In: IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC) 2023, 1-5 October 2023, Barcelona, Spain
DOI:10.1109/DASC58513.2023.10311301

Distributed Trajectory Management for Urban Air Mobility Operations with Ground-based Edge Intelligence

1st Cheng Huang, 2nd Ivan Petrunin and 3rd Antonios Tsourdos
School of Aerospace, Transport and Manufacturing
Cranfield University
Bedfordshire, MK43 0AL, UK
cheng-huang.huang, i.petrunin, a.tsourdos@cranfield.ac.uk

***Abstract*—Trajectory management is a critical undertaking in urban air mobility (UAM) to ensure safe, secure, and efficient operations. Cooperative targets have the capability to report their information while managing non-cooperative targets presents a challenge in the UAM operational environment (UOE). Consequently, ground-based non-cooperative surveillance assumes a vital role in monitoring anomalies. Given the difficulties associated with implementing centralized management in a large metropolitan area, this study proposes a distributed management architecture that leverages ground-based edge intelligence to enhance resilience in performing relevant tasks. It demonstrates that employing a developed edge computing system yields superior efficiency for heterogeneous sensors and their corresponding algorithms, such as detection, fusion, and tactical conflict management, compared to typical cloud servers. Furthermore, the proposed architecture incorporates an adaptive load balancing scheme, which monitors the real-time tasks and balances tasks among multiple edge devices to enhance the efficient resource management of the edge intelligence system. Ultimately, the distributed system offers energy-saving benefits and guarantees performance, making it suitable for providing services to diverse stakeholders involved in UAM.**

***Index Terms*—Distributed trajectory management, edge intelligence, ground-based surveillance, sensor fusion**

I. Introduction

Urban air mobility (UAM) facilitates rapid passenger and cargo transit services, and its high level of automation and rapid development necessitates the deployment of advanced techniques, such as intelligent resource management, efficient trajectory management, smart communication, navigation, and surveillance (CNS), etc., to ensure safety, security, and efficiency. Managing all flights, from the strategic phase to the tactical phase, is crucial in achieving these goals [1]. Potential conflicts in submitted plans are analyzed and resolved at the strategic stage, while tactical conflicts are predicted and resolved based on real-time states obtained via cooperative or non-cooperative surveillance. Furthermore, monitoring all flights, including both cooperative and non-cooperative targets, is essential during the tactical stage.

This research was partially supported by grants from the Funds of China Scholarship Council (202008420248).

To acquire tactical information, UAM vehicles or operators presently report their states cooperatively, owing to the low operation density and complexity during the initial stage of UAM Maturity Level (UML) [2]. However, high UMLs in the future and the detection of anomaly events, such as the intrusion of unintended objects, necessitate the utilization of non-cooperative surveillance techniques, especially in critical regions, e.g., the vicinity of a vertiport. To enhance tactical management capability, the development and maturation of a smart system are crucial. Numerous studies focus on onboard intelligence, utilizing optical cameras, radar, RF detection, etc., to enable airborne detection and avoidance [3] [4]. However, one concern is that airborne equipment failure can pose a safety issue for other airspace users. Consequently, the adoption of ground-based techniques becomes imperative in establishing a robust surveillance system. Furthermore, the ground system can share monitoring information within a wide-range region with all stakeholders, providing an additional advantage.

A ground surveillance system for intelligent computing typically includes various components and functionalities to gather, process, and analyze data for intelligence purposes, physically including sensors and computing techniques. There are several computing techniques, such as general computing, intelligent computing, high-performance computing, and edge computing, of which the definition can be given as follows:

- General Computing refers to the broad field of computer science and technology that encompasses a wide range of applications and systems. Platforms such as desktop computers, laptops, mobile devices, and embedded systems provide general-purpose computing capabilities for various applications.
- Intelligent Computing focuses on enhancing computing systems with artificial intelligence (AI) techniques. Examples include neural network accelerators like NVIDIA graphics processing units (GPUs), cloud AI services, and robotics platforms.
- High-Performance Computing (HPC) specializes in executing complex computational tasks efficiently, utilizing

TABLE I
PERFORMANCE METRICS COMPARISON

Computing Paradigm	Processing Power		Latency	Scalability	Energy Efficiency	Reliability	Security	Cost
	Low	High						
General Computing	✓		Low to High	Moderate	Variable	Moderate	Moderate	Variable
Intelligent Computing	✓		Low to High	Moderate	Low to Moderate	Moderate	Moderate	Variable
High-Performance Computing		✓	Low	High	Low	High	High	Very High
Edge Computing	✓		Low	High	High	High	High	Moderate

supercomputers, cluster computing, and grid computing.

- Edge Computing brings computation and data storage closer to the edge devices, such as NVIDIA Jetson, and Raspberry Pi, etc.

The TABLE I provides a comparison of different computing paradigms based on various performance and price factors. We can find that edge computing addresses the need for real-time processing, low latency, and privacy in edge devices and IoT applications. And it prioritizes latency, scalability, and energy efficiency compared with other categories.

Some works investigated the application of edge computing to various tasks. For example, edge computing was utilized for radar-camera fusion [5], in which camera and radar information were pre-processed on separate edge devices and made an association later. Tracking-level fusion with multiple sensors, e.g. camera, lidar, and radar, was also able to be performed with the edge system [6]. Other requirements like real-time fault diagnosis could also be achieved with the edge devices for the application of rotating machines with multi-sensor data [7]. In this way, edge computing offers several advantages when applied to various tasks, e.g. sensor fusion and trajectory management. By bringing computational capabilities closer to the data source, edge computing enables real-time processing and analysis at the edge of the network, offering distinct benefits.

As a result, this research proposes and experimentally evaluates a general architecture based on distributed edge computing to explore innovative techniques for managing UAM services, e.g., sensor fusion for trajectory prediction with ground non-cooperative surveillance, as well as trajectory management.

The contribution of this paper can be summarized as follows:

1) The architecture of managing UAM services with distributed edge intelligence is proposed to effectively manage UAM services. This architecture takes advantage of the capabilities of edge devices to perform various distributed operational tasks.
2) The adaptive load balancing scheme is integrated into the intelligence system to balance the resource dynamically, considering the real-time status of the device.
3) Edge-level multi-sensor fusion and subsequent tasks, such as tactical conflict detection, are successfully deployed on the edge computing system. Notably, each deployed approach achieves similar accuracy levels while maintaining acceptable inference times and remaining competitive in terms of performance metrics, such as CPU/GPU usage and GPU temperature, compared to cloud computing for UAM services.

Overall, these contributions enhance the effectiveness and efficiency of UAM services by leveraging distributed edge intelligence, adaptive load balancing, and edge-level multi-sensor fusion.

II. METHODOLOGY

In this section, the detailed architecture and workflow of edge computing for UAM operation management are demonstrated. And the workload balancing scheme for the edge system is also described to manage the system resources.

A. Architecture of the Edge Intelligence

The detailed configuration is illustrated in Fig.1, which demonstrates the construction of a ground infrastructure incorporating heterogeneous sensors. Data streams from sensors of ground infrastructure are processed by the distributed edge computing platform. Especially, edge computing can reduce the transmitting and workload for the cloud server side, while simultaneously improving the system's robustness. Subsequently, analyzed flight states on the edge are forwarded to stakeholders, e.g. the provider of services to UAM (PSU), UAS Service Suppliers (USS), and Vertiport Automation System (VAS), for various applications such as intruder tracking, trajectory deconfliction, etc. Particularly, the designed edge system is modular and distributed, which can enable scalability as well as automation.

Based on the given structure, the architecture can be divided into two stages: edge-level multi-sensor fusion and subsequent-task accomplishment, and are described as follows:

1) Edge-level multi-sensor fusion. The disparate data from heterogeneous sensors, e.g., camera, lidar, and radar, are gathered by the edge device, is dynamically fused with developed camera-lidar and camera-radar fusion techniques for resilient multi-object detection, tracking, and prediction under various environmental conditions.

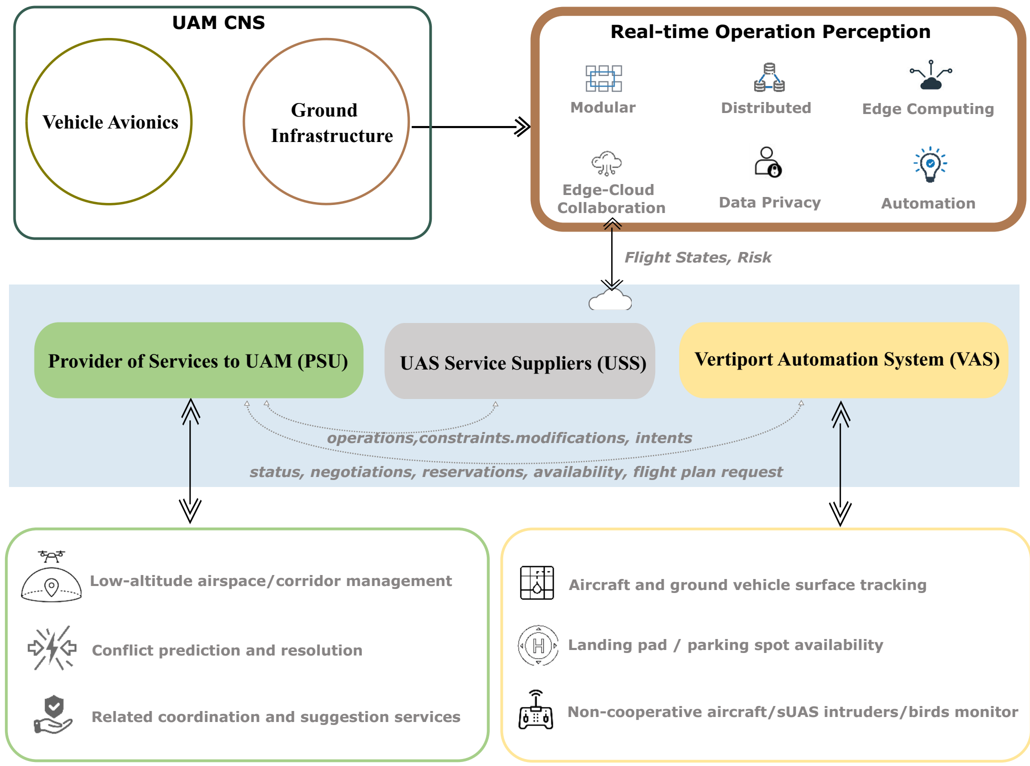

Fig. 1. The architecture of managing UAM services with edge intelligence.

The forecasted insights will be critical for stakeholders' follow-on activities.

2) Subsequent-task accomplishment. The predicted information is utilized across a myriad of scenarios and applications. In instances where broad-airspace surveillance is necessary, the Provider of Services to Urban Air Mobility (UAM) shall avail themselves of this information for managing low-altitude airspace, performing tactical conflict resolution, and providing additional services. Similarly, when the sensor network is deployed in and around critical zones such as vertiports or vertihubs, the Vertiport Automation System (VAS) must analyze this data to determine landing pad availability, track surface trajectories, and mitigate potential conflicts with non-cooperative targets.

The two-stage view is then split into practical steps as in Fig. 2, including (1). hardware and sensor acquisition, (2). edge device setup, (3). sensor stream acquisition, (4). edge-level object detection and fusion with deep learning, (5). trajectory prediction, (6). tactical conflict management, (7). system integration for workload balancing, and (8). final test and optimization. Following those steps, UAM-related services can be deployed and evaluated with edge intelligence.

B. Adaptive Load Balancing

The edge system comprises a lot of devices for computing. To distribute tasks or workloads efficiently among multiple edge devices in a network, Feedback Control Load Balancing is used in edge computing [8] [9]. It aims to achieve optimal resource utilization, improve performance, and maintain system stability.

Adaptive load balancing involves several steps, which is also depicted in Fig. 3:

1) Task monitoring: The system continuously monitors the performance metrics of edge devices, such as CPU usage p_{cpu}, CPU temperature t_{cpu}, memory usage mem, GPU usage p_{gpu} and GPU temperature t_{gpu}. These metrics provide insights into the current workload and resource availability of each device.

2) Load Check: Based on the collected metrics, the system evaluates the load on each edge device. We assign weights to each metric to reflect their relative importance in determining the overall device load. Let $w_{p_{cpu}}$ be the weight for CPU usage, $w_{t_{cpu}}$ be the weight for CPU temperature, w_{mem} be the weight for memory usage, $w_{p_{gpu}}$ be the weight for GPU memory usage, and $w_{t_{gpu}}$ be the weight for GPU temperature. Then the weighted formula for device load can be represented as in Eq. (1):

$$L = w_{p_{cpu}} \cdot p_{cpu} + w_{t_{cpu}} \cdot t_{cpu} + w_{mem} \cdot mem + w_{p_{gpu}} \cdot p_{gpu} + w_{t_{gpu}} \cdot t_{gpu} \tag{1}$$

And the real-time load L is compared with the load limitation L_{limit} across all devices. If the load deviation exceeds a certain threshold $L_{threshold}$, load balancing actions, such as task offloading or migration, are triggered.

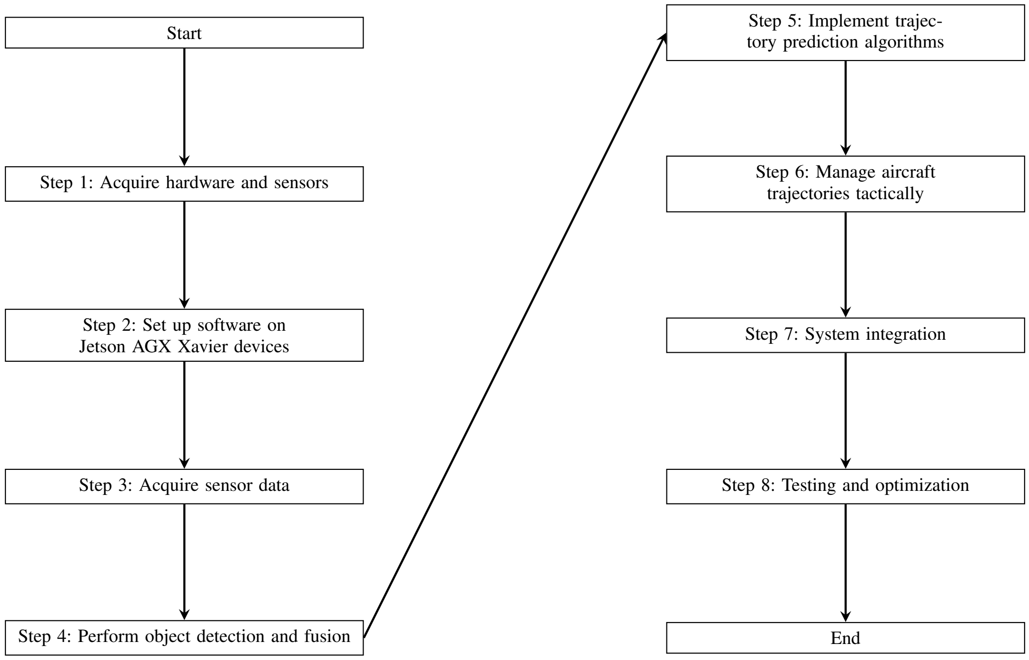

Fig. 2. Workflow diagram

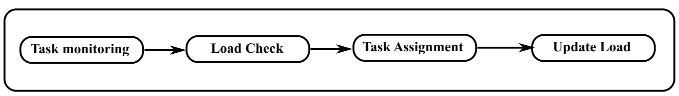

Fig. 3. The flow chart of adaptive load balancing.

3) Task Assignment: The overflow part of the load will be resolved by randomly transferring some tasks to other available devices.

4) Update Load: The load distribution is updated and repeats the process continuously.

Finally, the detailed steps are also illustrated in Algorithm 1.

III. EXPERIMENT

In this section, the edge intelligence system is constructed for evaluating various tasks in UAM. Deep learning-based object detection, sensor fusion, and trajectory management are applied to kinds of scenarios for en-route flight conflict management. The experiments finally prove the performance of the edge computing system.

A. System Construction

The experiment is carried out on the Multi-User Environment for Autonomous Vehicle Innovation (MUEAVI) road at Cranfield University. The sensor network in the MUEAVI system consists of cameras, lidars, and radars, whose deployment layout is depicted in Fig. 4, and in which Jetson AGX Xavier platforms enable a distributed edge computing system.

The Jetson AGX Xavier is selected as it is often considered better in terms of AI performance for several reasons: (1) It is specifically designed for AI applications and comes with

Algorithm 1: Adaptive Load Balancing

```
Input: Load limit L_limit, threshold L_threshold, load
       L[n], last_adjustment[n] for n devices
while True do
    for i = 0 to n do
        if load[i] < target_load − threshold or
         load[i] > target_load + threshold then
            for j = 0 to n do
                if load[i] > target_load + threshold
                 and
                 load[j] < target_load − threshold
                 then
                    Adjust the load by load balancing
                     algorithm;
                    Transfer one task from platform i to
                     platform j;
                    Update load[i] and load[j];
                    Update last_adjustment[i] and
                     last_adjustment[j];
                end
            end
        end
    end
    for i = 0 to 3 do
        Execute task on platform i;
    end
end
```

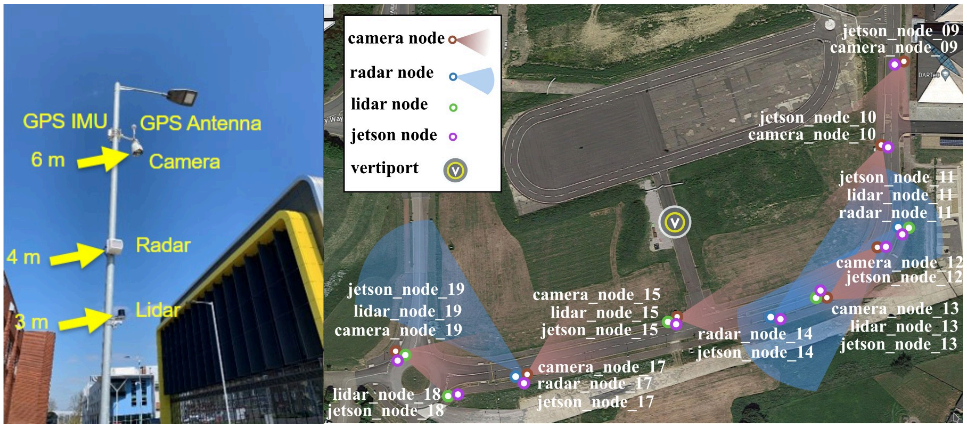

Fig. 4. The smart devices of the MUEAVI system.

dedicated hardware accelerators, such as Tensor Cores and deep learning accelerators. (2) It is optimized for energy efficiency, allowing for efficient processing of AI workloads while keeping power consumption relatively low. (3) NVIDIA provides comprehensive software support for the Jetson platform, including libraries, frameworks, and development tools specifically tailored for AI development, such as JetPack SDK.

To determine the performance level of the edge intelligence system, a cloud-side server, which equips with Nvidia RTX 2080 Ti, is also running for algorithm deployment. The detailed parameters are compared in TABLE II. It's important to note that the RTX 2080 Ti is primarily targeted at high-performance computing and gaming applications, where the number of CUDA Cores and Tensor Cores excels. However, for AI-focused workloads, the dedicated AI accelerators and energy-efficient design (30W/device) of Jetson AGX Xavier make it a compelling choice.

B. Result Analysis

We conducted a series of flight trials using small UAVs along the MUEAVI road, during which we collected a dataset specifically for training deep learning approaches. In particular, we deployed the DeepAssociation approach, which is a framework proposed in our previous work [10], for radar-camera detection, fusion, and tracking. These correspond to steps 4 and 5 in Fig. 2. Additionally, we utilized the graph-based tactical deconfliction algorithm, which was previously developed in our work [11], to validate short-term trajectory management, corresponding to step 6 in Fig. 2. All of these algorithms were deployed on both the edge computing system and the cloud server for evaluation and comparison.

In this experiment, the schematic diagram of managing UAM services with a cloud server is demonstrated in Fig. 5 at first. In this setup, camera streams from nodes 12 and 15, and radar stream node 14 are directly input to the server workstation. The server centrally processed all data and utilized the aforementioned approaches to accomplish object detection, fusion, and trajectory management tasks. As a comparison, the raw data from each sensor in the edge computing system was processed individually by specific edge platforms. More specifically, camera-related tasks are handled by Jetson nodes 12 and 15, the radar information is prepared by Jetson node 14, and the fusion and tactical deconfliction works are conducted on Jetson node 11. This configuration highlights the connection and flexibility of the edge computing system

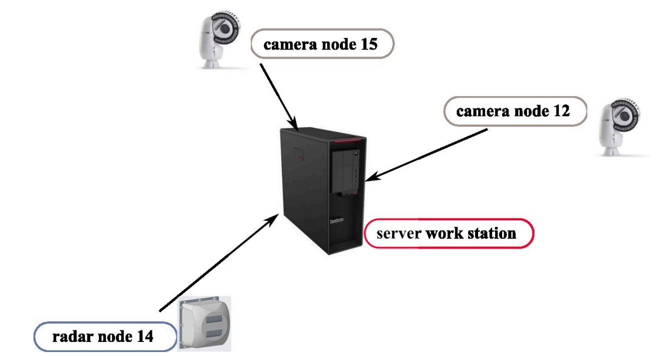

Fig. 5. The schematic diagram of managing UAM services with cloud server.

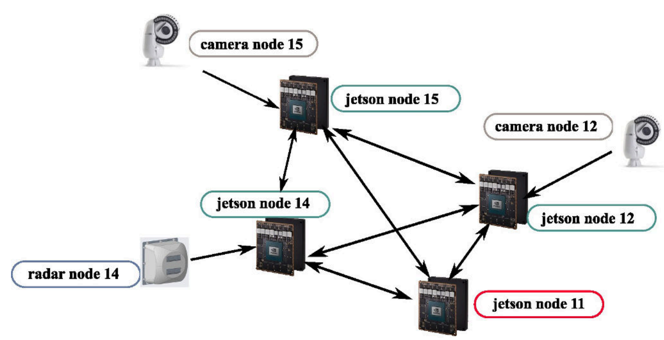

Fig. 6. The schematic diagram of managing UAM services with edge intelligence.

A comparison of the object detection performance between the server and the edge computing system reveals that the pre-processing and inference time on the edge side is longer than

TABLE II
DETAILED PARAMETERS COMPARISON: NVIDIA RTX 2080 TI VS JETSON AGX XAVIER

Parameter	**NVIDIA RTX 2080 Ti**	**Jetson AGX Xavier**
GPU Architecture	Turing	Volta
CUDA Cores	4352	512
Tensor Cores	544	64
GPU Memory	11 GB GDDR6	16 GB LPDDR4x
Memory Bandwidth	616 GB/s	137 GB/s
Max Power Consumption	260 W	30 W
Storage	2TB	32 GB eMMC 5.1
Dimensions	Dual-Slot, 267 mm length	105 mm x 105 mm

those on the server side, as presented in TABLE III. However, despite the slightly longer processing time, the edge computing system maintains an acceptable level of accuracy, comparable to that of the server.

TABLE III
COMPARISON OF SPEED FOR OBJECT DETECTION(IN MILLISECONDS)

Device	**Pre-process (ms)**	**Inference (ms)**	**NMS (ms)**
Jetson AGX Xavier	1.5	51.8	1.9
Server (2060Ti)	0.3	16.4	0.4

TABLE IV
COMPARISON OF SPEED FOR CAMERA-RADAR FUSION AND TACTICAL DECONFLICTION

Device	**Camera-Radar Fusion (ms)**	**Tactical Deconfliction (s)**
Jetson AGX Xavier	151.5	1.31
Server (RTX 2060 Ti)	87.6	0.73

To evaluate the distributed deployment of fusion and tactical deconfliction on the edge platforms, a noticeable difference can be observed from TABLE IV. The processing time on edge computing devices increases significantly, almost doubling that of servers due to the hardware performance disparity. However, the edge computing system can still be capable of maintaining the same level of accuracy as the servers since they share the same inference model.

During the inference process, it is crucial to consider not only accuracy but also changes in significant metrics and overall efficiency. We present a comprehensive comparison of these metrics in TABLE V. It is worth noting that the four-device edge platform demonstrates the ability to maintain lower power consumption while achieving high computational performance, leading to increased efficiency. Furthermore, the distributed system exhibits a larger total storage and memory capacity, enabling it to accommodate a greater volume of essential caches.

Referring to those physical metrics, we can find the specific difference when running the deployed algorithms as in Fig. 7. For instance, when comparing Jetson platforms to the server workstation, the deployed algorithms exhibit slightly higher memory consumption on the former. Furthermore, a noticeable increase in core temperature is observed for the server workstation, indicating intensive processes involved in centralized object detection, fusion, and trajectory deconfliction. In contrast, Jetson devices experience minimal changes in core temperature. Additionally, GPU usage on the server workstation rises by approximately 30%, whereas Jetson platforms utilize a maximum of 11.8% of the GPU. This finding suggests that edge devices are in a state of low energy consumption operation.

It is also critical to assess the real-time workload on each device in the edge computing system and balance the load adaptively. We employ Eq. (1) to obtain the load at first. We assign $w_{p_{cpu}} = 0.2$, $w_{t_{cpu}} = 0.2$, $w_{mem} = 0.1$, $w_{p_{gpu}} = 0.3$, and $w_{t_{gpu}} = 0.2$. Prior to executing the algorithms, the normal load for Jetson node 11 is determined as $L_{node11} = 0.341$, with the following initial metrics: $p_{cpu} = 2.0\%$, $t_{cpu} = 52.2°C$, $mem = 15.6\%$, $p_{gpu} = 42.1\%$, $t_{gpu} = 45.5°C$. Similarly, the initial loads for Jetson nodes 12, 14, and 15 are obtained as $L_{node12} = 0.344$, $L_{node14} = 0.331$, and $L_{node15} = 0.339$, respectively.

The device load limitation is set as $L_{limit} = 0.500$ and the threshold is defined as $L_{threshold} = 0.01$. The loads are visually compared in Fig. 8 and remain within the safe range of load limitation. If the load on any device exceeds this limitation, the offloading procedure will be triggered.

To demonstrate the offloading scheme, we add more tasks to Jetson 11 and observe the load changes. Consequently, the load value increases to $L_{node11} = 0.515$ which surpasses the acceptable margin as depicted in Fig. 9. At this point, the adaptive load balancing decides to transfer some tasks to Jetson node 14. Therefore, the load on Jetson 11 decreases

TABLE V
PERFORMANCE METRICS COMPARISON: JETSON AGX XAVIER ($\times 4$) VS. 1 RTX 2080 TI ($\times 1$)

Performance Metric	**Jetson AGX Xavier** ($\times 4$)	**RTX 2080 Ti** ($\times 1$)
Power Consumption (W)	120	260
Computational Ability (TFLOPS)	128	13.4
Storage (GB)	64 (eMMC)	11 (GDDR6)
Memory (GB)	64 (LPDDR4x)	11 (GDDR6)
Memory Bandwidth (GB/s)	2744	616
Efficiency (TFLOPS/W)	1.07	0.05

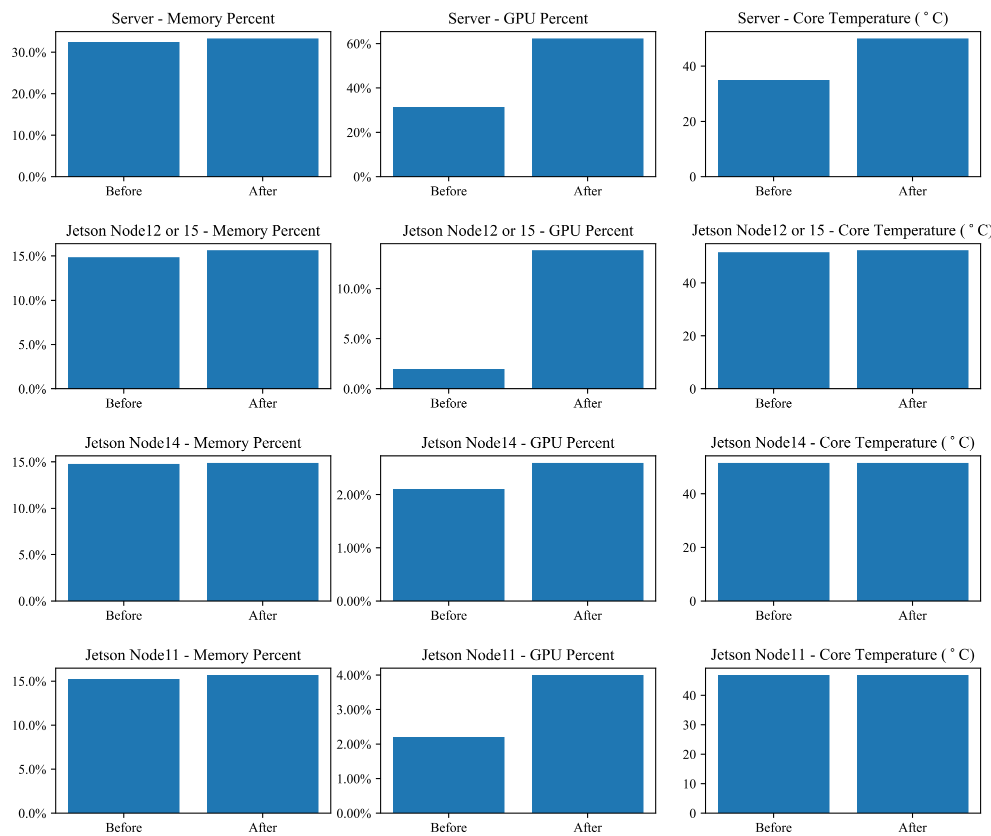

Fig. 7. Metric comparison.

to $L_{node11} = 0.464$, while the load on Jetson 14 increases to $L_{node14} = 0.390$ as illustrated in Fig. 10. These results indicate that the loads across all devices have been appropriately distributed and balanced.

Finally, we conclude that all deployed algorithms, e.g. multi-sensor fusion, and trajectory deconfliction can maintain performance on the designed edge intelligence system. Satisfactory performance and fast responsiveness of the edge platform, which employs adaptive load balancing for resources management, prove the capability for trajectory management, as well as various tasks for stakeholders.

IV. CONCLUSION

In the paper, an example of the infrastructure system is demonstrated and one general architecture with edge intelligence is developed, which may be deemed a good endeavor in terms of validating the effectiveness of the ground-based non-cooperative surveillance, e.g. object detection and fusion, and subsequent tasks for stakeholders, e.g. tactical deconfliction, as there is no similar well-functioning system and conditions for UAM research. The performance of all deployed algorithms

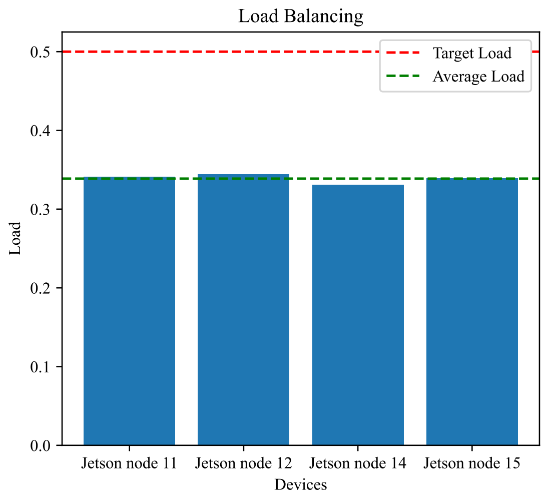

Fig. 8. Load comparison.

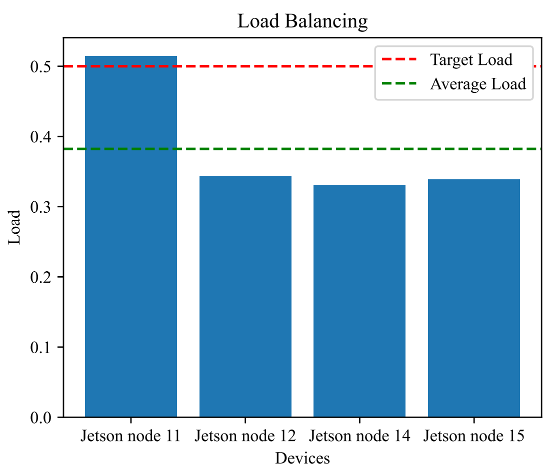

Fig. 9. Load comparison before offloading.

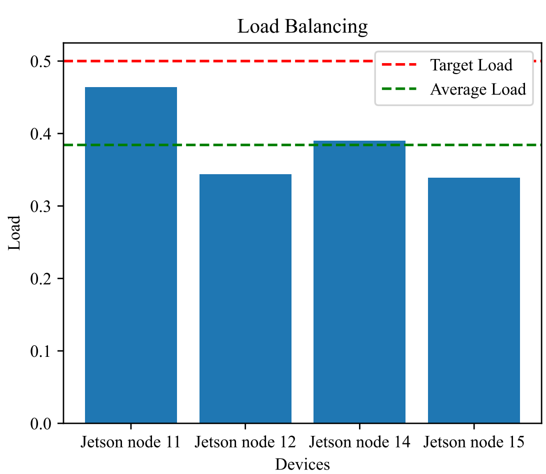

Fig. 10. Load comparison after offloading.

can be guaranteed by taking advantage of edge computing, such as efficiency, reliability, cost, etc. Meanwhile, the adaptive load balancing scheme proves to be able to ensure task balance among all edge devices.

With the proposed framework, the implementation of more practical UAM applications becomes possible, thereby promoting data-driven and digital management for sustainable urban environments. Future works include more intelligent works on the edge computing system, e.g. deep learning-based load balancing and collaboration scheme, to increase the intelligence level and resilience to uncertainties or device failures of the system.

References

[1] V. L. Stouffer, W. B. Cotton, R. A. DeAngelis, D. M. Devasirvatham, T. B. Irvine, R. E. Jennings, R. D. Lehmer, T. C. Nguyen, M. A. Shaver, and C. J. Bakula, "Reliable, secure, and scalable communications, navigation, and surveillance (cns) options for urban air mobility (uam)," 2020.

[2] B. P. Hill, D. DeCarme, M. Metcalfe, C. Griffin, S. Wiggins, C. Metts, B. Bastedo, M. D. Patterson, and N. L. Mendonca, "Uam vision concept of operations (conops) uam maturity level (uml) 4," 2020.

[3] C. Kang, H. Chaudhry, C. A. Woolsey, and K. Kochersberger, "Development of a peripheral–central vision system for small unmanned aircraft tracking," *Journal of Aerospace Information Systems*, vol. 18, no. 9, pp. 645–658, 2021.

[4] Y. Miao, Y. Tang, B. A. Alzahrani, A. Barnawi, T. Alafif, and L. Hu, "Airborne lidar assisted obstacle recognition and intrusion detection towards unmanned aerial vehicle: Architecture, modeling and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4531–4540, 2020.

[5] Y. Fu, D. Tian, X. Duan, J. Zhou, P. Lang, C. Lin, and X. You, "A camera–radar fusion method based on edge computing," in *2020 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2020, pp. 9–14.

[6] M. Gabb, H. Digel, T. Müller, and R.-W. Henn, "Infrastructure-supported perception and track-level fusion using edge computing," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1739–1745.

[7] G. Qian, S. Lu, D. Pan, H. Tang, Y. Liu, and Q. Wang, "Edge computing: A promising framework for real-time fault diagnosis and dynamic control of rotating machines using multi-sensor data," *IEEE Sensors Journal*, vol. 19, no. 11, pp. 4211–4220, 2019.

[8] B. V. Shobhana, S. Narayana, and B. Nath, "Load balancers need in-band feedback control," in *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, 2022, pp. 76–84.

[9] E. Gures, I. Shayea, M. Ergen, M. H. Azmi, and A. A. El-Saleh, "Machine learning-based load balancing algorithms in future heterogeneous networks: A survey," *IEEE Access*, vol. 10, pp. 37 689–37 717, 2022.

[10] C. Huang, I. Petrunin, and A. Tsourdos, "Radar-camera fusion for ground-based perception of small uav in urban air mobility," in *2023 IEEE INTERNATIONAL WORKSHOP ON Metrology for AeroSpace (MetroAeroSpace)*, 2023.

[11] C. Huang, I. Petrunin, and A. Tsourdos, "Dynamic graph propagation for performance-based tactical conflict resolution in urban air mobility," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 152–158.

2023-11-10

Distributed trajectory management for urban air mobility operations with ground-based edge intelligence

Huang, Cheng

IEEE

Huang C, Petrunin I, Tsourdos A. (2023) Distributed trajectory management for urban air mobility operations with ground-based edge intelligence. In: IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC) 2023, 1-5 October 2023, Barcelona, Spain

https://doi.org/10.1109/DASC58513.2023.10311301