Location Aware DFS Scheduling Based Improved Quality of Service Maximization with IoT Devices in Cloud

G. John Samuel Babu[1] and M. Baskar[2*]

[1]Research Scholar, Department of Computer Science and Engineering, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Tamil Nadu, India. gb5944@srmist.edu.in, Orcid:https://orcid.org/0009-0000-7414-5673

[2*]Associate Professor, Department of Computing Technologies, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu,Tamil Nadu, India. baashkarcse@gmail.com, Orcid: https://orcid.org/0000-0003-4655-7667

Received: December 06, 2022; Accepted: January 12, 2023; Published: March 30, 2023

Abstract

There are several techniques to maximize the Quality of Service (QoS) of task scheduling in a cloud environment. The approaches can be classified according to the feature, metrics, and methods used in scheduling. For example, performance-based resource selection or service selection identifies the service per its throughput performance some of the ways involved in scheduling according to the makespan time, and idle time. Further, various metrics like resource availability, reputation, and popularity are used in scheduling tasks in a cloud environment. The above methods introduce poor service selection and scheduling performance because of missing the constraints like data fetch performance, data production performance, and performance of other devices. By considering all these, an efficient Location Aware DFS Scheduling Based QoS Maximization (LADFS-QM) algorithm is presented in this article. The method starts with preparing the scheduling data set for the environment by applying Resource Level Normalization algorithm, which involves removing noisy records and preparing the data accordingly. Second, Frequent Availability Optimizer (FAO) is used in resource selection, which computes Frequency and Availability Trust (FAT) value towards feature selection. Third, the method applies DFS scheduling in scheduling the cloud task, which estimates Data Fetch Support (DFS) measure, and ranks the services according to Data Contribution Support (DCS), which is being measured based on the number of IoT devices in route, number of reliable transmissions, number of retransmission and success rate. Finally, route selection is performed according to TFS (Trusted Forwarding Score), measured based on the IoT devices and their support on transmission, latency, congestion, etc. The proposed LADFS-QM algorithm improves the scheduling performance and the environment's quality of service.

Keywords: Location Service, Scheduling, Smart Devices, Cloud Computing, LADFS-QM, QoS Maximization**.**

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 14, number: 1 (March), pp. 37-49. DOI: *10.58346/JOWUA.2023.I1.003*

*Corresponding author: Associate Professor, Department of Computing Technologies, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu,Tamil Nadu, India.

1 Introduction

The development of communication technologies helps various sectors in achieving higher performance. In this way, the cloud computing paradigm has been used by various organizations. Most organizations provide several services for their users to support accessing different resources and data. Such services are location independent and allow their users to access the services independently of their location. Upon receiving the request from the user, the service provider accesses the service data and returns the result for the user. Cloud is a service-oriented environment that allows organizations to maintain their data and resources to be accessed through the services provided. The service providers offer several services to perform the task, and different providers provide different services to achieve the same task. In a cloud environment, several users will generate service requests towards the environment, which must be scheduled over the available services to perform the task.

A task in the cloud may be a set of execution statements containing instructions to access various resources like data servers, memory parts, processors, and so on. Such tasks given behind the service request must be scheduled over the cloud. The cloud is an environment that contains several references for various resources towards resources published by organizations. Scheduling is the process of allotting the resource required for the task identified. Scheduling the task over a cloud environment, several scheduling approaches are available. The priority-based scheduling algorithms consider the task's priority in scheduling the task, whereas makespan-based scheduling schemes consider the completion time of the task. Resource utilization is the other way of scheduling strategy, which would compute the utilization of various resources by any service in scheduling the task. Latency-based approaches monitor the waiting time of the resource in scheduling the task. In this way, several approaches exist in scheduling tasks in a cloud environment.

The Quality of service of the cloud environment and services greatly depends on various metrics. The environment's performance is based on the performance of scheduling, and it is necessary to identify the specific service among available services with great fitness to support the task. By identifying the exact service for the request, the service's performance and the environment can be improved. On the other side, the selection of service greatly impacts the performance of the entire environment. It is necessary to consider the amount of valuable data any service produces. By choosing service accordingly, the growth of service can be achieved.

The performance of scheduling has a great impact on IoT devices. The network area would contain a number of devices to support communications. The presence of Internet of Things (IoT) devices must be considered and included for the growth of the environment. The services may be available at any service point at a specific location. In order to provide personalized and customized support for the users, it is necessary to consider the location of the user. The data can be rapidly provided quickly by identifying a set of services around the user location.

The trust of nodes in the data transmission path is another fact to be considered in maximizing the QoS of the environment. The trust of nodes in the route has been measured in many ways according to the behavior of nodes. Also, the service selected and allotted for the user must consider the service's availability at a specific time. By considering all these factors, an efficient Location Aware DFS Scheduling and QoS Maximization model is presented in this article. The working of the proposed model is sketched in detail in this part.

2 Literature Review

The scheduling problem in the cloud has been handled with various techniques in literature, and this section briefs a set of approaches to the problem.

Jing He, 2022 presented a linear equation-based scheduling model M-QoS-OCCSM, which handles dependent tasks and reduces the cost by meeting the user requirement. Yannian Hu, et.al 2020 prsents an intelligent algorithm-based workflow scheduling scheme, which uses ACO, PSO, and GA. LiWeiJia,, et, al, 2021 presents an Improved whale optimization algorithm (IWC), which considers the time and cost of the virtual machines in scheduling the task. RasoulRashidifar, et. al, 2022 presents a detailed review of various scheduling schemes is presented in Gopatoti JohnSamuel Babu,& M.Baskar 2022,which analyzes various metrics of scheduling.

P. Kuppusamy, et. al 2022 presents a Dynamic Opposition Learning based Social Spider Optimization (DOLSSO) scheme, which uses the superiority of tasks in scheduling. Monika Yadav&Atul Mishra, 2023 presents an ordinal optimization scheme, which uses a linear regression technique in scheduling the tasks with the least makespan.

NupurJangu & ZahidRaza, 2022 presents an enhanced Improved Jellyfish Algorithm (IJFA), which considers a variety of cloud and fog resource parameters, including speed, capacity, task size, number of tasks, and number of virtual machines for resource provisioning in a fog integrated cloud environment. Moses Ashawa, et. al, 2022 presents an LSTM-based scheduling model, which analyses the heuristics application resource utilization towards scheduling. Peng Liu, et.al 2022 presents a Multi-AGV Cyclical Offloading Optimization (MCOO) scheduling, which applies a greedy algorithm to finding optimal allocation. Further, the method uses multiple AGVs in a synchronized way with Reinforcement Learning-based A3C algorithm.

Tingting Fu, et.al 2022 presents a optimized memory allocation with Q-learning, which considers the data rate as the key to scheduling the task. Jun Liu & Xi Liu, 2022 presented a polynomial-time approximation scheme for scheduling the task according to the time complexity.

To maximize the cache utilization, RanaGhazali, et.al 2022 presented a reinforcement learning-based model CLQLMRS (Cache Locality with Q-Learning in MapReduce Scheduler.

Yu Zhou, et.al 2022 presented a collaborative task is offloading and resources allocation algorithm (CTORAA), which uses artificial intelligence in task offloading and energy harvesting.

Saravanan Muniswamy & RadhakrishnanVignesh 2022 presented a deep learning-based dynamic, scalable task scheduling (DSTS), which uses the modified multi-swarm coyote optimization (MMCO) method in the selection of service and cluster of the task using modified pigeon-inspired optimization (MPIO). Finally, deep convolutional neural network (DCNN) allows for dynamic priority-based scheduling.

Nweso Emmanuel Nwogbaga et.al 2022 presented a genetic algorithm and particle swarm optimization scheme , which select the services with a genetic algorithm and perform scheduling with particle swarm optimization. The unrelated Parallel Machines Scheduling Problem (UPMSP) algorithm is presented by Xingwang Huang, et.al 2022, which considers the completion time and sequence sequence-dependent setup times.

AbdelazizAbohamama, et.al, 2022 proposed a semi-dynamic real-time task scheduling algorithm, which generates a permutation and applies a genetic algorithm toward scheduling. The method considers the execution time as the factor in scheduling. S. Tuli, et.al 2022 presented a A3C-based scheduling scheme, which generates temporal patterns to optimize scheduling. T. A. L. Genez, et. al, 2019 present the effect of bandwidth in workflow scheduling, which considers the cost, makespan time, and inefficient bandwidth scheduling. M.Ali, et.al,2022 presented a multi-objective task-scheduling optimization named Discrete Non-dominated Sorting Genetic Algorithm II (DNSGA-II), which applies GA in selecting a schedule. A.Marahatta, et.al 2021 presented aenergy-efficient dynamic scheduling scheme (EDS), which uses historical data to identify the resources to support the tasks.

Y. Xiong, et.al 2019 presented a Johnson rule-based genetic algorithm scheduling named Johnson's-rule-based genetic algorithm (JRGA), which considers the characteristics of multiprocessor scheduling in CDCs. S. Nabi, et.al 2021 presented a resource-aware dynamic task scheduling approach, which considers data sets like HCSP, GoCJ, and Synthetic workload. Y. Wang and X. Zuo 2021 presents a particle swarm optimization with idle time slot-aware rules-based scheduling. Q. Wu, et. Al 2022 presents an Endpoint communication contention-aware List Scheduling Heuristic (ELSH) , which uses task ranking property as the key to scheduling the task.

All the above-discussed approaches suffer to achieve higher performance in scheduling.

3 Location Aware DFS Scheduling Based QOS Maximization (LADFS-QM) Model

The proposed location-aware DFS scheduling-based QoS maximization model reads the scheduling trace maintained. First, the traces are preprocessed with Resource Level Normalization Algorithm which removes the noisy records from the set. Second, feature selection is performed by computing different metrics. Third, according to the feature selection, the set of traces is collected and applies DFS scheduling in scheduling the cloud task, which estimates Data Fetch Support (DFS) measure, and ranks the services according to Data Contribution Support (DCS), which is being measured based on number of IoT devices in route, number of reliable transmissions, number of retransmission and success rate. Finally, route selection is performed according to TFS (Trusted Forwarding Score), measured based on the IoT devices and their support for transmission, latency, congestion, etc.

The functional schema of the proposed model is presented in Figure. 1, which receives the user request and schedule according to the DFS algorithm. The detailed work of the model is presented in this section.

3.1. Request Handler

The request handler is the process that continuously monitors the environment and waits for the request. Whenever a request is submitted, the method receives the features of the request and preprocesses the traces. With the preprocessed traces, the method performs DFS scheduling, and finally, the method involves TFS Forwarding.

3.2. FLN Preprocessing

The events of the schedule have been maintained in the scheduling trace. Such scheduling trace has been used as the key for selecting resources or services for any request. First, the method fetches the trace and finds the set of features. With the feature set identified for distinct features, the method computes the Feature Level Normalization Factor (FLNF). The value of FLNF is measured based on the range of values of the feature. To measure the FLNF value, the method finds the minimum, maximum, and median values. Further, with the values measured, the method computes the FLNF value.

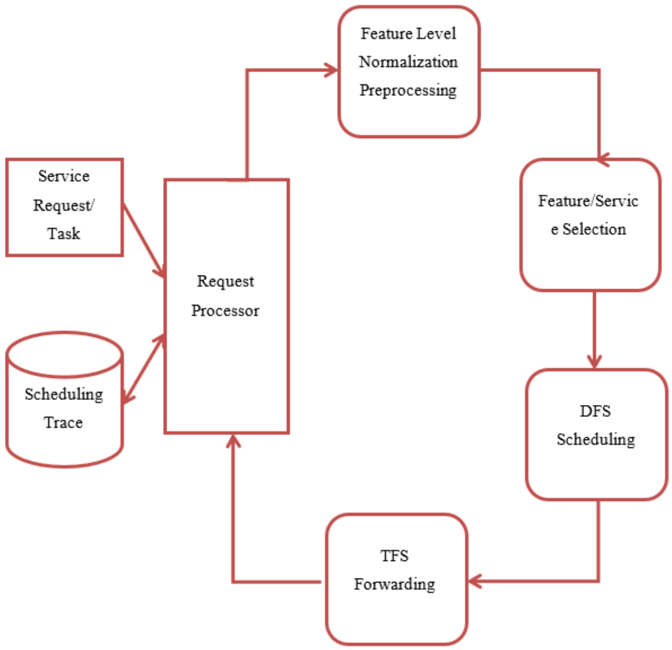

Figure 1: Architecture of Proposed LADFS-QM Model

Further, the method traverses through each tuple and removes the tuples with missing features. Similarly, with the tuples with missing values, the method generates a normalization value according to the FLNF of the feature. Such a normalized data set has been used to perform scheduling.

Algorithm:

Given: Scheduling Trace ST.
Obtain: Preprocessed trace Pst.
Start
Read ST.

Feature Set Fes = $Fes \cup (\sum_{i=1}^{Size(ST)}(Features \in ST) \cap Fes)$

For each feature fi

Compute minimum MinF = $\underset{i=1}{\overset{size(ST)}{Min}}(St(i).fi)$

Compute maximum MaxF = $\underset{i=1}{\overset{size(ST)}{Max}}(St(i).fi)$

Compute Median MeF = $\frac{MaxF}{2}$

Compute FLNF = $\frac{Dist(MaxF-MeF)}{size(ST)}$

End

For each trace Ti

If Ti∋ $\forall Features(fs) then$

Remove the trace

Continue

end
For each feature Fi
If Ti.Fi==null then
Ti(Fi) = Fi.Min+(FLNF×Rand(1,10))
End
End
Add to preprocessed set Pst.
End
Stop

The feature level normalization algorithm estimates the feature level normalization factor (FLNF) value for different features. Accordingly, the traces are identified for noisy records and normalized to the feature values to support effective scheduling.

3.3. Feature-Service Selection

The cloud environment would contain a number of services available to handle the task. It is necessary to identify the subset of services or resources. To perform this, the method first identifies the set of services according to the request given. Second, the method applies Frequent Availability Optimizer (FAO) is applied in resource selection, which computes Frequency and Availability Trust (FAT) value towards feature selection. The method computes the Frequency factor and availability trust (FAT) for each service available. Based on the value of FAT, the method identifies a subset of services to perform scheduling.

Algorithm:

Given: Preprocessed trace Pst, Service Request Sreq, Service Taxonomy SeT.
Obtain: Service Set Ses.
Start
Read Pst, SreqSeT.
Service Set Ss = Ss∪ (Ss ∩ $(\sum_{i=1}^{size(Pst)} Pst(i).Service.Type == SeT.Service.Type))$
For each service s
Compute Frequency factor FrF = $\frac{(\sum_{i=1}^{size(Pst)} Pst(i).Service==s)}{(\sum_{i=1}^{size(Pst)} Pst(i).Service.Type==s.Type)}$
Compute Availability Fact Af = $\frac{(\sum_{i=1}^{size(Pst)} Pst(i).Service==s\&\&Pst(i).state==Available)}{(\sum_{i=1}^{size(Pst)} Pst(i).Service==s)}$
Compute FAT = $\frac{Af}{FrF} \times 100$
If FAT>Th, then
SeS = $(\sum Services \in Ses) \cup s$
End
End
Stop

The feature and service selection algorithm computes various services' frequency and availability factors. Based on the value of frequency and availability factors, the method computes the value of FAT to support service selection.

3.4. DFS Scheduling and TFS Forwarding

The proposed location-aware DFS scheduling algorithm works according to the service characteristics like Data contribution and data fetch performance. To start with, the set of services according to the user location is identified. For the services selected, the method performs a features-service selection algorithm which returns a set of services. For the services identified, the method computes Data Fetch Support (DFS) measure and Data Contribution Support (DCS) to measure the Task Fitness Score (TFiS). The value of Data Contribution Support (DCS) is measured according to the previous history. Once a single service is identified based on the TFIS score, then a set of routes are identified to reach the user. Finally, route selection is performed according to TFS (Trusted Forwarding Score), which is measured based on the number of IoT devices on the route, the number of reliable transmissions, the number of retransmissions, and the success rate.

Algorithm:

Given: Services Identified SI, Service Trace STs, Service Request Sreq.
Obtain: Null.
Start
Read SI, STs, Sreq.
Pst = Preprocessing (STs)
Service set Ss = Feature_Service_Selection (Pst)
For each service s

Compute DFS = $\frac{\sum_{i=1}^{SIZE(Pst)} Pst(i).datalength}{Size(Pst)} \times \frac{\sum_{i=1}^{Size(Pst)} Count(Pst(i).NoofRead)}{size(Pst)}$

Compute DCS = $\frac{\sum_{i=1}^{SIZE(Pst)} size(Pst(i).result)}{Size(Pst)} \times \frac{\sum_{i=1}^{Size(Pst)} Count(Pst(i).State==Ok)}{size(Pst)}$

Compute Trusted Fitness Score TFIS = DFS$\times DCS$

End
Service S = Find the service with the maximum TFIS value.
Route List Rls = Find the routes to reach the service point.
For each route r

Compute TFS = $\frac{NoRT}{NoT} \times \frac{NoID}{NoT} \times \frac{NoST}{NoT}$

// where NORT – No of retransmission, NoT-No of transmission, NOID – No of IoT devices, NoST- number of successful transmissions.

End
Route R = select a route with maximum TFS and transmit the data.
Stop

The working of DFS scheduling and TFS forwarding has been presented in the above pseudo-code, which computes TFIS value to identify the service where the method computes TFS to find the optimal route to perform the transmission.

4 Results and Discussion

The proposed location-aware DFS scheduling towards the quality of service maximization has been implemented and its performance is evaluated in the presence of a different number of services in the cloud. The method has been validated for its scheduling, resource utilization, and time complexity performance. The obtained results have been compared with the results of other methods.

Table 1: Details of Evaluation

Parameter	**Value**
Tool Used	Cloud Sim
Number of Resources	300
Number of instances	10
Number of tasks	500
Number of Services	200

The details of the simulation environment used for the performance evaluation of the proposed model are presented in Table 1, where the method is measured for its performance in various factors and compared with other approaches. The methods are evaluated for their performance according to the data maintained by Microsoft azure cloud environment. The traecs of service access maintained by the Azure environment for the period of 6 months is collected and used for evaluation.

Table 2: Analysis of Scheduling

Scheduling Performance % vs. No of Tasks			
	100 Tasks	300 Tasks	500 Tasks
M-QoS-OCCSM	75	79	83
IWC	77	84	87
DOLSSO	84	89	93
MMCO	88	93	97
LADFS-QM	91	95	99

The performance of methods in scheduling the tasks has been measured in the presence of different tasks in the queue. The results produced by the methods are compared in Table 2, where the proposed LADFS-DM scheme has produced high scheduling performance than other methods. The methods performance is measured for the data set considered and their performance are obtained from concern article and compared with the result of our LADFS-DM model.

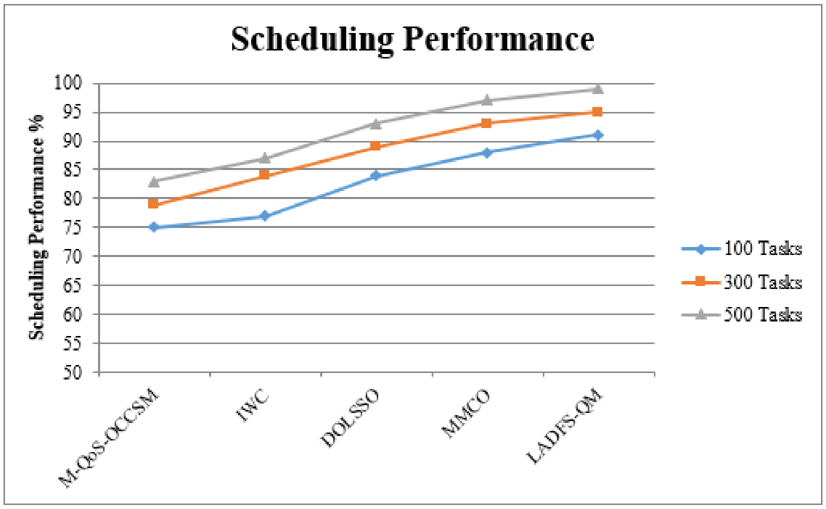

Figure 2: Analysis of Scheduling Performance

Analysis of scheduling performance is pictured in Figure 2, which denotes the proposed LADFS-QM algorithm has produced higher scheduling performance than other methods.

Table 3: Analysis of Energy Efficiency

Energy Efficiency % vs. No of Tasks			
	100 Tasks	300 Tasks	500 Tasks
M-QoS-OCCSM	76	72	69
IWC	82	75	73
DOLSSO	85	88	91
MMCO	89	92	96
LADFS-QM	92	96	98

The energy efficiency of various methods in scheduling has been presented in Table 3, which shows the proposed LADFS-QM method achieves higher energy efficiency than other approaches.

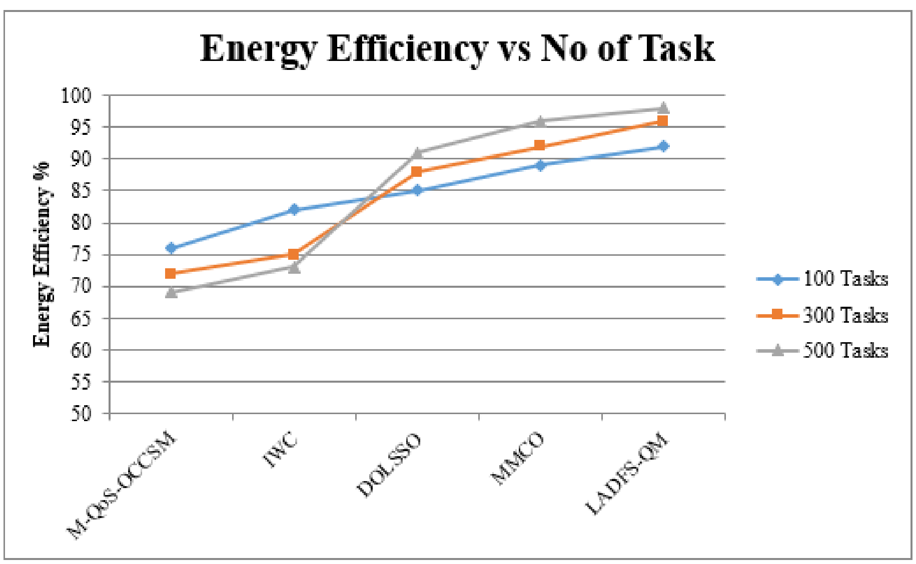

Figure 3: Analysis of Energy Efficiency

The efficiency of energy in scheduling has been measured for various approaches. The result of energy efficiency is pictured in Figure 3, and the LADFS-QM scheme achieves higher energy efficiency than others.

Table 4: Analysis of Resource Utilization

Resource Utilization Performance vs. No of Task			
	100 Tasks	300 Tasks	500 Tasks
M-QoS-OCCSM	71	74	78
IWC	74	79	83
DOLSSO	81	87	92
MMCO	87	92	97
LADFS-QM	89	94	99

The performance of methods in utilizing the resources is measured and presented in Table 4, where the proposed LADFS-QM has produced higher performance in all test cases.

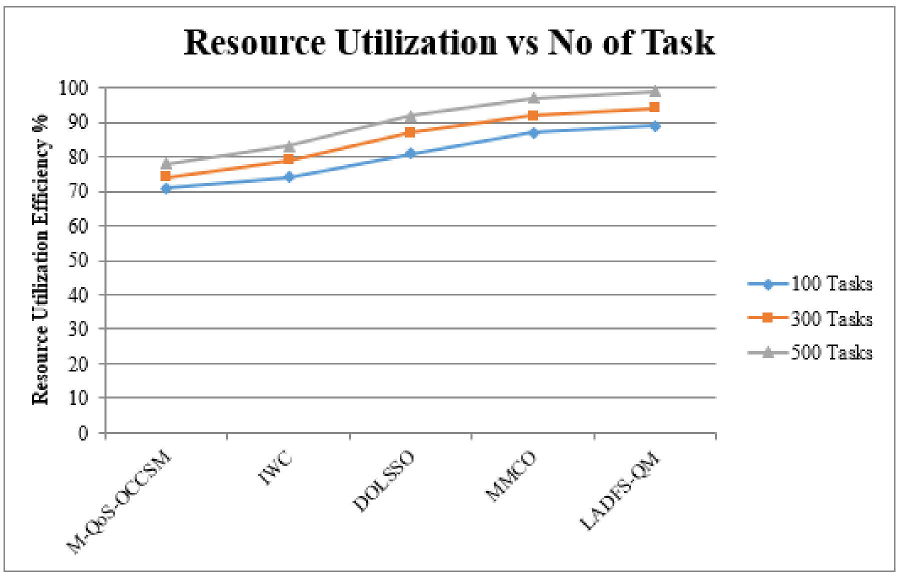

Figure 4: Analysis of Resource Utilization

The performance in resource utilization has been measured and presented in Figure 4, where the proposed LADFS-QM algorithm has produced higher resource utilization performance than other methods.

Table 5: Performance in Time Complexity

Time Complexity vs. No of Task			
	100 Tasks	300 Tasks	500 Tasks
M-QoS-OCCSM	33	45	58
IWC	29	36	47
DOLSSO	23	29	33
MMCO	20	24	28
LADFS-QM	17	20	25

The performance of methods in time complexity is measured and presented in Table 5. The LADFS-DM algorithm produces less time complexity than others.

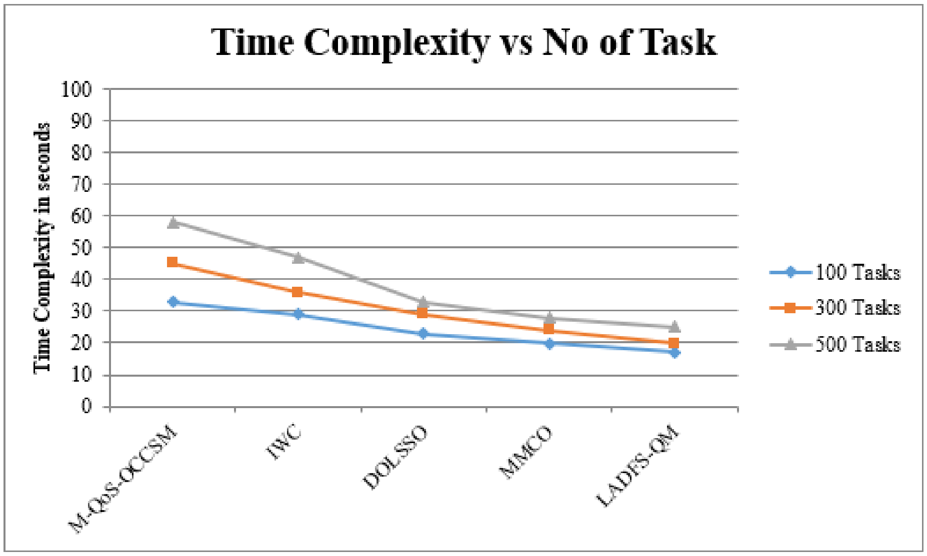

Figure 5: Analysis of Time Complexity

The value of time complexity introduced by various approaches is measured and compared in Figure 5. The LADFS-QM algorithm produces less time complexity.

5 Conclusion

This article presented a novel location-aware DFS scheduling algorithm for QoS maximization of the cloud environment. The method performs preprocessing of scheduling traces and performs feature selection to identify a subset of suitable services. With the services identified, the method applies DRS scheduling by computing the Trusted Fitness Score (TFIS) and selecting a route according to the Trusted Forwarding score (TFS). The proposed method improves the performance of scheduling and reduces energy consumption. The proposed method consider data fetch support values of services in scheduling which is suitable for the services whose traces are available and they choose the services based on their previous selection and performance. However, the services which are selected for minimum time would never get selected and it would affect the performance of the entire service. This increases the requirement of selecting a service based on other metrics and the service selection algorithm must identify services based on other metrics like minimum bound which identifies the service according to other metrics like availability metrics.

References

[1] Abohamama, A.S., El-Ghamry, A., & Hamouda, E. (2022). Real-time task scheduling algorithm for iot-based applications in the cloud–fog environment. *Journal of Network and Systems Management*, *30*(4), 1-35.

[2] Ali, I.M., Sallam, K.M., Moustafa, N., Chakraborty, R., Ryan, M., & Choo, K.K.R. (2020). An automated task scheduling model using non-dominated sorting genetic algorithm ii for fog-cloud systems. *IEEE Transactions on Cloud Computing*, *10*(4), 2294-2308.

[3] Ashawa, M., Douglas, O., Osamor, J., & Jackie, R. (2022). Improving cloud efficiency through optimized resource allocation technique for load balancing using LSTM machine learning algorithm. *Journal of Cloud Computing*, *11*(1), 1-17.

[4] Babu, G.J.S., & Baskar, M. (2022). Application of blockchain methodology in secure task scheduling in cloud environment. *Advances in Engineering Software*, *172*, 1-8.

[5] Fu, T., Peng, Y., Liu, P., Lao, H., & Wan, S. (2022). Distributed reinforcement learning-based memory allocation for edge-PLCs in industrial IoT. *Journal of Cloud Computing*, *11*(1), 73.

[6] Genez, T.A., Bittencourt, L.F., da Fonseca, N.L., & Madeira, E.R. (2015). Estimation of the available bandwidth in inter-cloud links for task scheduling in hybrid clouds. *IEEE Transactions on Cloud Computing*, *7*(1), 62-74.

[7] Ghazali, R., Adabi, S., Rezaee, A., Down, D.G., & Movaghar, A. (2022). CLQLMRS: improving cache locality in MapReduce job scheduling using Q-learning. *Journal of Cloud Computing*, *11*(1), 1-17.

[8] He, J. (2022). Decision Scheduling for Cloud Computing Tasks Relying on Solving Large Linear Systems of Equations. *Computational Intelligence and Neuroscience*, *2022*.

[9] Hu, Y., Wang, H., & Ma, W. (2020). Intelligent cloud workflow management and scheduling method for big data applications. *Journal of Cloud Computing*, *9*, 1-13.

[10] Huang, X., Chen, L., Zhang, Y., Su, S., Lin, Y., & Cao, X. (2022). Improved firefly algorithm with courtship learning for unrelated parallel machine scheduling problem with sequence-dependent setup times. *Journal of Cloud Computing*, *11*(1), 1-17.

[11] Jangu, N., & Raza, Z. (2022). Improved Jellyfish Algorithm-based multi-aspect task scheduling model for IoT tasks over fog integrated cloud environment. *Journal of Cloud Computing*, *11*(1), 1-21.

[12] Jia, L., Li, K., & Shi, X. (2021). Cloud computing task scheduling model based on improved whale optimization algorithm. *Wireless Communications and Mobile Computing*, *2021*, 1-13.

[13] John Samuel Babu, G., & Baskar, M. (2023). TRS scheduling for improved QoS performance in cloud system. *Computers, Materials & Continua*, *75*(1), 1547–1559.

[14] Kuppusamy, P., Kumari, N.M.J., Alghamdi, W.Y., Alyami, H., Ramalingam, R., Javed, A.R., & Rashid, M. (2022). Job scheduling problem in fog-cloud-based environment using reinforced social spider optimization. *Journal of Cloud Computing*, *11*(1), 1-14.

[15] Lemay, D.J., Basnet, R.B., & Doleck, T. (2020). Examining the Relationship between Threat and Coping Appraisal in Phishing Detection among College Students. *Journal of Internet Services and Information Security*, *10*(1), 38-49.

[16] Liu, J., & Liu, X. (2022). Energy-efficient allocation for multiple tasks in mobile edge computing. *Journal of Cloud Computing*, *11*(1), 1-14.

[17] Liu, P., Liu, Z., Wang, J., Wu, Z., Li, P., & Lu, H. (2022). Reinforcement learning empowered multi-AGV offloading scheduling in edge-cloud IIoT. *Journal of Cloud Computing*, *11*(1), 1-14.

[18] Marahatta, A., Pirbhulal, S., Zhang, F., Parizi, R.M., Choo, K.K.R., & Liu, Z. (2019). Classification-based and energy-efficient dynamic task scheduling scheme for virtualized cloud data center. *IEEE Transactions on Cloud Computing*, *9*(4), 1376-1390.

[19] Muniswamy, S., & Vignesh, R. (2022). DSTS: A hybrid optimal and deep learning for dynamic scalable task scheduling on container cloud environment. *Journal of Cloud Computing*, *11*(1), 1-19.

[20] Nabi, S., Ibrahim, M., & Jimenez, J. M. (2021). DRALBA: Dynamic and resource aware load balanced scheduling approach for cloud computing. *IEEE Access*, *9*, 61283-61297.

[21] Nwogbaga, N. E., Latip, R., Affendey, L. S., & Rahiman, A. R. A. (2022). Attribute reduction based scheduling algorithm with enhanced hybrid genetic algorithm and particle swarm optimization for optimal device selection. *Journal of Cloud Computing*, *11*(1), 1-17.

[22] Ramkumar, J., Baskar, M., Viswak, M., & Ashish, M.D. (2020). Smart shopping with integrated secure system based on IoT. *International Journal of Advanced Science and Technology*, *29*(5), 301-312.

[23] Rashidifar, R., Bouzary, H., & Chen, F.F. (2022). Resource scheduling in cloud-based manufacturing system: a comprehensive survey. *The International Journal of Advanced Manufacturing Technology*, *122*(11-12), 4201-4219.

[24] Tuli, S., Ilager, S., Ramamohanarao, K., & Buyya, R. (2020). Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks. *IEEE transactions on mobile computing*, *21*(3), 940-954.

[25] Wang, Y., & Zuo, X. (2021). An effective cloud workflow scheduling approach combining PSO and idle time slot-aware rules. *IEEE/CAA journal of automatica sinica*, *8*(5), 1079-1094.

[26] Wu, Q., Zhou, M., & Wen, J. (2021). Endpoint communication contention-aware cloud workflow scheduling. *IEEE Transactions on Automation Science and Engineering*, *19*(2), 1137-1150.

[27] Xiong, Y., Huang, S., Wu, M., She, J., & Jiang, K. (2017). A Johnson's-rule-based genetic algorithm for two-stage-task scheduling problem in data-centers of cloud computing. *IEEE Transactions on Cloud Computing*, *7*(3), 597-610.

[28] Yadav, M., & Mishra, A. (2023). An enhanced ordinal optimization with lower scheduling overhead based novel approach for task scheduling in cloud computing environment. *Journal of Cloud Computing*, *12*(1), 1-14.

[29] Zhou, Y., Ge, H., Ma, B., Zhang, S., & Huang, J. (2022). Collaborative task offloading and resource allocation with hybrid energy supply for UAV-assisted multi-clouds. *Journal of Cloud Computing*, *11*(1), 42.

Authors Biography

G.John Samuuel Babu is a Research Scholar in Department of Computer Science and Engineering, School of Computing, Kattankulathur Campus, SRM Institute of Science and Technology .He has completed his B.E. from Gulbarga University and M.Tech. from JNTU, Hyderabad. His area of research is Cloud Computing, Networking and Data Structures.

ORCID:https://orcid.org/0009-0000-7414-5673

Dr. M. Baskar is a Researcher cum Associate Professor in Department of Computing Technologies, School of Computing, Kattankulathur Campus, SRM Institute of Science and Technology. He received B.E. Computer Science and Engineering from Anna University, Chennai, M.Tech. Information Technology from Sathyabama University, Chennai and Ph.D.(Information and Communication Engineering) from Anna University, Chennai. His Area of research interest includes Computer Networks and Security, Parallel and Distributed Systems, Image Processing, Big Data, Machine Learning and IoT. He has published 58 Research Article in reputed International Journals and 21 Article in International Conferences. He has published 8 patents. He is acting as a reviewer in Cluster Computing, Journal of Web Engineering, Multimedia Tools and Applications, Neural Processing Letters and Concurrency and Computation: Practice and Experience. He is a Life time Professional body member of ACM,CSI, ISTE, IET, ISRD, IRED, IACSET, IAENG, SDIWC and UACEE. **Orcid:** https://orcid.org/0000-0003-4655-7667.