results in including a greater number of tasks than the previous methods [19], [25]. Furthermore, the scheduling of the task helps to improve the utilization of the processor. The scheduling of the task can be represented using the equation, i.e., when the task O=1 then:

$$Q^*(O,N) = \sum_{n=1}^{N} ns_n \tag{4}$$

suppose the task O is more than 1 then it is represented using the (5).

$$Q^*(O,N) = \sum_{n=1}^{N} s_n(n + Q^*(O-1, N-n)) + (\sum_{n>N} s_n)Q^*(O-1,N) \tag{5}$$

If the size of the first task is large, then the resources, which are available, are used and all the other tasks are terminated. After this all the other remaining task $0-1$ are scheduled. Further average scheduling after the above situations is $Q^*(O-1,N)$. The proposed model resource utilization efficiency is studied through experiment analysis in next section.

## 4. EXPERIMENT RESULT AND ANALYSIS

This section studies the performance in terms of makespan, energy efficiency and cost efficiency achieved using proposed Amazon web services-transport layer security (AWS-TLS) over existing energy-minimized scheduling of real-time parallel workflows (EMS-RTPW) [19] model. The AWS-TLS and EMS is implemented using Java programing language using cloudsim [2], [3]. The memory intensive nature inspiral is used for validating workload-scheduling models. Makespan, energy, and cost efficiency are metrics used for validating workload-scheduling models.

### 4.1. Makespan performance

Here the makespan efficiency of TLEPA and EMS is measured by varying the inspiral workload task size from 30 (small), 50 (large), and 100 (large) as shown in Figure 4. An average makespan reduction of 80.33% is achieved using TLEPA-WS over EMS-WS. All the observation can be obtained from the graph.

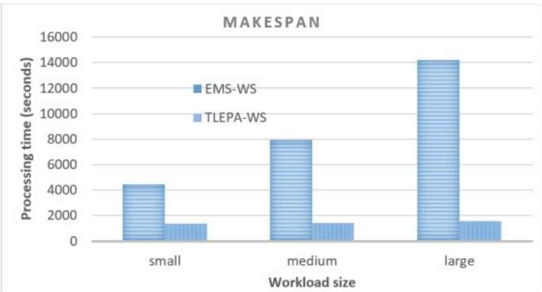


Figure 4. Makespan efficiency with different inspiral workload size

### 4.2. Energy efficiency

Here the energy efficiency of TLEPA and EMS is measured by varying the inspiral workload task size from 30 (small), 50 (large), and 100 (large) as shown in Figure 5. An average energy consumption reduction of 44.96% is achieved using TLEPA-WS over EMS-WS. All the observation can be obtained from the graph.

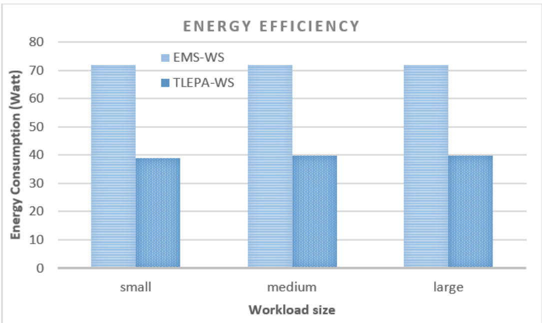


Figure 5. Energy efficiency with different inspiral workload size

### 4.3. Cost efficiency

Here the cost efficiency of TLEPA and EMS is measured by varying the inspiral workload task size from 30 (small), 50 (large), and 100 (large) as shown in Figure 6. An average cost reduction of 80.1% is achieved using TLEPA-WS over EMS-WS. All the observation can be obtained from the graph.

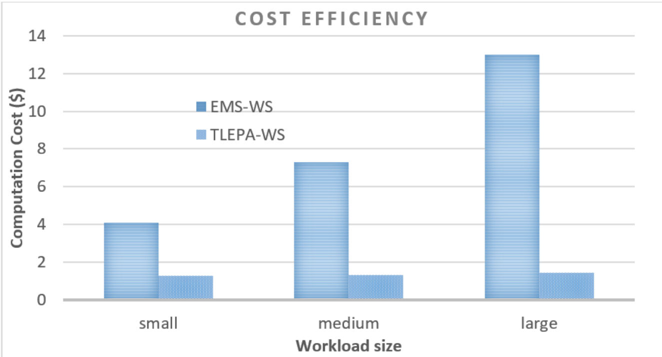


Figure 6. Cost efficiency with different inspiral workload size

## 5. CONCLUSION

This paper presented an efficient WS that assures task level SLA. No prior work has considered scheduling workload considering task-level SLA. The model can reduce energy and maintain high-level of performance; thereby significantly reducing overall execution cost. Experiment outcome shows the TLEPA-WS is very efficient in terms of energy efficiency i.e., an improvement of 44.96% is experienced by TLEPA-WS over EMS-WS. The makespan for execution of workload is reduced by 83.33% by TLEPA-WS over EMS-WS. Similarly, cost for execution of workload is reduced by 80.1% by TLEPA-WS over EMS-WS. The future work would test the model considering different resource intensive workload such as central processing unit (CPU), I/O and memory. Alongside, improve scheduling considering much larger complex task.

## REFERENCES

[1] L. Wu, R. Ding, Z. Jia, and X. Li, "Cost-effective resource provisioning for real-time workflow in cloud," *Complexity*, vol. 2020, pp. 1–15, Mar. 2020, doi: 10.1155/2020/1467274.

[2] L. Chen, X. Li, Y. Guo, and R. Ruiz, "Hybrid resource provisioning for cloud workflows with malleable and rigid tasks," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 1089–1102, Jul. 2021, doi: 10.1109/TCC.2019.2894836.

[3] J. Wang, X. Li, R. Ruiz, J. Yang, and D. Chu, "Energy utilization task scheduling for mapreduce in heterogeneous clusters," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 931–944, Mar. 2022, doi: 10.1109/TSC.2020.2966697.

[4] Z. Wen, R. Qasha, Z. Li, R. Ranjan, P. Watson, and A. Romanovsky, "Dynamically partitioning workflow over federated clouds for optimising the monetary cost and handling run-time failures," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1093–1107, Oct. 2020, doi: 10.1109/TCC.2016.2603477.

[5] 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, 2019, doi: 10.1007/s13369-018-3664-6.

[6] 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, 2019, doi: 10.1016/j.future.2018.10.046.

[7] G. Ismayilov and H. R. Topcuoglu, "Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing," *Future Generation Computer Systems*, vol. 102, pp. 307–322, 2020, doi: 10.1016/j.future.2019.08.012.

[8] A. M. Manasrah and H. B. 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.

[9] S. Yassir, Z. Mostapha, and T. Claude, "Workflow scheduling issues and techniques in cloud computing: a systematic literature review," *CloudTech 2017: Cloud Computing and Big Data: Technologies, Applications and Security*, 2019, pp. 241–263. doi: 10.1007/978-3-319-97719-5_16.

[10] 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," *Journal of Supercomputing*, vol. 76, no. 1, pp. 455–480, 2020, doi: 10.1007/s11227-019-03033-y.

[11] J. K. Konjaang and L. Xu, "Cost optimised heuristic algorithm (COHA) for scientific workflow scheduling in IaaS cloud environment," *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, BigDataSecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, 2020, pp. 162–168, doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00038.

[12] Z. Li, L. Zhang, Y. Cai, and H. Ochiai, "Sensor selection for maneuvering target tracking in wireless sensor networks with uncertainty," *IEEE Sensors Journal*, vol. 22, no. 15, pp. 15071–15081, Aug. 2022, doi: 10.1109/JSEN.2021.3136546.

[13] F. Liu, C. Jiang, and W. Xiao, "Multistep prediction-based adaptive dynamic programming sensor scheduling approach for collaborative target tracking in energy harvesting wireless sensor networks," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 693–704, Apr. 2021, doi: 10.1109/TASE.2020.3019567.

[14] S. R. Jondhale and R. S. Deshpande, "Kalman filtering framework-based real time target tracking in wireless sensor networks using generalized regression neural networks," *IEEE Sensors Journal*, vol. 19, no. 1, pp. 224–233, Jan. 2019, doi: 10.1109/JSEN.2018.2873357.

[15] Y. Zhang and L. Gao, "Sensor-networked underwater target tracking based on grubbs criterion and improved particle filter algorithm," *IEEE Access*, vol. 7, pp. 142894–142906, 2019, doi: 10.1109/ACCESS.2019.2943916.

[16] Y. Wang, H. Liu, J. Zhou, and Y. Wan, "Distributed target tracking in sensor networks by consistency algorithm and semantic moving computing of internet of things," *IEEE Access*, vol. 10, pp. 51634–51642, 2022, doi: 10.1109/ACCESS.2022.3170477.

[17] N. D. Zade, S. N. Deshpande, and D. Sita, "Investigative analysis of suboptimal filter for state estimation in object tracking wireless sensor network," *IEEE Sensors Letters*, vol. 4, no. 10, pp. 1–4, Oct. 2020, doi: 10.1109/LSENS.2020.3026263.

[18] B. N. B. Reddy, B. Pardhasaradhi, G. Srinath, and P. Srihari, "Distributed fusion of optimally quantized local tracker estimates for underwater wireless sensor networks," *IEEE Access*, vol. 10, pp. 38982–38998, 2022, doi: 10.1109/ACCESS.2022.3164515.

[19] 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.

[20] 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.

[21] 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, pp. 1–12, Nov. 2019, doi: 10.1016/j.sysarc.2019.08.004.

[22] 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.

[23] 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.

[24] N. Malik, M. Sardaraz, M. Tahir, B. Shah, G. Ali, and F. Moreira, "Energy-efficient load balancing algorithm for workflow scheduling in cloud data centers using queuing and thresholds," *Applied Sciences*, vol. 11, no. 13, pp. 1–17, Jun. 2021, doi: 10.3390/app11135849.

[25] 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.


## BIOGRAPHIES OF AUTHORS

**Jagadevi Bakka** working as assistant professor in Department of Computer Science and Engineering, East Point College of Engineering and Technology, Bangalore. Her teaching experience is 18 years and her area of interest is cloud computing, big data analytics, data mining, software engineering, and data structures. She can be contacted at email: jagdevi198@rediffmail.com or jagadevi.bakka@gmail.com.

**Dr. Sanjeev C. Lingareddy** received his Ph.D. in the year of 2012 from JNTU, Hyderabad and currently working as professor and Head for the Department of Computer Science and Engineering at Sri Venkateshwara College of Engineering, Bengaluru. He has 24 years of rich experience in the Academics and 7 years of research experience. He has published more than 25 research articles in International Journals. He is a Member of Indian Society for Technical Education (MISTE) and an active member in many technical events. His research area includes Wireless sensor network, wireless security, cloud computing and cognitive network. He can be contacted at this email: sclingareddy@gmail.com.