Application Migrations in Edge Computing Using Real Time Workload Ranking Algorithm

Saravanan Velrajan and Thangakumar Jeyaprakash
Department of Computer Science, Hindustan Institute of Technology and Science Chennai, India

Keywords: 5G Networks, Edge Computing, Application Performance Management, Workload Migrations.

Abstract: Service Providers host a multitude of diverse applications in edge clouds in 5G networks for improving the user experience. The demand for automation of workflows in different industries has created the need for 5G-based private wireless networks with edge infrastructure. The varied nature of the workloads hosted at the edge clouds, lead to increased focus on service quality and performance. Application workload migrations are done in edge infrastructure to improve the SLA adherence of services and to improve user experience. This research proposes a new adaptive ranking approach for application workload migrations in edge computing. The proposed Adaptive Workload Ranking Algorithm (AWRA) predicts the available time for workload migrations and triggers migration of application workloads. The selection of workloads to be migrated is made by balancing the resource utilisation of application instances and the estimated time taken for migration. Our experiments demonstrate that AWRA effectively manages overload conditions of varied applications hosted in edge infrastructure. AWRA outperformed state-of-the art algorithms such as ASMWA and EWMA3 in minimising SLA violations during workload migrations.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The growth of 5G networks has created a paradigm shift in wireless communication capabilities. Characterised by a significant bandwidth surge and a dramatic latency reduction, 5G has unlocked the potential for applications such as online gaming, live video streaming, holograms and smart communities. Wireless service providers deploy edge cloud at the network edges to guarantee QoS for applications accessed by mobile subscribers, as given in Figure 1 (Velrajan and Sharmila, 2023).

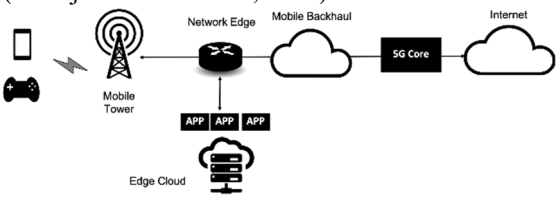


Figure 1: Edge Cloud Deployment at Service Provider Edge

Edge computing provides new monetization opportunities for service providers. However, the diverse applications hosted at the edge servers in 5G networks have strict Quality of Service (QoS) needs (Velrajan and Sharmila, 2023). The diverse applications deployed in the edge have varying QoS characteristics. For example, a video streaming application would expect the 5G network infrastructure to support high bandwidth; an autonomous car application would need support for lower latencies; and an Industry 4.0 deployment would require higher reliability from the network. Managing the QoS of users at the edge is complex because of the distributed deployment of the edge infrastructure and the dynamic nature of the workload.

Edge infrastructure continuously monitors the availability and performance of applications running on the edge servers (Velrajan and Sharmila, 2023). When an application instance in edge server becomes overloaded, it impacts the Service Level Agreements (SLAs) of users. Thus, wireless service providers perform workload migrations in edge to prevent applications from becoming overloaded (Velrajan and Sharmila, 2023). However, migrating a workload from one edge host to another often results in disruption of services to users, impacting their quality of experience (Velrajan and Sharmila, 2023). Existing research primarily focuses on workload migrations in centralised cloud environments. A new

Velrajan, S. and Jeyaprakash, T.
Application Migrations in Edge Computing Using Real-Time Workload Ranking Algorithm.
DOI: 10.5220/0013591600004664
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 2*, pages 320-327
ISBN: 978-989-758-763-4
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

approach is necessary to handle the challenges created by dynamic workloads, diverse applications, and the distributed deployment architecture of edge computing in 5G networks.

Our research focuses on workload migrations to manage application overload scenarios in edge infrastructure in 5G networks. We adopt a proactive approach to initiate workload migrations that minimise the impact on QoS experienced by users.

This work introduces two key advancements to enable QoS-aware migration of workloads in edge clouds:

1. Adaptive Workload Ranking Algorithm (AWRA) for proactive workload migrations within edge clouds.
2. We evaluate the performance of AWRA against other workload migration methods through a comprehensive analysis.

AWRA ranks workloads to migrate, considering the user sessions handled and the migration time needed. Furthermore, AWRA uses a variable application performance limit to initiate workload migrations. Our simulations reveal that AWRA outperforms existing approaches in minimising QoS impact during workload migrations.

2 LITERATURE REVIEW

Ensuring optimal application performance in edge cloud necessitates efficient workload migration strategies. This review focuses on recent workload migration advancements, specifically geared towards resource optimisation, user mobility, energy savings and application performance management.

2.1 Optimisation of system resources

Optimising system resource allocation is important for delivering high-quality workloads to users in edge clouds running resource hungry applications like gaming and live video delivery. Several existing studies highlight the importance of workload migration in a timely fashion to optimise resource utilisation and ensure workload continuity for users.

The experiments done by Chen et al. focus on continuity of user workloads during migrations while optimising resource utilisation. The authors aim to perform migrations without impacting the service quality for users (Chen et al., 2024). Through their simulation, the authors show that initiating workload migrations during appropriate time and identifying the best node for migrating workloads are important for achieving outstanding service quality.

The studies done by Wan et al. use federated mechanisms to assign resources to user services, with a goal of lowering energy consumption and minimising the impact to service quality (Wan, 2024). The authors use resource-based and task-based methods to optimise compute and power resources in an edge computing infrastructure deployed in IoT networks. The authors claim that the proposed solution improves the performance of edge infrastructure and lowers power consumption, in addition to optimising resource management.

2.2 Management of user mobility

Quite often user services get migrated in edge clouds due to the mobility of subscribers, as their sessions get relocated across the different mobile towers. Autonomous vehicles connecting to 5G networks want uninterrupted and low-latency connectivity to the workloads hosted in the edge clouds.

Cao et al. introduce a dynamic path selection algorithm to manage workload migrations during subscriber mobility scenarios (Cao et al., 2024). The authors claim to reduce the impact of migrations on user session latency, and at the same time, operating within the constraints of computing cost.

Shokouhi et al. present a hierarchical architecture based framework for user tasks offloading in edge clouds during mobility situations (Shokouhi et al., 2024). The framework introduced by the authors uses a predictive approach to decide the appropriate edge server to deliver service to the user. The authors claim that the presented approach is best suited to reduce the cost of offloading user tasks during mobility scenarios.

2.3 Optimise energy consumption

Zolfaghari adopts a predictive mechanism to migrate workloads in cloud data centers, to achieve energy and performance tradeoff (Zolfaghari, 2024). The proposed VM selection method for migrations in the cloud, is claimed to achieve energy savings, complying with SLAs to the best extent possible.

Ouhame et al. use self-optimising machine learning approach to migrate VMs in a cloud environment with workloads that are dynamic in nature (Ouhame et al., 2024). The proposed algorithm uses an intelligent learning approach that disables VM nodes in cloud that are not in use. The proposed solution claims to achieve quicker workload migration decisions, improved SLA adherence and reduced energy consumption in datacenter clouds. Machine learning based intelligent techniques are

used often in cloud to optimise energy consumption, as they directly reflect on operational cost savings. However, it is equally important to honour the SLAs for user sessions and avoid delays.

2.4 VM migrations in centralised cloud

Workload migrations in the cloud has been gaining attention in the recent times with researchers exploring techniques such as migration of workloads using VMs or containers. However, the distributed nature of edge infrastructure prevents the seamless extension of cloud workload migration techniques without compromising on the user's service quality (Velrajan and Sharmila, 2023).

Kulshrestha et al. use a method based on Exponential Weighted Moving Average (EWMA3) to migrate VMs hosted in the centralised datacenters (Kulshrestha and Patel, 2021). The authors compare the behaviour of EWMA3 with other popular VM selection algorithms and demonstrate that EWMA3 reduces SLA violations in the cloud during workload migrations.

A proactive method is introduced by Patil et al. for overload detection in the cloud to migrate VMs, especially during burst load conditions (Patil and Patil, 2023). The authors claim that the proposed algorithm efficiently manages recurrent VM migrations and instant overload conditions in the cloud environment.

The studies on VM migration in cloud are mostly restricted to host overload management and energy optimisation and do not consider application-specific characteristics during migration. Also, the VM migration techniques adopted in the centralised cloud environments are not extensible for the distributed edge cloud architectures.

2.5 Application Performance Management

Velrajan et al. have introduced a distributed mechanism to migrate workloads at the edge minimising the impact on user's QoS (Velrajan and Sharmila, 2023). The proposed approach decides the best application performance limit for initiating workload migrations using an enhanced Particle Swarm Optimization (PSO) algorithm. This approach is suitable in environments where latency-sensitive workload migrations must be performed.

Velrajan et al. have proposed a dynamic migration approach in edge computing using a continuously adapting workload migration window (Velrajan and Sharmila, 2023). The proposed Adaptive Service Migration Window Algorithm (ASWMA) significantly minimises SLA violations in a QoS-sensitive environment while performing workload migrations. However, it doesn't differentiate workloads during migration, impacting critical workloads that handle user sessions.

While significant progress has been made in workload migrations in edge cloud environments, there's still room for improvement, especially in studying the behaviour of workload migrations in QoS-constrained environments. Existing workload migration approaches do not consider the time taken for workload migrations. However, in the real world, there is a need for time-bound proactive workload migrations for improved application performance management.

Furthermore, existing approaches do not differentiate the workloads of an application during workload migrations. For example, a user facing workload that streams video to users may have a higher rank than the workload that generates dashboards in a video streaming server. Thus, a migration approach that ranks workloads would help improve QoS and reduce SLA violations in QoS-sensitive applications such as on-demand or live video streaming, smart surveillance and AR/VR.

3 ADAPTIVE WORKLOAD MIGRATIONS

This work introduces an Adaptive Workload Ranking Algorithm (AWRA) to address the growing need for efficient and QoS-aware workload migrations in edge clouds. The algorithm uses the application load, number of services handled by the application and workload migration time to decide which workloads to prioritise during migrations. AWRA's proactive approach based on workload priority minimises users' QoS impact during workload migration. Table 1 captures the abbreviations and symbols used in this paper.

Table 1: Abbreviations and Symbols

Parameter	Description
appPerfLimit	Application performance limit at which workload migrations are initiated
APS	Application Performance Score representing application's overall resource utilisation and user's QoE
$APS_{initial}$	Application Performance Score before initiating workload migration

Parameter	Description
APS_{final}	Application Performance Score after completing workload migration
APS_{delta}	Adjustment done to the application performance limit for migrations
RU	Application's Aggregated Resource utilisation
QoE	Quality of Experience for the users
$MT_{forecast}$	Predicted Migration Time for Workloads
MT_{actual}	Actual Time taken to migrate workloads
$Wklds_n$	Workloads list with the user sessions count for an application instance
$Wklds_p$	Ranked workload list for an application instance
slaViolations	SLA Violations during the migration of workloads

Figure. 2 visually depicts the critical steps involved in workload migrations based on real-time application performance and user experience metrics using the AWRA algorithm.

AWRA monitors the performance of applications running on the edge hosts using Application Performance Score (APS) (Velrajan and Sharmila, 2023). Unlike static performance limits configured in a centralised cloud environment (VMWare, 2024), the APS metric considers overall system resource utilisation and user-experienced QoS for each application instance. Thus, APS uniquely represents the characteristics of diverse applications hosted in edge cloud.

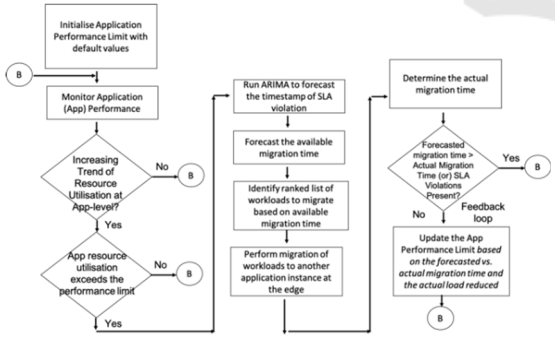


Figure 2: Flowchart for AWRA Algorithm.

The calculation of APS is detailed in equation (1). Resource utilisation of an application represented by *RU(app_id)* aggregates the utilisation of system resources including CPU, disk storage, network ports and RAM; and *QoE* is the users' quality of experience from the workload (Velrajan and Sharmila, 2023).

$$APS\ (app\_id) = RU(app\_id) + (QoE/RU(app\_id)) \quad (1)$$

At the beginning of the execution, AWRA initialises the application performance limit (*appPerfLimit*) to trigger migrations to DEFAULT_LOAD_LIMIT, which is 70% of resource utilisation. However, as the algorithm continuously performs workload migrations, the *appPerfLimit* value gets dynamically adjusted to represent the most optimal application migration performance limit.

AWRA invokes *Linear_Regression* algorithm to detect whether the application's overall load increases or decreases. *Linear_Regression* helps identify the application instance's performance trend. When the application's performance score (*APS)* grows and exceeds the application's performance limit for migration (*appPerfLimit*), AWRA decides to migrate the workload to prevent SLA violations. Before initiating migrations, AWRA invokes a simple moving average algorithm to forecast the available time to complete workload migrations ($MT_{forecast}$).

AWRA strives to complete application migration by ranking workloads and selecting workloads that can fit the forecasted time interval. AWRA creates a prioritised workloads list ($Wklds_p$) considering two factors – user sessions handled and the migration time. While dynamic algorithms such as ASMWA assume that all workloads consume the same time for migration, AWRA considers the migration time of individual workloads. Thus, AWRA is well-positioned to migrate an optimal number of workloads without violating SLAs.

AWRA then iterates through the ranked workloads list and invokes the MigrateAppWorkloads() procedure to trigger migrations from heavily loaded instances to application running in target host with available resources. AWRA attempts to migrate as many workloads as possible within the forecasted migration time.

The migration procedure is stopped when the forecasted migration time is exceeded or when all the workloads belonging to an application are migrated. When the workloads of an application instance cannot be migrated in the estimated time duration, it is considered an SLA Violation. AWRA decides whether to adjust the application performance limit for migrations based on the SLA Violation incurred during the workload migration.

AWRA updates the application performance limit when the actual migration time exceeds the forecasted migration time or when all the workloads part of the application are not migrated (Velrajan and Sharmila, 2023). AWRA calculates APS_{delta} and

adjusts the application performance limit for migrations as given in (2).

$$APS_{delta} = \frac{(APS_{initial} - APS_{final}) * (MT_{actual} - MT_{forecast})}{MT_{actual}} \quad (2)$$

By continuously adjusting the application performance limit used to trigger migration, AWRA dynamically adapts to the indeterministic workloads served by the edge cloud infrastructure. Dynamic adaptation of AWRA to the changing workloads contributes to the improved user experience and dramatic reduction in workload quality violations.

Algorithm: Adaptive Workload Migration Algorithm (AWRA)
Input: Edge Network Topology with Applications, Edge Hosts and Edge Clusters, Resource Utilisation of
Applications, user sessions' QoE.
Output: No return value

```
proc AWRA (topology, RU, QoE)
{
appPerfLimit = DEFAULT_LOAD_LIMIT

foreach app_id in
 topology(app_instances) do

 appPerf =run
  Linear_Regression(RU(app_id))
 APS_initial= ComputeAPS(RU(app_id), QoE)

 if ((appPerf != LOAD_INCREASE) or
  (APS_initial < appPerfLimit))
  goto next app instance
 closeif

 MT_forecast = MovingAverage(RU(app_id))
 TS_start = GetTimestamp (app_id)
 Wkld_n = GetWorkloadsList (app_id)
 sort Wklds_n BY reverse=True
 Wklds_p = NULL
 MT_total = 0
 foreach wkld in Wklds_n
  if (MT(wkld) + MT_total > MT_forecast)
   break
  Wklds_p.append(wkld)
  MT_total = MT_total + MT(wkld)
 endfor

 slaViolations = MigrateAppWorkloads
              (Wklds_p)
 TS_end = GetTimestamp (app_id)
 MT_actual = (TS_end - TS_start)

 APS_final = ComputeAPS(RU(app_id), QoE))

 if ((slaViolations) or
    (MT_forecast < MT_actual))
  APS_delta = (APS_initial - APS_final) *
       (MT_actual - MT_forecast)/MT_actual
  appPerfLimit = appPerfLimit-APS_delta
 closeif
endfor
return
}
```

4 EXPERIMENTS & RESULTS

AWRA was validated in a simulated environment using a edge cloud topology built using NetworkX, a popular network topology simulation toolkit. We leveraged SciPy, Numpy and Pandas Python libraries for data analysis and reporting. We used a video streaming application profile, to validate the behavior of AWRA.

A topology similar to the real-world service provider networks is simulated using 20 edge clusters, each with 6 hosts. The tests were performed with 100 application instances running across the different edges in edge cloud. Testing was done using 25,000 user sessions accessing the video streaming server.

User tasks modelling video streaming users were created and mapped to various application instances running on the edges, randomly. The compute, storage, network and memory utilisation of each application instance was continuously monitored and recorded. AWRA keeps track of application instances' resource utilisation at the workload level granularity. Furthermore, user sessions' QoS metrics are periodically collected. APS is calculated from the application's aggregated resource consumption and users' quality of experience metrics collected periodically from the edge cloud.

As time progressed, the tasks handled by an application instance increase, resulting in increased resource consumption at the edge hosts. Our experimental findings in Fig 3 reveal a direct correlation between an application instance's hardware resource consumption and the user sessions handled. Fig 3 also shows the application instance's aggregated resource utilisation (App. RU), which grows with a rise in user sessions.

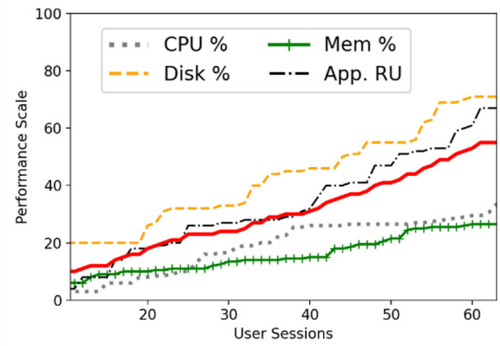


Figure 3. Overall Resource Utilisation of Application

Our experiments demonstrated AWRA's ability to prevent application overload by proactively triggering workload migrations. As AWRA migrates workloads by ranking them based on the user session count and the load reduction achieved, it reduces SLA violations and improves the QoS experienced by users, as represented in Fig 4.

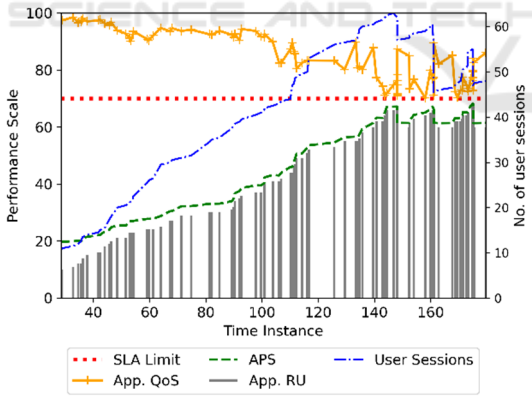
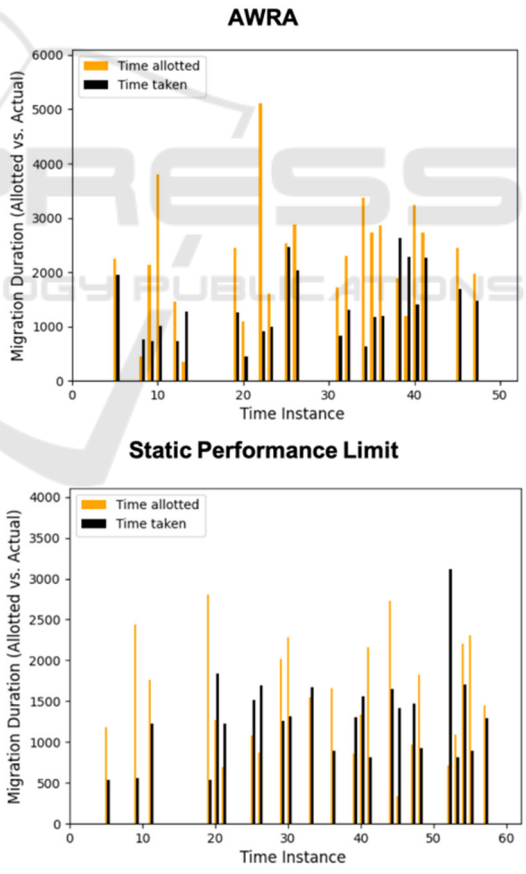


Figure 4. QoS-aware Migration of Workloads

The performance of AWRA was compared with 3 different algorithms for workload migrations – ASMWA, EWMA3 and static performance limit algorithms. We evaluated these algorithms objectively based on their ability to minimise SLA violations during the migration of application workloads in edge cloud.

In our analysis, 17% of workload migrations exceeded the allocated migration time for AWRA, 26% for ASMWA, 41% for EWMA3, and 42% for migrations done using static performance limit, as given in Figure 5. AWRA's ability to predict the available workload migration time and rank workloads substantially decreases the workload migration time violations.

Our experiments prove that a dynamic application performance limit is more suitable for handling workload migrations in environments with indeterministic workloads. Using a greedy approach to pack workloads based on migration interval helps minimise workload migration time-related violations in AWRA. Migrations done using static performance limit could not rapidly adapt to the changing performance characteristics of application instances, resulting in more SLA violations. ASMWA and EWMA3 algorithms could perform relatively better, because of their ability to adapt to real-time load conditions as given in Figure 5.



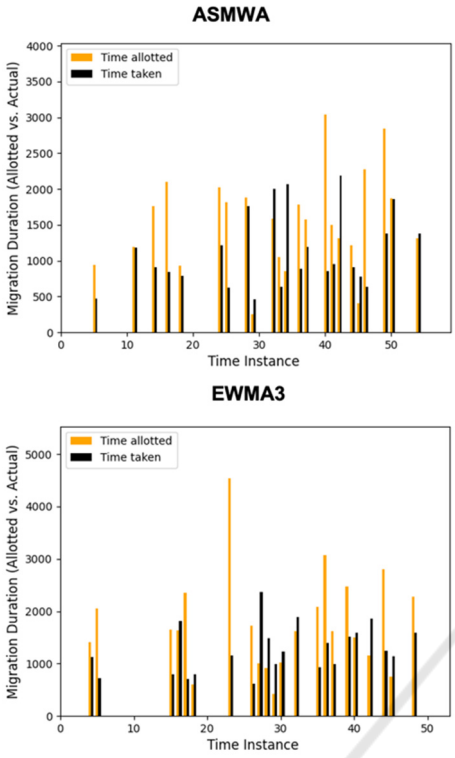


Figure 5. Workload Migration Interval – Allotted vs. Actual

We compared AWRA's performance with state-of-the-art algorithms in minimising QoS degradation. Our simulations demonstrated that AWRA reduces QoS violations by 13% compared to ASMWA and 33% compared to EWMA3. As represented in Figure 6, AWRA achieves a 67% improvement over migrations using a static performance limit. Prioritising workloads for migration and using an adaptable mechanism to decide the application performance limit enables AWRA to migrate an optimal number of workloads within the forecasted migration time.

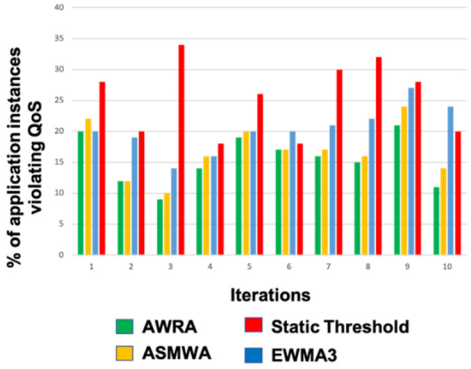


Figure 6. Comparison of QoS Violations of algorithms

The experiments showed that AWRA outperforms previously established algorithms, including ASMWA, EWMA3, and static performance limit, in adhering to SLAs and enhancing the user experience while migrating application workloads in edge cloud. Our study demonstrates that AWRA proactively adapts to changing load conditions in the edge infrastructure, compared to the existing methods used for workload migrations.

5 CONCLUSION

We proposed an Adaptive Workload Ranking Algorithm (AWRA) that performs application migrations in edge cloud by dynamically ranking workloads based on the load and migration time. AWRA distinctly uses a dynamic application performance limit for migrations, compared to the existing static approaches. This enables AWRA to quickly adapt to dynamic workloads. We studied the effectiveness of AWRA in migrating workloads at the edge and compared it with the existing algorithms - static performance limit, ASMWA and EWMA3 by running simulations. Our research established that AWRA significantly outperformed static performance limit, ASMWA and EWMA3 methods in reducing user QoS violations and improving the service quality. Compared to ASMWA, AWRA achieved a 13% improvement, while it delivered a 33% reduction in violations compared to EWMA3. AWRA performed 67% better than migrations done using static performance limit.

We observed that AWRA performs better in managing application overload conditions in edge topologies where there is gradual increase in workloads. When the workloads are unpredictable or when there are traffic bursts, 17% of workload migrations exceeded the allocated migration time using AWRA. Thus, future research can look at AWRA's efficiency under regular load conditions and at the same time, its adaptability to traffic bursts. In addition, there is opportunity to compare the performance of AWRA with other algorithms that rely on distributed decision-making methods such as PSO.

REFERENCES

Wan, X. (2024). Dynamic Resource Management in MEC Powered by Edge Intelligence for Smart City IoT. Journal of Grid Computing.

Chen, C., Song, Y., Jiang, Y., & Zhang, M. (2024). Optimizing Network Service Continuity with Quality-Driven Resource Migration. Electronics, 13(9), 1666.

Patil, A., & Patil, R. (2023). Proactive and dynamic load balancing model for workload spike detection in cloud. Measurement: Sensors.

Cao, B., Ye, H., Liu, J., Tang, B., Tao, Z., & Deng, S. (2024). SMART: Cost-Aware Service Migration Path Selection Based on Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems.

Velrajan, S., & Sharmila, V. C. (2023). Service Migrations in Multi-Access Edge Computing Using Adaptive Migration Window. In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I) (Vol. 6). IEEE.

Velrajan, S., & Sharmila, V. (2023). QoS-aware service migration in MEC using closed-loop adaptive PSO algorithm. Journal of Network and Systems Management.

Shokouhi, M. H., Hadi, M., & Pakravan, M. R. (2024). Mobility-Aware Computation Offloading for Hierarchical Mobile Edge Computing. IEEE Transactions on Network and Service Management.

Zolfaghari, R. (2024). Energy-performance-aware VM migration in cloud by using prediction and fuzzy approaches. Engineering Applications of Artificial Intelligence.

Ouhame, S., Hadi, M. Y., Mrhari, A., & Laassar, I. (2024). Artificial intelligence-based cloud-IoT resource management for energy conservation. International Journal of Advances in Applied Sciences.

VMWare DRS Migration Threshold (2024). docs.vmware.com. Retrieved from https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-resource-management/GUID-8ACF3502-5314-469F-8CC9-4A9BD5925BC2.html

Kulshrestha,S., & Patel,S.(2021). An efficient host overload detection algorithm for cloud data center based on exponential weighted moving average. International Journal of Communication Systems.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS