IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 29 July 2023, accepted 23 August 2023, date of publication 31 August 2023, date of current version 18 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3310808

RESEARCH ARTICLE

Improving Application Support in 6G Networks With CAPOM: Confluence-Aided Process Organization Method

JAMIL ABEDALRAHIM JAMIL ALSAYAYDEH[1], (Member, IEEE), IRIANTO[2], AHMED JAMAL ABDULLAH AL-GBURI[1], AND SAFARUDIN GAZALI HERAWAN[3]

[1]Department of Electronics and Computer Engineering Technology, Fakulti Teknologi Kejuruteraan Elektrik dan Elektronik (FTKEE), Universiti Teknikal Malaysia Melaka (UTeM), Melaka 76100, Malaysia
[2]Department of General Education, Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates
[3]Industrial Engineering Department, Faculty of Engineering, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding author: Jamil Abedalrahim Jamil Alsayaydeh (jamil@utem.edu.my)

ABSTRACT Systems requiring terahertz transmission and high sampling capabilities can be supported by sixth-generation (6G) technology with minimal latency and excellent service throughput. Regardless of the distributions of data and services, High-Performance Computing (HPC) enhances speed and provides diversified applications and functionality. The Confluence-Aided Process Organization Method (CAPOM) is suggested in this article to take advantage of process allocations while using an HPC paradigm. The process allocations and completions are scheduled based on prior and current system conditions to minimize waiting time based on the 6G qualities. This implies that state requirements for process allocation, distribution, and completion are carried out with the assistance of federated learning. The initial state allocations are based on the user/application request; in other allocations, the application's request for completion time and capacity for processing are considered. Offloading and shared processing are, therefore combined to maximize resource deliveries. The federated learning states are checked post-completion times to mitigate the waiting duration of dense service demands. Indicators such as distribution ratios, latency, wait time, and processing rate are considered for the effectiveness of the proofs. The suggested CAPOM achieves an 8.67% higher processing rate, 9.09% reduced latency, 8.76% less wait time, and a 6.73% higher distribution ratio for the various capacities.

INDEX TERMS 6G, federated learning, HPC, process allocation, service distribution.

I. INTRODUCTION

High-performance computing (HPC) is a progressive approach that supports providing better services or performance to users and systems by delivering high-quality solutions for the problems in the system or computer. HPC, or supercomputing, performs at high-quality computer performance [1]. In an aggregated computing process, better user performance is administered through precise allocations. HPC is widely used in designing new products and producing better products to avoid unwanted threats or errors [2]. HPC is also used in decision-making to test scenarios or problems and to provide improved solutions for product development on a computer. HPC systems often use clusters of networks to build a complete HPC system [3]. The cluster monitors and stores network data to process input and output from the processing unit and facilitates performing a particular task. HPC is most commonly used in a remote processing system to provide better service to users without compromising the quality of the service. HPC is more reliable, efficient, and easier to manage than other systems or processes and offers better scalability. HPC is used to solve large problems in engineering or computer fields and provides better solutions for the problems. HPC employs the graph-oriented programming

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

(GOP) technique to understand the exact details used to solve the problems [4], [5]. 6G refers to the sixth-generation wireless network, which provides a better communication process among users using a cellular data network. As a successor to 5G, 6G is widely used in communication systems [6]. 6G offers the fastest speed, which is vital for communication. It is much faster than 5G in transferring data from one user to another, providing a significantly improved user communication process [7]. Additionally, 6G enhances the quality of service (QoS) in wireless communication systems. Reducing the latency rate in the communication process and utilizing a high-frequency ratio sustains the communication process's capacity [8]. HPC is utilized in 6G to enhance the system's performance by understanding the problems present in the networks. Computing systems are mostly used in computers and devices, which help to provide better service to users by identifying threats or problems and solving them by offering the best solutions. HPC resolves problems in the communication process and provides better service to the users by understanding the exact details of the networks [9], [10]. High-performance computing (HPC) requires a large amount of data motion processes and computational systems to provide better solutions to the problems present in the network.

Machine learning (ML) algorithms are used in HPC to solve problems by providing a deep learning algorithm to understand the volumes of data or threats [11]. The ML approach utilizes more data to deliver better performance to the network or users [12]. HPC is mostly used in many fields, such as the Internet of Medical Things (IoMT) and the Internet of Things (IoT), which aids in enhancing the total performance of the system by providing better QoS [13]. Within networks, HPC plays a significant role in identifying and solving problems arising from data management and unnecessary data transfer processes. ML-assisted HPC is vital in many fields, including the computational screening process [14]. The QoS metrics in HPC include throughput, accessibility, security, reaction time, and dependability. Computational screening is one of the main processes in HPC, aiming to provide better service and quality for the users. ML is also used in HPC to predict or identify the exact high performance while training the data available in the model. Furthermore, ML-assisted HPC is used to detect failures or failed processes, which may affect the system's overall performance. ML helps to find out the exact cause of failure and provides better solutions to the problems [15], [16]. However, existing systems achieve high latency and minimum throughput when implementing 6G communication. The research difficulties are overcome by applying High-Performance Computing functionalities (HPC), which speed up the communication process. The Confluence-Aided Process Organization Method (CAPOM) is motivated and suggested to improve the overall allocation with minimum latency. Then, the overall contribution and motivation of the work are listed as follows.

- Designing the Confluence-Aided Process Organization model (CAPOM) to improve process allocation.
- Introducing High-Performance Computing (HPC) functions for reduced latency and maximize throughput values.
- The experimental outcomes have been implemented and the suggested CAPOM model achieves high performance, low latency and high service rate

The remainder of this paper shall be arranged in the following manner: Section II will introduce the related research. Section III describes the proposed confluence-aided process organization method. Processing capacity allocation and service distribution using federated learning for Low-Latency and High-Service-Rate applications are motivated and described in Section IV, followed by related analysis and discussion in Section V. Finally, we summarize our conclusions in Section VI.

II. RELATED WORKS

Mavromoustakis et al. [17] proposed a new offload-aware recommendation scheme for the Internet of Things (IoT). This offload scheme enables **e**very available service for users and improves performance by enhancing Qualities of Experience (QoE) and Quality of Service (QoS). It also provides a machine communication process by enabling the resources available in the recommendation scheme. Numerical outcomes demonstrate that the suggested scheme reduces the energy consumption rate of the system and provides a better user experience. Lin et al. [18] presented a machine communication process by enabling the resources available in the recommendation scheme. Numerical outcomes demonstrate that the suggested scheme decreases the energy consumption rate of the system and provides a better user experience. Yoon et al. [19] implemented a deep neural network (DNN)–based object detection offloading framework for mobile edge devices. The proposed method is mainly used to decide whether to issue offload or not to the particular process and create a proper data set for further processing. The proposed framework is used to identify the exact objects under process and increases the accuracy ratio in the detection process. Simulation outcomes illustrate that the recommended framework enhances the efficacy and efficiency of the object detection process. Xu et al. [20] suggested a blockchain-enabled resource management process for the 6G communication process. The blockchain approach is widely used in many Internet of Everything (IoE) based systems and devices to enhance the total efficiency and performance of the device. The blockchain method provides a better integration process, which helps improve the monitoring and management process by utilizing the resources available in the database. The proposed blockchain-enabled method improves the overall effectiveness of the system.

Yan and Choudhury [21] discussed a deep Q-learning-based joint optimization approach to perform offloading tasks for mobile edge computing (MEC) systems. The proposed

Q-learning approach reduces the complexity of the device and provides optimal solutions for the optimization process. It also reduces the latency rate while performing offloading tasks, which helps to improve the Quality of Service (QoS). Numerical outcomes demonstrate that the suggested technique improves the overall performance and system effectiveness and decreases the energy consumption ratio in the computation procedure. Khan et al. [22] proposed an efficient hybrid deep learning-enabled model for the congestion control process in 5 G-based networks. Congestion control plays an important role in the 5G network, which provides better Quality of Service (QoS) to the users. The long short-term memory (LSTM) algorithm is used in the proposed deep learning-enabled approach to improving the network's offloading process. Compared with other existing approaches, the suggested technique enhances the accuracy ratio and decreases the time consumption ratio, enhancing the network's efficiency. Zhang and Fu [23] deliberated an energy-efficient computation offloading scheme named Dynamic programming-based energy-saving offloading (DPESO) for task scheduling processes in an edge-computing system. The proposed method is based on the time-division multiple access methods mainly used in the scheduling process. DPESO is primarily utilized to decrease the latency rate in the computation process. Simulation outcomes demonstrate that the recommended DPESO technique increases the system's efficiency by decreasing the energy consumption rate. Mukherjee et al. [24] proposed a layered message transfer framework for the social Internet of Things (IoT) utilizing a software-defined network (SDN). The proposed framework is widely used in IoT-based devices to transfer messages from one user to another without lagging or failure. SDN is used to improve the slices available in the optimization process. It is also used to manage user messages and generate a proper dataset for further use. Numerical outcomes illustrate that the suggested technique improves the Quality of Service (QoS) in social scenarios and increases the coverage area using the SDN approach. Naouri et al. [25] introduced a three-layered task offloading framework named DCC for mobile edge computing (MEC) systems. DCC stands for cloudlet layer, device layer, and cloud layer. DCC is used to perform offloading low-quality tasks and produce a proper communication process for the users by improving the efficiency of the computation process. A greedy task graph partition offloading algorithm is used in DCC to perform scheduling tasks for the optimization process. Simulation outcomes display that the suggested DCC framework enhances the total performance of the system by when compared with other techniques. Alqahtani et al. [26] proposed a proactive caching technique with offloading (PCTO) approach for mobile edge computing (MEC) systems by using the Machine learning approach. The deep recurrent learning algorithm is used in PCTO to improve the interval that occurs while providing user services. To perform offloading instances, certain data are trained using the PCTO scheme. Compared with other approaches, the suggested PCTO enhances the system's total performance by decreasing the failures in caching and offloading process. Chakrabarti. [27] presented a new offloading approach using a deep learning method for mobile augmented reality (MAR) applications. The proposed offloading approach uses a deep reinforcement algorithm (DRL) to improve energy constraints and offloading tasks and transfer data for further services. The proposed method is used to divide certain phases to perform offloading tasks with the help of the DRL approach. Computer vision algorithm is also used in MAR applications to perform computation processes without any energy consumption rate. Guo and Zhang [28] proposed a fairness-oriented computation offloading process for the cloud-assisted edge computing system. The suggested technique improves the offloading approach by performing certain data transmission strategies. The optimal cloud-edge strategy is used here to analyze the data to recognize the optimal offloading strategy available in the system. Numerical outcomes demonstrate that the suggested technique enhances the system's performance by decreasing the latency ratio of response time for mobile users. Shahidinejad et al. [29] introduced a context-aware multi-user offloading approach for mobile edge computing (MEC) systems. It is used in a multi-user system to collect contexts. The Federated learning (FL) algorithm uses the offloading approach to use distributed capabilities to enhance the system's total performance. Compared with other approaches, the recommended technique increases the efficiency and efficacy of the system by enhancing the accuracy ratio in the offloading progression and reducing the energy consumption ratio in the computation procedure. Chen and Wang [30] introduced a decentralized computation offloading approach for multi-user mobile edge computing (MEC) systems by using deep deterministic policy gradient (DDPG). The proposed DDPG is used to identify the offloading strategies and produce a proper dataset to improve the efficiency of the MEC users. Numerical outcomes display that the suggested DDPG approach enhances the users' total performance and quality of services. In their work, Ali et al. [31] proposed a new multi-task computation offloading approach using an allocation memory algorithm for device-to-device communication. A fit algorithm is also used to design tasks on multiple devices. The proposed method improves performance by performing proper offloading tasks. Compared with other techniques, the recommended approach enhances the Quality of Service (QoS) in cell scenarios and decreases the latency ratio in the computation progression. Similarly, Kathole et al. [32] applied an energy-aware blockchain model in 6G network IoE applications. This study uses the cyber twin-related UAV 6G network structure to respond to the user request by managing the communication resources. During this process, blockchain is applied to improve security while sharing resources in a cloud environment. Additionally, Chen et al. [33] recommended **a** User-Centric Resource Allocation in 6G from an economic perspective. This study provides the economic perspective of quality solutions for every user request. The quality of experience is provided according to the user's subjective values, and the users are prioritized to allocate the resources. Market rule and auction theory are integrated to improve resource allocation efficiency during the evaluation. Lastly, Alsulami et al. [34] introduced a federated deep learning approach to manage the resources and optimize the quality of services in 6G. Machine learning and cutting-edge technologies are widely applied for the resource allocation process in 5G, and the federated reinforcement learning approaches are incorporated with vehicle communication to improve the quality of services in 6G. Dong et al. [35] applied the United Framework of Integrated Sensing and Communications (ISAC) to improve the resource allocation process in 6G. Every request's probability value is computed with the help of the Cramer-Rao Bound approach that identifies the resources QoS, location, and tracking is performed. The effective identification of resources maximizes resource allocation flexibility and efficiency. Guo et al. [36] introduced a federated reinforcement learning approach for allocating resources in device-to-device communication in 6G. The main intention of this study used to minimize power consumption and maximize the sum capacity by providing quality services to the user request. Sheng et al. [37] recommended a coverage enhancement process to allocate the resource by considering the resource configuration and constellation. This study uses the satellite-terrestrial integrated network to analyze the user request and configurations to improve the overall resource allocation efficiency. In their work, Ashwin et al. [38] applied a hybrid quantum deep learning model to manage resources in 6G. The hybridized approach uses recurrent and convolutional neural networks to estimate resource distribution, configuration, and slice collection. From the estimated information, load balancing and error are computed using recurrent networks. This process helps to manage the QoS in resource allocation. In another study, Thantharate et al. [39] introduced an adaptive network slicing structure for resource management in 6G systems. This study employs transfer learning with a network slicing structure to predict the load, resulting in a 30% lower error rate and a maximizing the correlation coefficient by 6%. Moreover, Han et al. [40] recommended an equity, diversity, and inclusion (EDI)-based resource management process in 6G applications. The EDI-based approach analyzes the user request regarding communication requirements and quality of services to improve communication by reducing the distribution variance. According to various research studies, 6G networks use different frameworks and machine learning techniques to maximize resource allocation. However, a high-performance computing procedure is required to support the user's high demands. The existing methods lack high throughput and reduced latency. This research objective and novelty is addressed by applying the Confluence-Aided Process Organization Method (CAPOM) for process allocation. The proposed system reduces the latency and maximizes throughput values by utilizing High-Performance Computing (HPC) functions.

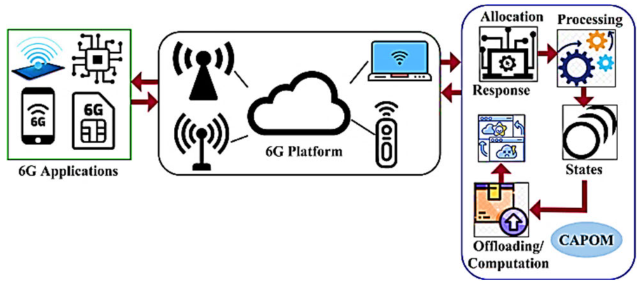


FIGURE 1. The proposed method is in 6G.

III. PROPOSED CONFLUENCE-AIDED PROCESS ORGANIZATION METHOD (CAPOM)

The Sixth-generation (6G) aids terahertz applications communication with high-performance computing since becoming unmanageable due to the high sampling support of users with less latency and high service rates of the 6G-assisted applications. The challenges in this proposed work are increased rapidity and heterogeneous application support of the user data and service distributions of different latencies. The 6G applications spanning across various domains such as UAV network (Unmanned Aerial Vehicle), eHealth remote monitoring, smart city, UM-MIMO BS (Ultra Massive-MIMO Base Station), VLC (Visible Light Communication), sCell-UE (Small Cell User Equipment), BCI (Brain Computer Interface), and AI (Artificial Intelligence), etc., require diverse application services. Therefore, regardless of the data and service distributions of the heterogeneous applications, high sampling support with high service rates and less latency is an important consideration. The proposed framework of CAPOM mainly focuses on this consideration by leveraging the overall development of the process allocations and completions through system state management. In this manuscript, latency and service rates are administrable for the applications, and their processing with the available system states application. Fig. 1 illustrates the proposed method in a 6G platform.

The 6G platforms access their process and services through requests and responses using the 6G applications. The CAPOM model functions between the user and applications. In this method, process allocations and offloading/completion for the available system states and HPC is ease for succeeding service rate outcomes for the diverse applications and users (Fig. 1). Furthermore, the design goal of this model is to minimize the completion time to reduce the waiting latency of dense applications and **to** maximize the available system states. The proposed method functions in two forms: process allocation and offloading/completion occur concurrently. The service allocation process varies for denser and non-denser service distributions to handle the diverse density of the users/applications. Then the notations utilized in this work is illustrated in Table 1.

TABLE 1. Notation description.

Notation	Description
θ_i	Process allocations
S_{R_q}	Service Request
S_{R_s},	Service Response
t	Time
WT_j	Waiting time
T_{R_q}	Request time
T_{R_s}	Response time
CT	Completion time
φ_n	Offloading
PT	Processing time
C_m	Capacity
ρ_{P_s}	Probability of allocating process
ρ_f	First instance of process allocation
$\boldsymbol{m}$	Stable probability
$A_o(\gamma)$	State allocation function
γ	Waiting latency
φ_n	Required sequence

The initial functions of the 6G platform service management technology is the main goal as stated in equation (1)"

$$\max_{i\in t} \theta_n \forall S_{R_q} = S_{R_s} \text{ and } \min_{j\in R_s} WT_j \; \forall r_q \tag{1}$$

As per equation (1), the variables $\theta_i, S_{R_q}, S_{R_s}, t$ denotes the process allocation of n^{th} service t, request, and responses, respectively. In the next consequence representation, the variables WT_j, T_{R_q}, and T_{R_s} and CT represent waiting time, request time, response time, and completion time, respectively. Here, the waiting time is computed as $WT_j = T_{R_q} - T_{R_s}; \min_{i\in t} \varphi_n \forall i \in R_q$. The next instance of minimizing the offloading/completion is denoted using the variable $\varphi_n \forall i \in R_q$. If $A = \{1, 2, \ldots, A\}$ denotes the state of the user/application, then the number of services in the processing time (PT) is $R_q \times t$, whereas the 6G assisted application request is $A \times R_q$. In the following overall request of $A \times R_q$, $R_q \times t$ are the acceptable services for waiting. Process allocation and states offloading processes are reliable using latency and density of the upcoming request of the 6G applications. In this sequence, the distribution of state and remaining services is essential to identify the non-denser application in additional services. The demanding application is the capacity (C_m) of the m state applications, the remaining time needed for process completion/offloading is the using metrics for increasing services rate and distribution ratio. The process allocation of the states assigning for the existing m is performable using federated learning. Then, depending upon the service distribution, the process allocation states are the increasing factor. For this implication, the process allocation, distribution, and completion is the prevailing sequence for deriving various conditions. The process allocation of

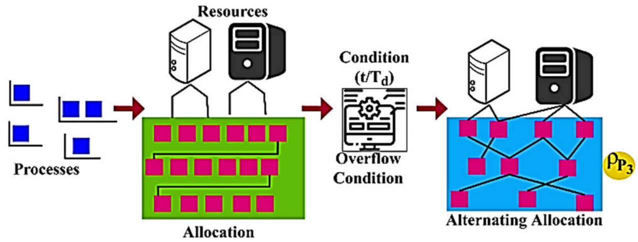


FIGURE 2. Alternative allocation process.

services and the available states for allocation are necessary for the following section.

CASE 1: Initial allocation of process

SOLUTION 1: In this process allocation, the service distribution of $\boldsymbol{t}\times\boldsymbol{R_q}$ for all $\boldsymbol{m}$ based on C_m is the considering factor. The probability of allocating process $(\boldsymbol{\rho_{P_s}})$ in a consequence, the manner is given in equation (2) as

$$\rho_p = \left(1 - \rho_f\right)^{n-1} \forall n \in t \tag{2}$$

In equation (2), variable $\boldsymbol{\rho_f}$ represents the first instance of process allocation, which is computed by using the request $\boldsymbol{R_q} \in \boldsymbol{m}$ and $\boldsymbol{R_q} \in \boldsymbol{t}$; $\boldsymbol{\rho_f} = \left(1 - \frac{\boldsymbol{R_q}\in\boldsymbol{m}}{\boldsymbol{R_q}\in\boldsymbol{t}}\right)$. The sequential service distribution follows the stable probability of $\boldsymbol{m}$ such that there is no remaining process, and it is estimated as in the above equation (1). Hence, the allocation of processes (***PA***) for $\boldsymbol{\rho_{P_s}}$ is as follows

$$PA(m) = \frac{1}{\left|R_s - R_q + 1\right|} \cdot \left(\rho_{P_s}\right)_i, \quad if \; \forall n \in t \tag{3}$$

Although, the process allocation for m as in equation (3), is valid for both $\left(A \times R_q\right)$, $\left(t \times R_q\right)$ make certain waiting time service distributions. The gathering services of assigning t to decrease the impact of the functions $\left(A \times R_q\right) > \left(t \times R_q\right)$, the process allocation is illustrative using the data and service distributions. Hence, the planning-based conditions of $A > t$ and ρ_f is less to satisfy the above-derived equation (1). The modified solution in this case 1 is the extending ρ_f and therefore, the waiting time outputs in the waiting latency of dense application demands.

CASE 2: Alternative process allocation

SOLUTION 2: In this Alternative process allocation, the unstable condition of $\boldsymbol{A} > \boldsymbol{t}$ is high, and therefore the service distribution and allocation of the process is an unchangeable time sequence. In Fig. 2, the alternative allocation process is illustrated.

The allocation is performed for the overflow condition identified using t/T_d wherein the available processes are verified for their sequences. The process sustainability is verified in the alternating sequence and first state allocation; hence, the allocations are prevented from overflowing. However, for the first allocation, if an overflow is experienced, the allocations are alternating for precise resource distribution (Fig. 2). Along with the stable time of m, the denser application and remaining states are the considered processes. The probability of a sequential process of allocation $\left(\rho_{P_n}\right)$ is given as in equation (4)

$$\rho_{P_n} = \frac{\rho_{T_s} \times PA(m) \times \left[\left(R_s - R_q\right)\rho_f - \left(\frac{R_s - R_q}{m}\right)\frac{WT}{T_{R_q}}\right] A_o(\gamma)}{A_o(\gamma) \times m} \tag{4}$$

In equation (4), $A_o(\gamma)$ factor represents the state allocation function for t, request, and response are represented as the R_q, R_s and the waiting time is WT. The sequential process allocation is computed concerning the weight time of the services, which is defined as the $A_o(\gamma) = \int_0^t WT^{t-1}(1-WT)^{t-1}\,dt(WT)$. The allocated resources belong to the process allocation that is defined as the $A_o(\gamma) \in PA(m) = \int_1^{R_q} WT^{t-1}.\frac{\rho_f}{T_{R_q}}\left(1-\rho_{P_s}\right)^{t-1} dt\left(R_q\right)$. In this state allocation process, the unstable assigning services to the m is a dense application issue. As mentioned above, the process allocation requires more waiting time completion time, thereby increasing the processing capacity. From the above-determined cases 1 and 2, the process allocation and completions of application demands based on $A > t$ and m dense application and waiting time are the identifiable constraints. In particular, these constraints are noticed using federated learning to alleviate the problems through the learning process. The following session illustrates the processing capacity for the offloading/completion process to alleviate the defining problems.

IV. OFFLOADING/COMPLETION USING PROCESS CAPACITY

The definition of the processing capacity of the completion process is based on federated learning. The federated learning is one of the effective machine learning techniques used to train the data to improve system efficiency. It aids application support with less latency and high service rates. The above-discussed case 1 and 2 allocation processes joint with the resolving sequences using federated learning. The service distribution process depends on various metrics for identifying the dense application and waiting latencies during service distribution. Therefore, the conditions for service distribution differ, which follows the process through processing capacity. The processing capacity is prescribed for both case 1 and case 2 by computing the m available probability and allocation of states for planning time. The first state allocation (SA) relies on maximum processing capacity (P_c) and $A_o(\gamma)$ is estimated as

$$A_o(\gamma, P_c) = \left[R_s - \left(\frac{WT}{T_{R_q}}\right) \times \frac{1}{m}\right] - SA(m) + 1 \tag{5}$$

In equation (5), CT denotes the completion time of the allocation and the processing capacity depending allocation of the states for case 1, as in ρ_{P_s} and state allocation (m). Here, the m is computed as the $m = \sum_{i \in t} State\ Allocation\,(m)_i - \left(\rho_{P_n}\right)_i$. Now, the chances of performing alternative allocations sequentially are

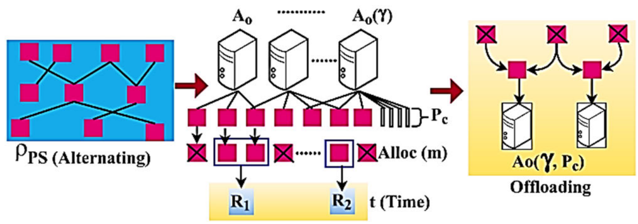


FIGURE 3. Offloading process illustration.

$$\rho_{P_s}\left(\frac{t}{T_d}\right) = \frac{1}{\sqrt{2m\gamma^2}} expersion\left[-\frac{R_q - \rho_f \times R_s}{\gamma}\right] \tag{6}$$

In equation (6), alternative resource allocation is computed with waiting latency γ, response, and request concerning m. Therefore, the waiting latency is estimated as $\gamma = R_q - \rho_f * m$. As per the above equation (6), the main goal is to exploit offloading jointly and shared processing A and t to increase the service distribution and reduce the waiting time and hence, the actual R_s is computed as

$$R_s = max\left[\frac{\rho_{P_s} \times R_q}{SA(m) - \rho_f * R_q}\right] \tag{7}$$

In equation (7), the difference is $\left[1 - \frac{\rho_{P_s}}{SA(m) - \rho_f * R_q}\right]$ and this alternative allocation is the waiting time-dense application instances of R_q. The sharing R_q is $\left[R_q * A_o(\gamma, P_c)\right]$ is the φ_n requiring sequences, and therefore the waiting latency is demandingly increased. The ranges for increasing waiting latencies as per the above equation (6), the range is derived as $min\ possibleR_s = \gamma = R_q - \rho_f * m$ and max possible is equating RHS of equations (2) and (6). the equating process is defined in equation (8).

$$\left.\begin{aligned} \left(1-\rho_f\right)^{n-1} &= \frac{-R_s + \rho_f R_q}{\sqrt{2\pi}\gamma^2} \forall n \in t \\ R_s &= \rho_f R_q - \left(1-\rho_f\right)^{n-1}\sqrt{2\pi}\gamma^2 \end{aligned}\right\} \tag{8}$$

In equation (8), the range of waiting latency denser applications is either of r_s or γ, in both above-derived cases, if $\rho_f = 0$, then $\gamma = R_q = R_s$ is the maximum possible that is defined as $R_s = \sqrt{2\pi}\left(R_q\right)^2 (min)$. and if $\rho_f = 1$, $R_s = R_q - m/R_s = R_q$ that is defined as $R_s = R_q\,(max)$. Hence, they take the place of $R_q = R_s$ is a feasible solution, and then, the waiting latency for all $i \in t$ and $j \in R_s$ in the above equation. Fig. 3 presents the offloading process illustration.

In the alternate allocation process, the A_0 to $A_0(\gamma)$ Instances are used for identifying resource distribution. This is based on the actual capacity and the available intervals for preventing further waiting time. This is stabilized based on the allocation time and the sequence required for m. Contrarily, the allocations at t are offloaded if the capacity exceeds the limit of the available resources (Fig. 3). The service distribution in this framework is all the existing m, where the allocation requests and responses are shared processing. Therefore, the waiting time is compact, as in the above equation (1). The offloading/completion ranges are $(R_q - \rho_f * m)$ and $\sqrt{2\pi\gamma}\,(R_q)^2$, which defines the process allocation and waiting time, along with completion time, for the sequence R_q. The completion process of $(R_q - \rho_f * m)$ and $\sqrt{2\pi\gamma}\,(R_q)^2$ from the existing $t \in R_q$ is illustrated in the following.

The offloading/completion-based processing capacity (R_q, R_s) and $(R_{s_{t-1}}, R_{s_t})$ depend on the available t from the responses. The probability of ρ_f and ρ_{P_s} and ρ_{P_n} is the deciding factor for both offloading and response. The offloading of process capacity takes place in (R_q, R_s) and $(R_{s_{t-1}}, R_{s_t})$ is sharing based on T_d for $A_o(\gamma)$ is given as

$$Allocation(m) = \begin{cases} \dfrac{m - (p_f * R_q)}{m + (\rho_{P_s})\,R_q}, & \forall R_s = R_q \\ \dfrac{m - (p_f * R_q)}{m + (\rho_{P_s} + \rho_{P_n} - \rho_f)\,R_q}, & \forall R_s < R_q \end{cases} \tag{9}$$

From equation (9), the offloading sequences of $(\rho_{P_s} + \rho_{P_n} - \rho_f)$ is found using the allocation of states (m). Therefore, the available m forces the rest of the responses for offloading/completion, the remaining states until the next et. This process is estimated as

$$\left.\begin{aligned} \frac{m}{m+R_q} &= \frac{1}{(R_s - R_q + 1)} \\ m + R_q &= mR_s - mR_q + m \\ R_s &= \frac{(m+1)R_q}{m} \end{aligned}\right\} \tag{10a}$$

The remaining state $R_s \forall T \in R_q$ is as estimated using the above equation (10a), and therefore, the alternative allocations are essential for allocating the remaining R_q.

$$\begin{aligned} m - \frac{R_q}{m + R_q} &= \frac{1}{(R_s - R_q + 1)}, \text{ as } \rho_f = 0,\ \rho_{p_s} \\ &= \rho_{P_n} = 1 \end{aligned} \tag{10b}$$

In equation (10b), the alternative allocation process performed by considering the request and response is defined as $R_S = \frac{2R_q + mR_q + R_q^2}{(m - R_q)}$. It has been further defined as the $mR_s - R_qR_s = R_q + R_q + R_q^2 + mR_q$. In this offloading condition, m or $\left(m - \frac{R_q}{\gamma}\right)$ is the service distribution irrespective of the users and applications. In the alternative sequence of state allocation, minimizing the response is discussed to reduce dense application and waiting latency. Fig. 4 presents the state representative for the offloading and completion process.

The states are determined using the alternating and resource distribution states estimated for $A_0(\gamma)$ and R_s. This is validated for $Alloc(m)$ and ρ_s such that the wait time is not increased further. The wait time-induced processes are offloaded for further allocation, whereas the response time-induced requests are termed as completed. This process is independently based on the federated stated for which the

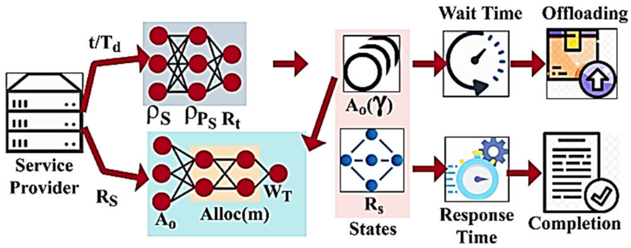


FIGURE 4. State representation–completion/ offloading.

different states are assessed (Fig. 4). The service distribution is the process that follows either of the R_s as in the above equation. It varies for both the R_s as the initial stage and no more m whereas the alternative sequences of states reallocation as $(m - R_q)$ is the retaining process. From the discussion mentioned above in the allocation of states for $\varphi_n \in R_s = \frac{(m+1)R_q}{m}$ is reliable, and it does not require waiting time. The completion time (CT) of a t in this process allocation is the considering metric, and it differs for each m depending on the processing capacity (P_c). This completion time (CT) using equation (11) for R_s in equation (10b)

$$CT = \begin{cases} \dfrac{P_c}{Alloc(m)}, \forall R_s = R_q \text{ or } R_s = \dfrac{(m+1)R_q}{m}, \\ \quad if\ r_s < r_q \\ \dfrac{P_c}{Alloc(m)} + \dfrac{A_o(\gamma, T_d)\,(\rho_{P_s} + \rho_{P_n} - \rho_f)}{Alloc(N)}, \\ \quad if\ \forall R_s = \dfrac{2R_q + mR_q + R_q^2}{(N - r_q)} \end{cases} \tag{11}$$

As in equation (11), $CT \in [T_{R_q}, T_{R_s}]$ and the final solution of CT (i.e.) $(CT * R_q)$ is the maximum et_n and service rate is increased for handling $(m - R_q)$ requests. Hence, the process allocation of all $t \in R_q$ increases both φ_n and $et_n \forall i \in R_q$. This sequential process allocation reduces the waiting latencies and completion time and increases the distribution ratio and processing rate. The proposed model parameters are evaluated using an experimental setup consisting of a computer running MATLAB 2016 simulation software with an Intel i7 Quad-core 3.2 GHz CPU and 64 GB RAM. Fig. 5 presents the analysis of allocation/ instance and its type for different requests.

Fig. 5 presents the analysis of allocation/ instance and its type for different requests. The target is to achieve less wait time for process $alloc\,(m)$, wherein ρ_{p_n} is required for distinguishing R_S. If the $alloc\,(m)$ is interrupted by R_t as in equation (10a), then $A_o(\gamma, P_C)$ relies on the next sequence. This results in offloading as required for $\frac{t}{T_d}$. Contrarily the sequence is split as $\theta_i \forall i \in t$ in ρ_{P_S} and hence completion is achieved. Therefore, the allocations (less) experience a constant allocation regardless of the alternating sequences. This is common for varying processes observed in S_{R_q} before $\left(\frac{t}{T_d}\right)$. An analysis for waiting, completion times, and states are presented in Fig. 6 for different allocations/instances in Fig. 6.

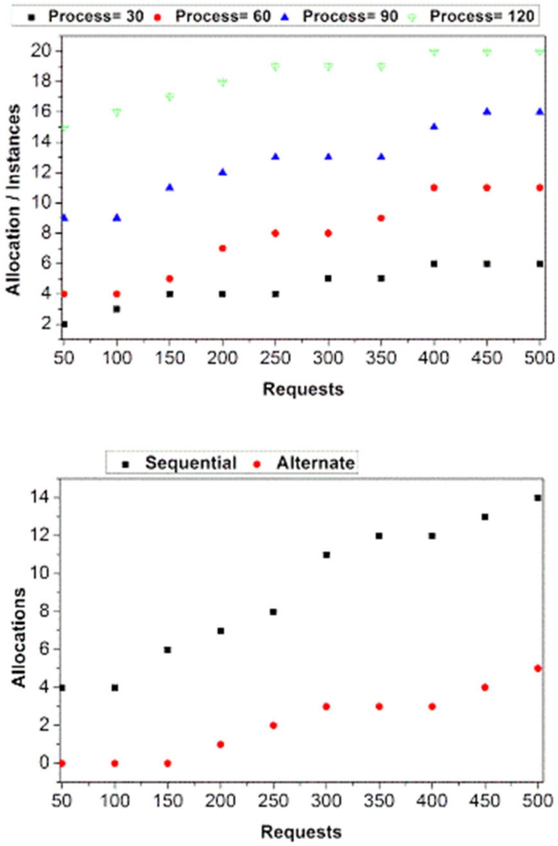


FIGURE 5. Allocation/ instance and its type for different requests.

In Fig. 6, the waiting, completion time, and the states observed for different allocations/instances are analyzed. The ρ_{P_n} is required to validate $A_o(\gamma)$ under $alloc(m)$. This is divided using γ and R_s states that retain t alternating allocations without reducing R_s validations. In contrast to the allocations, ρ_{P_s} is required for ρ_f identification and hence, ρ_f Induced allocations are completed with fewer time intervals. Therefore, the change in allocations is performed without R_S hindrance and, therefore, $\rho_{P_S}\left(\frac{t}{T_d}\right)$ is mandatory for reducing $WT_j \forall j \in CT$. Hence the ρ_f is observed from the consecutive sequences without increasing waiting allocations. Fig. 7 presents an analysis of completion time under different states.

An analysis of completion time for different states and processes is represented in Fig. 7. As the processes are fewer, $\rho_f \in \rho_{P_S}$ such that wait time is less and hence CT is less. If $A_o(\gamma)$ is not satisfied under ρ_f, then $\rho_{P_S}\left(\frac{t}{T_d}\right)$ is verified for $alloc(m)$ and hence (R_t) is estimated. If it does not meet the R_s, then ρ_{P_n} is validated where the wait time is high, hence the completion time. By deciding $A(\gamma)$ or $A(P_C)$, the consecutive allocation is validated without an increase in completion time.

V. DISCUSSION

The suggested technique's performance is examined using MATLAB experiments, considering 70 users sharing a

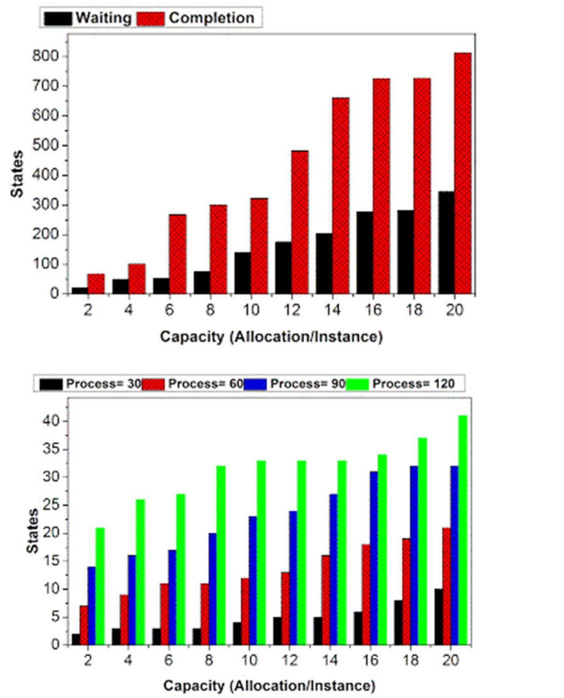


FIGURE 6. Waiting, completion times, and states for different allocation/ instances.

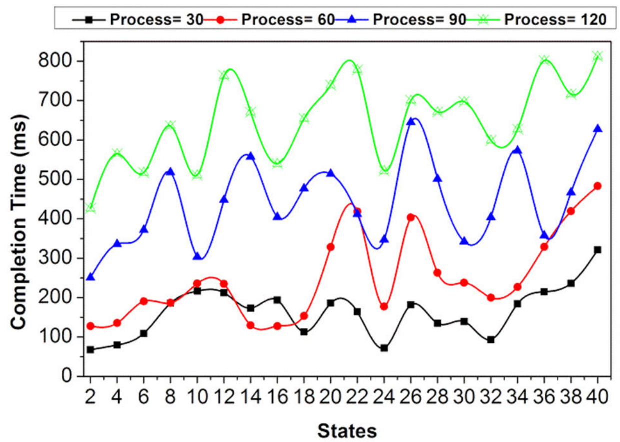


FIGURE 7. Completion time under different states.

common 6G resource. The service demands are processed and met using 9 service providers capable of handling 20 processes. A total of 140 processes are considered for validating the performance. If an active process is offloaded after 240ms of wait time; hence, a new allocation is preferred. This setup considers the metrics of distribution ratio, latency, waiting time, and processing rate for analysis. The methods CAMUO [28], ACDRA [17], and PCTO [25] are considered in this analysis.

A. DISTRIBUTION RATIO

The distribution ratio is high in the suggested technique is great compared to the other factors (Refer to Fig. 8). In this framework, θ_i, S_{R_q} and S_{R_s} are the allocation of the process

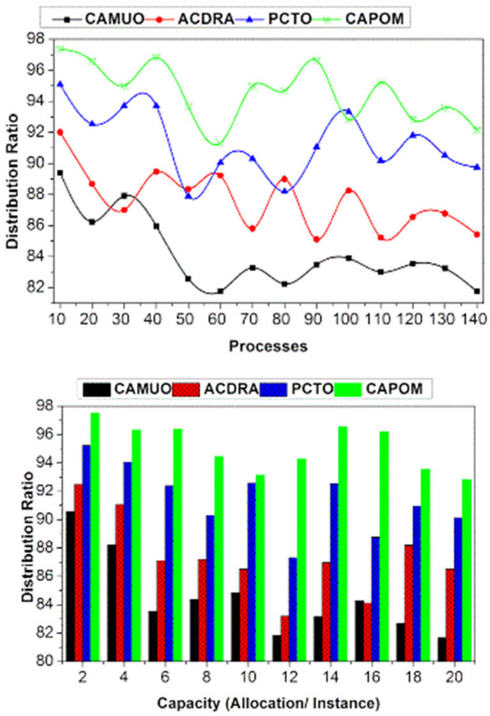


FIGURE 8. Distribution ratio analysis.

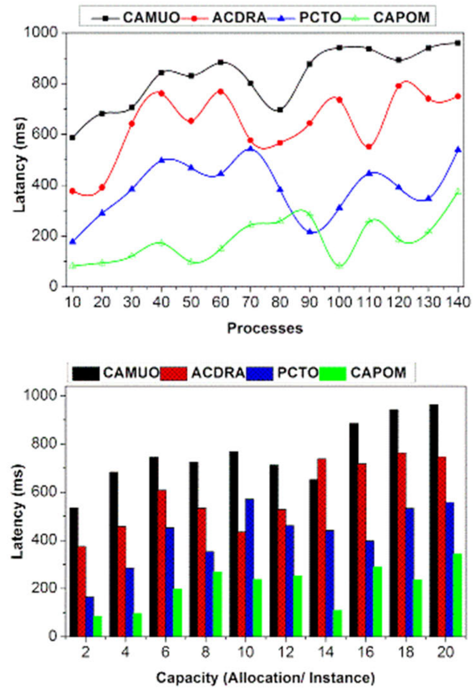


FIGURE 9. Latency analysis.

for identifying CT. If case 1 occurs for improving R_s based on $\varphi_n \forall i \in R_q$ [as in equation (1)], then $A \times R_q$ and $R_q \times t$ are acceptable states is computed. Based on this implication, A is determined. The Utmost ρ_f due to ρ_{P_s} and ρ_{P_n} dense applications are considered. This consideration requires high service distributions, preventing multiple $A_o(\gamma)$ process allocation and modifications. Hence, the offloading/completion to the user or application is administered as derived in equation (5) with WT consideration. The first state allocation is performed; in the alternative allocations are estimated for which the $\left[\left(R_s - R_q\right)\rho_f - \left(\frac{R_s - R_q}{m}\right)\frac{WT}{T_{R_q}}\right]$ is alone validated. In this condition, the change in processing capacity in $\frac{-R_s + \rho_f R_q}{\sqrt{2\pi\gamma^2}}$ and its existing sequence $A > t$ are mutually shared processing. This process helps in preventing additional waiting latency, as mentioned above. Therefore, for the $A > t$, the θ_i validation, improving the existing state, the distribution ratio under discrete sequences is high. Based on the multiple allocations, m is estimated that combine t such that $\frac{WT}{T_{R_q}}$ is presented. In the proposed framework, the processing relies on $\left(R_s - R_q\right)\rho_f$ and hence the γ change waiting latencies are considerably less.

B. LATENCY

The proposed framework process waits for latency and completion time as it does not provide sequential process allocation for 6G-assisted applications. The alternative allocations are the demands and high sampling support $\frac{(m+1)R_q}{m}$ performed for $\varphi_n \in R_s$ in different φ. This impact is addressed using the dense application $A_o(\gamma)$ demands, preventing computation failures. The two different cases $A \times R_q$ and $R_q \times t$ are analyzed without augmenting the service distributions. Similarly, the $(m+1)$ based state allocation requires φ_n and R_s computation for occupying additional process allocation. The processing capacity sequence from R_{s_t} to $R_{s_{t-1}}$ be performed for different $\left(R_s - R_q\right)\rho_f$ validations, preventing extra service necessities. The γ performed service distributions, and $A > t$ described demands are detached for further alternative process allocations, averting completion time and waiting latency. This proposed framework performs further process allocations and service distributions based on $\left(R_q, R_s\right)$ for which a processing capacity $\rho_f R_q$ is given. This is common for θ_i, S_{R_q} and S_{R_s} for which the framework attains less latency, as offered in Fig. 9.

C. WAITING TIME

The proposed framework needs less waiting time compared to the other factors. There are two prime cases for less waiting time in the suggested framework. First, the process allocation instance based on $\left(R_q, R_s\right)$ and $\left(R_{s_{t-1}}, R_{s_t}\right)$ is perceived as determining additional process allocation θ_i. This service distribution augments the waiting time regardless of discrete $\frac{(m+1)R_q}{m}$ preventing $A_o(\gamma)$ waiting latencies. In the contradictory process allocation, $2R_q + mR_q + R_q^2$ being the next reason identified for alternate process allocation. For the above-discussed cases, the waiting time is great due to $\varphi_n \in R_s$ and prolonged m. To decrease this completion time factor, $\left(R_q, R_s\right)$ to $\left(R_{s_{t-1}}, R_{s_t}\right)$ under discrete γ and A is repeatedly processed for the accessible state's allocation. The allocation

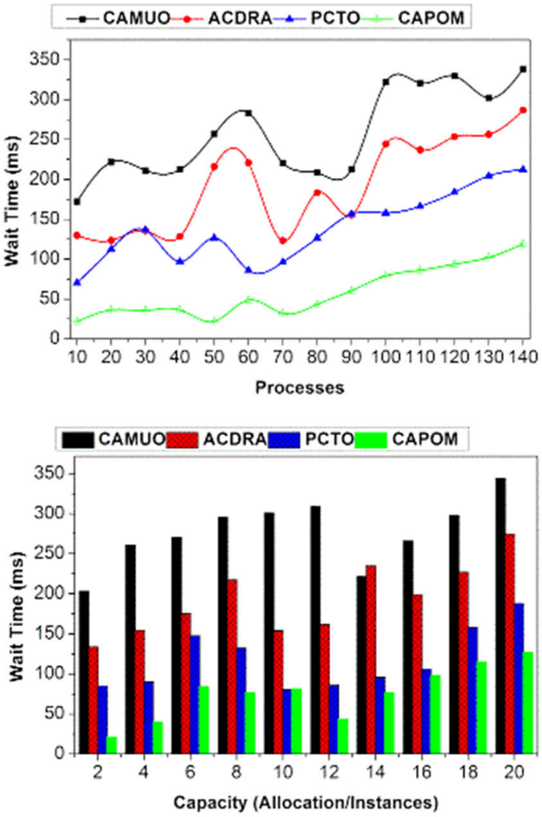


FIGURE 10. Waiting time analysis.

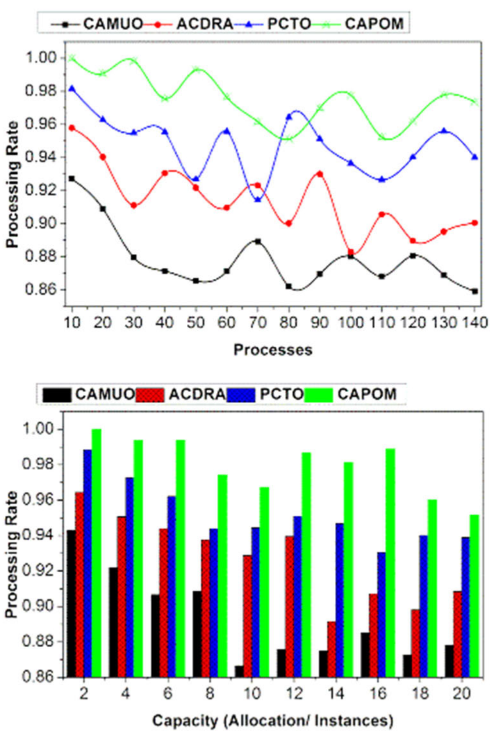


FIGURE 11. Processing rate analysis.

of the process is modeled based on the discrete scenario. The proposed framework distinguishes the process capacity equation (3) from equations (10) and (11) for a sequence of allocations. The validation process limits the need for completion time, preventing extra time. Therefore, the waiting time for various users/applications and service distributions is reduced for high-performance computing, as illustrated in Fig. 10.

TABLE 2. Comparative analysis for processes.

Metrics	CAMUO	ACDRA	PCTO	CAPOM
Distribution Ratio	81.75	85.42	89.75	92.143
Latency (ms)	959.76	750.65	539.61	374.606
Wait time (ms)	338.41	286.67	212.11	119.179
Processing Rate	0.8592	0.9005	0.9401	0.9735

TABLE 3. Comparative analysis of capacity.

Metrics	CAMUO	ACDRA	PCTO	CAPOM
Distribution Ratio	81.69	86.53	90.13	92.847
Latency (ms)	963.03	746.84	558.32	343.377
Wait time (ms)	344.14	274.56	187.91	127.586
Processing Rate	0.878	0.9085	0.9388	0.9518

D. PROCESSING RATE

This proposed framework achieves a high processing rate for various process allocations and service distributions (Refer to Fig. 11). The waiting latency is alleviated based on $\frac{1}{|R_s - R_q + 1|}$ conditions for leveraging process allocation through high-performance computing. The T_{R_s} and CT based allocation of the process using previous and current system states performance $\frac{m-(\rho_f * R_q)}{m+(\rho_{P_s})R_q}$ in identifying the waiting latency in (R_q, R_s) to $(R_{s_{t-1}}, R_{s_t})$ instances. Further, the $\frac{2R_q + mR_q + R_q^2}{(m - R_q)}$ is performed for increasing the processing rate beyond the prolonged $A > t$ and hence the $A_o(\gamma)$ is increased. In the different allocation of states, the WT is performed for detecting waiting time in A as in equation (8). Therefore, (R_q, R_s) to $(R_{s_{t-1}}, R_{s_t})$ be modified depending on $A > t$; this process allocation has to satisfy two distinct cases for retaining the processing rate. First state (R_q, R_s) in both S_{R_q} and S_{R_s} such that $A_o(\gamma)$ is retained. As per the retained case m, R_q is functioned based on $[-R_s + \rho_f R_q]$ and therefore, the $A > t$ is satisfied. If this condition is satisfied, processing capacity is increased to reduce the waiting latencies. In the alternative process allocations, $(R_{s_{t-1}}, R_{s_t})$ The process allocation and offloading based on service distributions are defined. In this proposed scheme, the defined γ is aided for m and R_q validation for improving the process allocation. This leads to further alternative process allocation in the assisted applications and t. Tables 2 and 3 provide the comparative analysis summary for the above discussion.

Inference: The proposed method maximizes the distribution ratio and processing rate by 6.5% and 7.36%, respectively. In order, it reduce the latency and waits time by 8.34% and 9.55%

Inference: The proposed CAPOM achieves a 6.73% high distribution ratio, 9.09% less latency, 8.76% less wait time, and 8.67% high processing rate.

VI. CONCLUSION

A confluence-aided process organization method using high-performance computing in 6G service processing is presented in this article. The service demand to resource allocation process consists of completion and offloading to identify backlogs. The admitted and offloaded processes are independently classified based on first and alternating sequences. Based on the waiting time and allocation probability, the states are updated to improve service distribution. The resource capacity and its corresponding completion time are accounted for for ease of processing and allocation, which modifies the current state. Based on classified independent states, the allocations and processing are performed, reducing the latency for multi-process distributions. The proposed CAPOM achieves a 6.73% higher distribution ratio for various capacities, a 9.09% reduction in latency, an 8.76% reduction in wait time, and an 8.67% higher processing rate.

ACKNOWLEDGMENT

The authors would like to thank the Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM), for their support in this research.

REFERENCES

[1] J. Cao, W. Feng, N. Ge, and J. Lu, "Delay characterization of mobile-edge computing for 6G time-sensitive services," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3758–3773, Mar. 2021.

[2] D. Spatharakis, I. Dimolitsas, D. Dechouniotis, G. Papathanail, I. Fotoglou, P. Papadimitriou, and S. Papavassiliou, "A scalable edge computing architecture enabling smart offloading for location based services," *Pervasive Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101217.

[3] M. Idrees, M. M. Maqbool, M. K. Bhatti, M. M. U. Rahman, R. Hafiz, and M. Shafique, "An approximate-computing empowered green 6G downlink," *Phys. Commun.*, vol. 49, Dec. 2021, Art. no. 101444.

[4] K. Wang, X. Wang, and X. Liu, "A high reliable computing offloading strategy using deep reinforcement learning for IoVs in edge computing," *J. Grid Comput.*, vol. 19, no. 2, pp. 1–15, Jun. 2021.

[5] L. Zhang, S. Zhuge, Y. Wang, H. Xu, and E. Sun, "Energy-delay tradeoff for virtual machine placement in virtualized multi-access edge computing: A two-sided matching approach," *Pervasive Mobile Comput.*, vol. 14, pp. 6603–6621, Jun. 2023.

[6] T. K. Rodrigues, J. Liu, and N. Kato, "Application of cybertwin for offloading in mobile multiaccess edge computing for 6G networks," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16231–16242, Nov. 2021.

[7] Z. Zhou, M. Shojafar, J. Abawajy, and A. K. Bashir, "IADE: An improved differential evolution algorithm to preserve sustainability in a 6G network," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 1747–1760, Dec. 2021.

[8] T. R. Raddo, S. Rommel, B. Cimoli, C. Vagionas, D. Perez-Galacho, E. Pikasis, E. Grivas, K. Ntontin, M. Katsikis, D. Kritharidis, E. Ruggeri, I. Spaleniak, M. Dubov, D. Klonidis, G. Kalfas, S. Sales, N. Pleros, and I. T. Monroy, "Transition technologies towards 6G networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–22, 2021.

[9] Y. Wei, M. Peng, and Y. Liu, "Intent-based networks for 6G: Insights and challenges," *Digit. Commun. Netw.*, vol. 6, no. 3, pp. 270–280, Aug. 2020.

[10] H. Li, F. Fang, and Z. Ding, "Joint resource allocation for hybrid NOMA-assisted MEC in 6G networks," *Digit. Commun. Netw.*, vol. 6, no. 3, pp. 241–252, Aug. 2020.

[11] T. K. Rodrigues, K. Suto, and N. Kato, "Edge cloud server deployment with transmission power control through machine learning for 6G Internet of Things," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 2099–2108, Oct. 2021.

[12] R. Pugliese, S. Regondi, and R. Marini, "Machine learning-based approach: Global trends, research directions, and regulatory standpoints," *Data Sci. Manage.*, vol. 4, pp. 19–29, Dec. 2021.

[13] A. Shakarami, A. Shahidinejad, and M. Ghobaei-Arani, "An autonomous computation offloading strategy in mobile edge computing: A deep learning-based hybrid approach," *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021, Art. no. 102974.

[14] S. Tang, W. Zhou, L. Chen, L. Lai, J. Xia, and L. Fan, "Battery-constrained federated edge learning in UAV-enabled IoT for B5G/6G networks," *Phys. Commun.*, vol. 47, Aug. 2021, Art. no. 101381.

[15] A. Mekrache, A. Bradai, E. Moulay, and S. Dawaliby, "Deep reinforcement learning techniques for vehicular networks: Recent advances and future trends towards 6G," *Veh. Commun.*, vol. 33, Jan. 2022, Art. no. 100398.

[16] J. Wang and Y. Zhang, "Using cloud computing platform of 6G IoT in e-commerce personalized recommendation," *Int. J. Syst. Assurance Eng. Manage.*, vol. 12, pp. 654–666, Jan. 2021.

[17] C. X. Mavromoustakis, G. Mastorakis, and J. M. Batalla, "A mobile edge computing model enabling efficient computation offload-aware energy conservation," *IEEE Access*, vol. 7, pp. 102295–102303, 2019.

[18] K. Lin, Y. Li, Q. Zhang, and G. Fortino, "AI-driven collaborative resource allocation for task execution in 6G-enabled massive IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5264–5273, Apr. 2021.

[19] G. Yoon, G.-Y. Kim, H. Yoo, S. C. Kim, and R. Kim, "Implementing practical DNN-based object detection offloading decision for maximizing detection performance of mobile edge devices," *IEEE Access*, vol. 9, pp. 140199–140211, 2021.

[20] H. Xu, P. V. Klaine, O. Onireti, B. Cao, M. Imran, and L. Zhang, "Blockchain-enabled resource management and sharing for 6G communications," *Digit. Commun. Netw.*, vol. 6, no. 3, pp. 261–269, Aug. 2020.

[21] P. Yan and S. Choudhury, "Deep Q-learning enabled joint optimization of mobile edge computing multi-level task offloading," *Comput. Commun.*, vol. 180, pp. 271–283, Dec. 2021.

[22] S. Khan, A. Hussain, S. Nazir, F. Khan, A. Oad, and M. D. Alshehri, "Efficient and reliable hybrid deep learning-enabled model for congestion control in 5G/6G networks," *Comput. Commun.*, vol. 182, pp. 31–40, Jan. 2022.

[23] Y. Zhang and J. Fu, "Energy-efficient computation offloading strategy with tasks scheduling in edge computing," *Wireless Netw.*, vol. 27, no. 1, pp. 609–620, Jan. 2021.

[24] A. Mukherjee, N. Dey, A. Mondal, D. De, and R. G. Crespo, "iSocialDrone: QoS aware MQTT middleware for social Internet of Drone Things in 6G-SDN slice," *Soft Comput.*, vol. 27, pp. 5119–5135, Aug. 2021.

[25] A. Naouri, H. Wu, N. A. Nouri, S. Dhelim, and H. Ning, "A novel framework for mobile-edge computing by optimizing task offloading," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 13065–13076, Aug. 2021.

[26] F. Alqahtani, M. Al-Maitah, and O. Elshakankiry, "A proactive caching and offloading technique using machine learning for mobile edge computing users," *Comput. Commun.*, vol. 181, pp. 224–235, Jan. 2022.

[27] K. Chakrabarti, "Deep learning based offloading for mobile augmented reality application in 6G," *Comput. Electr. Eng.*, vol. 95, Oct. 2021, Art. no. 107381.

[28] K. Guo and R. Zhang, "Fairness-oriented computation offloading for cloud-assisted edge computing," *Future Gener. Comput. Syst.*, vol. 128, pp. 132–141, Mar. 2022.

[29] A. Shahidinejad, F. Farahbakhsh, M. Ghobaei-Arani, M. H. Malik, and T. Anwar, "Context-aware multi-user offloading in mobile edge computing: A federated learning-based approach," *J. Grid Comput.*, vol. 19, no. 2, pp. 1–23, Jun. 2021.

[30] Z. Chen and X. Wang, "Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, pp. 1–21, Dec. 2020.

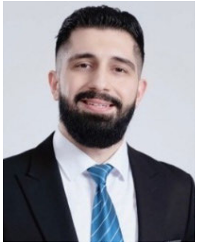
[31] E. B. Ali, S. Kishk, and E. H. Abdelhay, "Multi-device multi-task computation offloading in device to device communication," *Wireless Pers. Commun.*, vol. 123, pp. 1883–1896, Mar. 2022.

[32] A. B. Kathole, J. Katti, D. Dhabliya, V. Deshpande, A. S. Rajawat, S. B. Goyal, M. S. Raboaca, T. C. Mihaltan, C. Verma, and G. Suciu, "Energy-aware UAV based on blockchain model using IoE application in 6G network-driven cybertwin," *Energies*, vol. 15, no. 21, p. 8304, Nov. 2022.

[33] J. Chen, B. Qian, Y. Xu, H. Zhou, and X. S. Shen, ''Towards user-centric resource allocation for 6G: An economic perspective,'' *IEEE Netw.*, vol. 37, no. 2, pp. 254–261, Mar./Apr. 2023.
[34] H. Alsulami, S. H. Serbaya, E. H. Abualsauod, A. M. Othman, A. Rizwan, and A. Jalali, ''A federated deep learning empowered resource management method to optimize 5G and 6G quality of services (QoS),'' *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–9, Mar. 2022.
[35] F. Dong, F. Liu, Y. Cui, W. Wang, K. Han, and Z. Wang, ''Sensing as a service in 6G perceptive networks: A unified framework for ISAC resource allocation,'' *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3522–3536, May 2023.
[36] Q. Guo, F. Tang, and N. Kato, ''Federated reinforcement learning-based resource allocation in D2D-enabled 6G,'' *IEEE Netw.*, early access, Sep. 5, 2022, doi: 10.1109/MNET.122.2200102.
[37] M. Sheng, D. Zhou, W. Bai, J. Liu, H. Li, Y. Shi, and J. Li, ''Coverage enhancement for 6G satellite-terrestrial integrated networks: Performance metrics, constellation configuration and resource allocation,'' *Sci. China Inf. Sci.*, vol. 66, no. 3, Mar. 2023, Art. no. 130303.
[38] M. Ashwin, A. S. Alqahtani, A. Mubarakali, and B. Sivakumar, ''Efficient resource management in 6G communication networks using hybrid quantum deep learning model,'' *Comput. Electr. Eng.*, vol. 106, Mar. 2023, Art. no. 108565.
[39] A. Thantharate and C. Beard, ''ADAPTIVE6G: Adaptive resource management for network slicing architectures in current 5G and future 6G systems,'' *J. Netw. Syst. Manage.*, vol. 31, no. 1, p. 9, Jan. 2023.
[40] W. Han, X. Wang, and J. Mei, ''EDI-driven multi-dimensional resource allocation for inclusive 6G communications,'' *IEEE Netw. Lett.*, vol. 5, no. 2, pp. 110–114, Jun. 2023.

JAMIL ABEDALRAHIM JAMIL ALSAYAYDEH (Member, IEEE) received the degree in computer engineering and the M.S. degree in computer systems and networks from Zaporizhzhia National Technical University, Ukraine, in 2009 and 2010, respectively, and the Ph.D. degree in engineering sciences, with a specialization in automation of control processes, from the National Mining University, Ukraine, in 2014. Since 2015, he has been a Senior Lecturer with the Department of Electronics and Computer Engineering Technology, Faculty of Electrical and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka (UTeM). He is also a Research Member with the Center for Advanced Computing Technology. He has supervised bachelor's and master's students. His research interests include formal methods, simulation, the Internet of Things, computing technology, artificial intelligence, and machine learning: computer architecture, algorithms, and applications. He is the author/coauthor of more than 40 research publications in his research, which were cited by more than 88 documents. He actively publishes research articles and receives grants from the government and private sectors, universities, and international collaboration. He is a member of the Board of Engineers Malaysia (BEM). He is also a reviewing member of various reputed journals.

IRIANTO received the Ph.D. degree in applied and computational statistics from Universiti Putra Malaysia, in 2009 and 2021, respectively. Since graduation, he started to teach with several universities in Indonesia. From 2012 to 2021, he was a Lecturer with the Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka. In 2021, he was an Assistant Professor with American International University, Kuwait. Since 2022, he has been an Assistant Professor with Rabdan Academy and Zayed Military University, United Arab Emirates. His research interests include applied mathematics and multivariate statistical process control.



AHMED JAMAL ABDULLAH AL-GBURI received the M.Eng. and Ph.D. degrees in electronics and computer engineering (telecommunication systems) from Universiti Teknikal Malaysia Melaka (UTeM), Malaysia, in 2017 and 2021, respectively. From December 2021 to March 2023, he was a Postdoctoral Fellow with the Microwave Research Group (MRG), Faculty of Electronics and Computer Engineering, UTeM. He is currently a Senior Lecturer with the Faculty of Electrical and Electronic Engineering Technology (FTKEE), UTeM. He has authored and coauthored several journals and conference proceedings. His research interests include microwave sensors, metasurfaces, UWB antennas, array antennas, and miniaturized antennas for UWB and 5G applications. He was a recipient of the Best Paper Award from the IEEE Community and won a number of gold, silver, and bronze medals in international and local competitions.

SAFARUDIN GAZALI HERAWAN is currently a Senior Lecturer with Bina Nusantara University, Jakarta, Indonesia. His current research interests include automotive engineering, renewable energy, and heat recovery technologies. He is the author/coauthor of more than 80 research publications, which were cited by more than 290 documents.