IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 August 2023, accepted 7 September 2023, date of publication 15 September 2023,
date of current version 22 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315856

RESEARCH ARTICLE

Dual-Phase Resource Allocation Algorithm in Software-Defined Network SDN-Enabled Cloud

AMIRAH H. ALOMARI[1,2], SHAMALA K. SUBRAMANIAM[1], (Member, IEEE), NORMALIA SAMIAN[1], (Member, IEEE), ROHAYA LATIP[1], (Member, IEEE), AND ZURIATI AHMED ZUKARNAIN[1], (Member, IEEE)

[1]Department of Communication Technology and Networking, Faculty of Computer Science, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia
[2]Faculty of Computer Science, King Khalid University, Abha 62529, Saudi Arabia

Corresponding authors: Amirah H. Alomari (aalamre@kku.edu.sa) and Shamala K. Subramaniam (shamala_ks@upm.edu.my)

This research was generously supported by a grant from the Universiti Putra Malaysia Contract Research Grant (Vot No: 6300375) and a scholarship from King Khalid University which covered tuition fees and living expenses during the research process.

ABSTRACT Software Defined Networks enabled-cloud (SDN-Cloud) is experiencing rapid evolution to accommodate the explosive growth of data-driven applications. However, traditional resource allocation algorithms are encountering limitations in efficient resources management. While some existing algorithms strive to minimize power consumption, they introduce network delays, impacting overall performance. Thus, this study aims to address the prevalent challenges of performance efficiency and energy saving within distributed systems. Artificial Intelligence techniques including machine learning and fuzzy logic, are increasingly utilized to develop more adaptive and intelligent resource management models. However, given the dynamic nature of SDN-cloud environments, rapid decision-making during VM allocation is essential to prevent network delays. Furthermore, the limited computational resource of SDN controller requires cautious consideration, as extensive calculations will result in network overhead or increased power consumption. Moreover, achieving subtle balance between network performance and power efficiency still an open challenge. This research introduces Dual-Phase resource allocation Algorithm (D-Ph) for heterogeneous SDN-Cloud networks with the integration of fuzzy logic. D-Ph algorithm indicates the level of utilization of both physical and virtual machines (PM and VM) in datacenters. It aims to find the appropriate host with the necessary capabilities to meet VM resource requirements, specifically processing capacity and memory. The performance of the D-Ph algorithm is evaluated by measuring the response time, serve time of network and central processing unit (CPU), Quality of Service (QoS) violation rate, and power consumption. Results have shown distinctly that D-Ph algorithm maintain high network performance while significantly reduce total power consumption in heavy-loaded large scale network.

INDEX TERMS Artificial intelligence, distributed networks, power management, network performance, quality of service.

I. INTRODUCTION

Over the past decade, technology has rapidly evolved, significantly expanding its application. Traditional networks, with each application maintaining its own data center, led to high operational and maintenance costs. However, the revolution of cloud computing which infusing a data-centric computing paradigm has countered these issues. Cloud providers deliver on-demand and scalable services to their clients, in different locations, through variety of virtualized resources [1]. Recently, Software-Defined Networks (SDN) based Cloud has combined the strengths of SDN and cloud computing, primarily to enhance management. This integration is achieved by implementing SDN ability of observing entire the network via a centralized controller that is responsible on management tasks in cloud datacenter [2]. This innovation contributes in overridden existing challenges in traditional networks such as mobility, scalability, and security [3]. Such challenges are influenced by several factors, for instance resources heterogeneity, inconsistent workload, and resources dependency.

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

Even though, latest network architectures contribute in cost reduction compared to traditional networks, however, application providers have to ensure that offered services meet QoS and Service Level Agreement (SLA) in order to meet tenants' satisfaction to fulfil incoming requests with highly quality standards. Nevertheless, as the number of users and heterogeneity of devices increased, cost associated with power consumption also increased in order to satisfy QoS constraints.

Furthermore, VM allocation represents a key challenge encountered by resource allocation strategies due to its direct impact on network performance and power consumption, both of which have a tradeoff relationship. Each objective is critical in every distributed system, for instance the importance of network performance lies in its association with SLA and QoS which leads to end user satisfaction. While power consumption is a nation concern, directly influenced by resource utilization and management strategies. As a result, several studies aim at achieving balance between these two objectives [4], [5]. Consequently, extensive research has been dedicated in finding reasonable solutions for this issue by exploring different techniques and strategies.

The challenge lies in the fulfillment of QoS for application requests while reducing operational cost, as data center servers are shared by various applications. Consequently, the utilization level of servers is a key deterministic of power consumption. As low utilized servers indicate that each performs inadequately workload computation which lead to an increase of power consumption [6]. While overprovisioning of resources can lead to overloaded hosts and thus degradation of overall performance if not managed properly. Therefore, cloud providers are required to manage resource efficiently in order to ensure high level of network performance with minimum power demands. Therefore, an efficient resource management strategy is vital part of any system. Different studies focus on different aspects on resource management to accomplish specific objectives. Many studies have been conducted on developing efficient resource management techniques.

SDN-enabled Cloud is a revolutionary field, hence, the number of studies addressing it is in a growing stage which provides extensive opportunities to contribute. Furthermore, most of the current studies are focusing either on improving network performance or reducing power consumption. Additionally, the relationship between different resources such as CPU and RAM has not been considered in many studies. These constraints have served as a motivation for the following research questions to be addressed in this paper. The research questions are as follows:

1- How to utilize Artificial Intelligence techniques, particularly fuzzy logic, to address the challenges of efficiently utilizing the most resource-intensive components—CPU and RAM—without compromising network performance?

2- How can fuzzy logic be employed to establish a dynamic connection between host capacities and VM requirements within an uncertain and ambiguous real-time environment, facilitating rapid VM allocation decision-making by the central controller?

3- How to address the aforementioned concerns while minimizing power consumption and avoiding substantial network overhead, given the dynamic nature of SDN-cloud environments and the limitations of SDN controllers' computational resources?

Therefore, we particularly consider resource allocation problem specifically VM allocation. This allocation take place at host level and can be defined as the process of allocating VMs into PMs based on predefined conditions [6], [7]. Furthermore, we employ the concept of fuzzy logic which is type of artificial intelligent which operate through series of decision-making processes to produce an output. Fuzzy logic is utilized in various fields such as green computing, machine learning, artificial intelligent and cloud computing. It is also used to verity of issues in cloud computing such as load balancing, resource scheduling, job scheduling, and QoS optimization [8], [9]. It imitates human reasoning in ambiguous real-world situation where the process involves three main steps: fuzzification, inferencing, and defuzzification to form a fuzzy system. Incomplete information, such as numerical data and linguistics values, is processed in fuzzy systems in order to produce practical output to support decision making and control processing [10].

The nature of VM allocation, where real-time decisions must be made to efficiently manage resources and meet varying workload demands, the choice of an appropriate methodology is of paramount importance. While deep reinforcement learning methods have shown remarkable capabilities in various domains, their application in VM allocation presents several challenges. These challenges include the need for extensive computational resources, time-consuming training periods, and the requirement of vast amounts of training data. Furthermore, the complexity of deep reinforcement learning algorithms may lead to reduced interpretability, hindering the ability to validate and fine-tune decisions based on domain expertise. In contrast, fuzzy logic emerges as a compelling alternative due to its inherent advantages. Fuzzy logic-based algorithms offer simplicity and lower resource requirements, which suits the nature of limited computational resource of SDN, allowing for faster decision-making processes while ensuring real-time responsiveness in dynamic data center environments. Additionally, the interpretability of fuzzy logic enables decision-makers to comprehend the underlying reasoning behind allocation choices, instilling confidence in the results and facilitating practical adjustments based on real-world insights [11], [12].

The adoption of fuzzy logic emerges as an efficient solution to effectively tackle the VM allocation challenge. Considering the implications of VM allocation on overall data center efficiency, cost-effectiveness, and user experience, combines with the remarkable attributes of fuzzy logic including simplicity, adaptability, interpretability, and

real-time performance. Moreover, the scalability and practicality of fuzzy logic-based algorithms further solidify their position as a promising choice for dynamic and large. scale data center environments, where accuracy, efficiency, and responsiveness are of paramount significant.

In this research, we contribute to the area of Software-Defined Cloud (SDN-Cloud) by developing a VM allocation that is inspired by the fuzzy logic system for heterogeneous datacenter. The allocation decision is conducted through series of steps that include building ratio, sorting, and comparing to find the best candidate host for a specific VM.

We propose a novel technique that utilizes fuzzy logic in order to determine the suitability of a host to accommodate a specific VM, based on ratio calculation between different resources, namely processing capacity and memory. These ratios serve to indicate the host's utilization level prior to the allocation process. Hence, in this paper, we take into account several factors, including the most power-consuming resource (processing capacity and memory), the level of utilization for both PM and VM, the suitability of the chosen host for a specific VM, and an exploration of how our proposed strategy impacts network performance and power consumption.

Our VM allocation approach pivot on two main phases: fuzzification and de-fuzzification. The Fuzzification phase involves calculating distinct resource ratios as indicators of overall utilization, which are also used to establish associations between hosts and VMs. Subsequently, hosts are sorted based on their resource ratios, from the most-utilized to the least-utilized host. The de-fuzzification phase then commences with the extraction of actual available resource values from the ranked hosts, starting from the most-utilized host. This leads to re-calculation of the resource ratio. Similarly, the VM requested resource ratio calculation is carried out with the aim of optimizing the host's capability to accommodate the requested VM. Crucially, the adaptability of fuzzy logic to handle uncertainties and vague information aligns seamlessly with the unpredictable nature of VM allocation scenarios, where workloads fluctuate and resource availability varies. By dynamically adjusting resource ratios based on diverse factors, such as workload patterns and traffic fluctuations, the fuzzy logic-based approach ensures efficient VM allocation, maximizing resource utilization without compromising on host capacities.

The key contributions of this research are:

- Fuzzy-Based VM allocation in SDN-Cloud platform that takes into consideration network performance and power saving objectives.
- Two-Level VM allocation that considers most-consuming resource, namely processing capacity and memory, of two types of machines, physical hosts and VMs, contributes in significantly to power saving with zero negative impact on network performance in SDN-Cloud heterogeneous datacenter.

The rest of this paper is organized as follows: Section II provides an overview of the background. Section III presents the related work. The problem statement, proposed system architecture and Dual-Phase VM allocation algorithm are discussed in Sections IV and V, respectively. Simulation experiments are presented in Section VI. Finally, conclusion is presented in Section VII.

II. BACKGROUND

In this section, we will provide a brief introduction to cloud computing, SDN, and SDN-Clouds. We will discuss the similarities and differences between aforementioned architectures.

A. CLOUD DATACENTER

Cloud computing is paradigm that provides on-demand computing resources, where profit is gained through pay-per-use model and resources are provisioned in advance. One of the defining characteristics of cloud computing is the provision of on- demand services such that resources are automatically allocated whenever a client makes a request. Furthermore, the cloud ensures a variety of resources, employing a feature known as resource pooling. Cloud providers offer a vast range of resources for clients, allowing them to select based on their specific goals, such as storage and processing tasks. These resources can be shared by multiple users simultaneously, yet in a way that remains isolated and secure. Also, pay-per-use model is another characteristic of the cloud, the provided services are measured. Hence, users pay only for what they actually consumed. In addition, elasticity is a prominent feature of cloud computing as clients can scale resources demand up or down to their work requirements. Broad network access is another primary characteristic, allowing clients to access these resources anytime, anywhere using compatible devices. Therefore, cloud computing facilitates consumers work and reduce the maintenance cost of traditional IT networks [13], [14].

Figure 1. illustrates the evolved entities in cloud computing, as defined by the National Institute of Standard Technology (NIST) [13]. Various services, including computation, storage, and applications are categorized under three main sections: Software as a Service (SaaS), Platform as a service (PaaS), Infrastructure as a service (IaaS).

SaaS provides users with applications that run on the cloud; however, users have no control over provided services. In contrast, PaaS provides users with a development environment to deploy applications, without control over resources. IaaS supplies users with all essential resources to be provisioned for their applications, such as operating systems, but without direct control over core infrastructure of the cloud.

Furthermore, there are four types of cloud deployment models: private, public, community, and hybrid. A private cloud is used by a specific organization, with resources accessed by exclusively by its clients. Public clouds are available for communal use. Community clouds serve a limited subset of an organization due to specific concerns such as security. Finally, hybrid clouds merge two or more of aforementioned deployment methods [13].

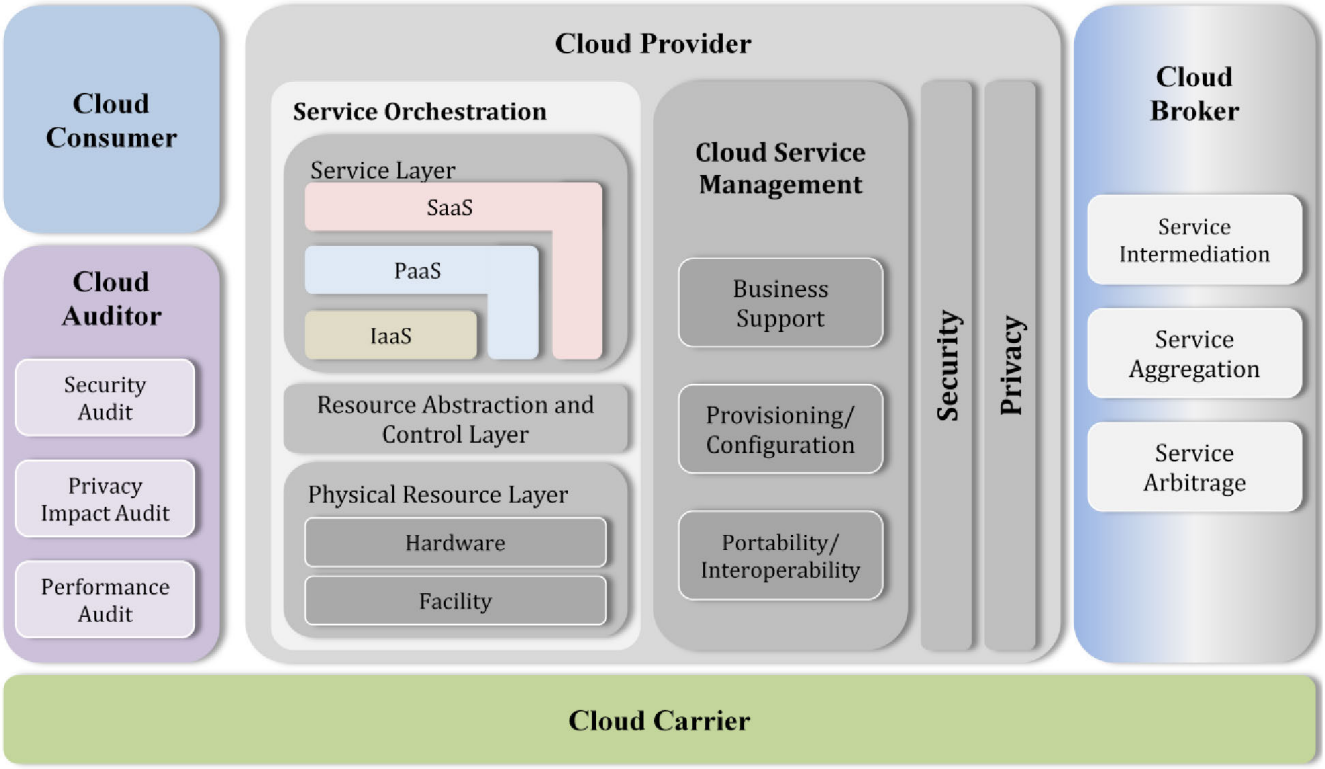

FIGURE 1. The cloud conceptual reference model [13].

Virtualization is a key element in cloud computing and is facilitated by a hypervisor that resides between the hardware and the operating system. This function can manage multiple requests from different tenants by creating multiple VMs on single PM. Consequently, scalability is achieved by increasing or decreasing resources, alongside power efficiency, and better resource utilization. The most renowned open source platform for cloud computing is OpenStack. It enables administrators to manage a datacenter with a wide range of resources and allow users to provision these resources through an application programming interface (API) [15].

B. SOFTWARE-DEFINED NETWORKS (SDN)

Traditional networks rely on network devices to make forwarding and routing decisions through hardware tables embedded in the devices themselves, such as bridge and router. Furthermore, traffic rules, including filtering and prioritizing, are locally implemented on each device. However, SDN has brought advancement in this domain by simplifying the design of network devices. It aims to reduce the complexity of both hardware and software components of network devices.

The basic concept of SDN is to transfer controllability from network devices to a single central device, namely the control unit, which manages and controls operations. The control unit has the ability to observe the entire network and make decisions regarding forwarding and routing. Conversely, the task of forwarding is handled by network hardware devices, which also manage filtering and traffic prioritization [16].

OpenFlow is a well-known SDN design that follows the fundamental architecture of decoupling the control plane from the data plane. In this design, the controller and switches communicate through the OpenFlow protocol. The switches contain flow tables and flow entries that includes matching fields, counters and a set of actions. The controller is capable of performing a series of actions on the flow entries, such as updating, deleting, and adding entries [17]. According to [18], SDN architecture comprises southbound and northbound communications. Southbound communication represents the interaction between the controller and switches, facilitating control through an open interface that uses a standardized protocol, such as the OpenFlow protocol. Conversely, northbound communication represents the communication between services and the controllers. An example of this is high-level network services seeking information about the network policy from network controllers, which then results in these controllers communicating with each other to fulfill the request.

C. SOFTWARE-DEFINED CLOUD DATACENTER (SDN-CLOUD)

Even though cloud computing is a powerful technology, there are still some unresolved issues and challenges, such as network mobility, scalability and security. Consequently, the

possibility of extending SDN into cloud environment is under investigation [19]. The OpenFlow protocol, used in SDNs to standardize communication between controllers and network services from one link to switches, facilitates what is referred to as northbound and southbound communication [17].

This role is notably served in datacenters by Open vSwitch (OVS), a software-based virtual switch. OVS enables OpenFlow to control and manage flows via a built-in external interface. The switch is located inside the hypervisor and includes two modules: the fast path and the slow path. The fast path module, residing in the kernel of the OVS software, is responsible for forwarding and counting table entries. Conversely, the slow path, located in the user space, manages forwarding rules and communicates with external interfaces, such as OpenFlow and the virtualization layer. This function allows for the creation and connection of new virtual switches to virtual and physical interfaces.

In addition to this, OpenDayLights is an open source platform that supports northbound communication in SDN. It does this by creating controller which offers an API to high level applications and services [18], [19].

Therefore, the need for aforementioned frameworks be designed and planned in advance is essential in order to support core cloud functions such as scheduling and resource allocation. Even though SDN can be extended to various infrastructure by modifying elements in controller, this research focuses primarily on computing resources.

III. RELATED WORK

This research considers multiple criteria related to VM allocation in the SDN-Cloud environment, including artificial intelligent-based VM allocation approaches, variations in host and VM computing resource, heterogeneity, and primary objective of either performance or power savings. Since there have been limited studies conducted in the field of SDN-Cloud environment, we explore these aspects in the context of cloud computing as well.

Different VM allocation algorithms have been developed using various strategies, each with different objectives. For example, the Priority Aware VM Allocation (PAVA) algorithm [20] primarily considers priority when allocating VMs, with prioritization predetermined as crisp value i.e., 0 or 1. This algorithm operates by adapting co-localization of hosts based on edge connection and placement of VM based on available resource in term of processing capacity and bandwidth in the SDN-Cloud. This strategy contributes to network performance by minimizing response time.

The process of VM allocation on datacenter networks in EQVMP [21] comprise three main steps: hop reduction, VM sorting based on requested resources, and mapping VMs to the best-fit hosts. Hop reduction is implemented by maintaining similar VM quantities across different groups of hosts. Similarly, VM placement in [22] considers high power consumption hosts, which are selected after a task classification has been conducted to find the appropriate VM in the initial stage. Correspondingly, MAPLE system [23] utilizes effective bandwidth to allocate resources while preserving QoS constraints. It allocates sufficient bandwidth to handle requests, as analyzed using traffic traces collected by servers. As the MAPLE system employs First Fit Decreasing (FFD) algorithm, which was later extended to MAPLEx [24] to incorporate a server localization feature.

Furthermore, the MAPLE-Scheduler [25] is developed for SDN and leverage SDN advantages by dispersing monitoring agents in the network to gather information for management decision purposes.

Other VM allocation techniques adopt artificial intelligent approaches, such as HGAPSO which utilize Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for the VM allocation process in cloud datacenter [26]. Additionally, the genetic algorithm is combined with the tabue search algorithm in the VM placement process to reduce power consumption [27]. In another approach, VM allocation utilizes fuzzy logic implemented in the controller, coupled with reinforcement learning in order to locate as many VMs as possible on as few hosts as possible. This aims to reduce power and improve resources utilization [28]. Although previous VM allocation algorithms were designed for specific objectives, some were proposed to focus on multiple objectives simultaneously, aiming to improve factors such as energy consumption, network performance, SLA violation, and more. For instance, VM placement has been proposed to maintain SLA by enabling early detection of overloaded PM resources [29]. The predictive anti-correlated VM placement algorithm PACPA [30] succeeds in reducing power consumption and maintaining SLA by considering CPU utilization prior to VM allocation and adopting neural networks to predict future VM allocation dynamically. Although different techniques have been utilized, few contributions were dedicated specifically to the area of SDN-Cloud.

Moreover, most studies managed to contribute to one objective at the cost of another. For example, the proposed VM allocation detailed in references [20], [21], [23], [24], [25], and [26] achieves notable results in enhancing network performance. This is in contrast to the VM allocation techniques that were developed as per references [22], [27], [28] and [30]. The contribution on these studies primarily focus in reducing power consumption. Moreover, few studies have been devoted to involvement in an SDN-Cloud environment. Therefore, in this research, we aim to establish a balanced relationship between power wastage and network performance in SDN-Cloud, while considering the utilization level of multiple resources (computing power and memory) of both PM and VM. Table 1 summarizes the aforementioned studies in terms of the implemented feature, targeted environment, core resources used in the design of the proposed algorithms, heterogeneity of VMs requests, and primary contribution to overall system.

IV. PROBLEM STATEMENT

The datacenter contains n physical hosts $H = \{h_1, h_2, h_3, \ldots, h_n\}$ where each host has a limited capacity of resources

TABLE 1. Related work comparison table.

Ref.	VM Allocation Feature	Environment	Resources	VM Type	Contribution
J. Son & R. Buyya[20]	priority & co-location	SDN-Cloud	Computing Power Only	Heterogeneous	Network Performance
S.-H. Wang , et al[21]	Hop Reduction	SDDCN	Computing Power, Memory, & Bandwidth	Homogeneous	Network Performance & Energy Efficiency
Z. Ding, et al[22]	Tasks Classification	DC	Computing Power Only	Homogeneous	Energy Efficiency
R. Wang et al [23]	Effective bandwidth	Cloud	Bandwidth	Homogeneous	Network Performance
R. Wang et al [24]	Deployment of Anti-colocation constraint	Cloud	Bandwidth	Homogeneous	Network Performance
R. Wang et al [25]	Link Utilization	SDN	Bandwidth	Homogeneous	Network Performance
N. K. Sharma and G. R. M. Reddy [26]	Genetic Algorithm & Particle Swarm Optimization	Cloud	Computing Power & Memory	Heterogeneous	Energy Efficiency
D. Zhao , et al[27]	Genetic Algorithm & Tabue Search Algorithm	Cloud	Computing Power & Memory	Heterogeneous	Energy Efficiency
A. Jumnal, et al [28]	Fuzzy & Reinforcement Learning	Cloud	Computing Power & Memory	Homogeneous	Energy Efficiency
S. Alanazi and B. Hamdaoui [29]	K-Nearest Neighbor Regression	Cloud	Computing Power	Heterogeneous	Network Performance
R. Shaw, et al [30]	Neural networks	Cloud	Computing Power	Heterogeneous	Energy Efficiency
Dual-Phase Resource Allocation	Fuzzy-Based & Resource Ratio	SDN-Cloud	Computing Power & Memory	Heterogeneous	Network Performance & Energy Efficiency

denoted by h_i=$\{CPU_i, RAM_i\}$. On the other hand, there are m VMs to be allocated to a physical host, defined as VM=$\{vm_1, vm_2, vm_3, \ldots, vm_j\}$. Each vm_j has its resource requirements specified by vm_j= $\{CPU_j, RAM_j\}$. Our goal is to allocate the maximum possible number of *VMs* into each individual host H, with the purpose of minimizing the total number of active hosts. This will result in reduced power consumption while maximizing resources utilization. However, this objective should be accomplished without compromising QoS, which in this study is equated to response time. Therefore, the primary objectives can be expressed as follow:

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{n} Energy(h_i) \\ &\text{and maximize } \sum_{i=1}^{n} CPUUtilization(h_i) \\ &\text{and maximize } \sum_{i=1}^{n} MemoryUtilization(h_i) \qquad (1) \\ &\text{subject to :} \\ &\sum_{j=1}^{|VM|} ResourceRequirment(vm_j) < ResourceCapacity(h_i) \qquad (1a) \\ &\sum_{j=1}^{m} ResourceRatio(vm_j) \leq UtilizationRatio(h_i) \qquad (1b) \end{aligned}$$

Energy efficiency is a critical concern for cloud data centers, as they consume substantial amounts of power. Minimizing energy consumption directly translates to cost savings and reduced environmental impact. By optimizing the VM allocation to consolidate VMs on fewer hosts, the algorithm aims to power down idle hosts and minimize energy consumption Therefore, the primary objective is to reduce power consumption by reducing the total number of active hosts. Power reduction is attained through a decrease of active hosts. This decrease in hosts is achieved by maximizing host utilization, where allocation is performed based on each host's utilization level. Furthermore, host utilization considers different resource requirements, specifically processing capacity (CPU) and storage (memory).

Maximizing CPU utilization ensures that computing resources are efficiently utilized, minimizing the number of idle CPU cores. Higher CPU utilization results in improved performance and responsiveness, as computational tasks are distributed effectively across the data center. Similarly, maximizing memory utilization prevents the wastage of valuable RAM resources. Efficient memory allocation ensures that VMs have adequate memory to run applications optimally, avoiding performance bottlenecks and improving overall system performance. Hence, to meet the main objectives, our proposed algorithm ensures that the resource demand by the total number of allocated VMs in the same host does not exceed the host's capacity. Moreover, the resource ratio of required resources by each VM will not surpass the available resources ratio of a hosted machine. This is to ensure that QoS is not compromised by preventing overutilization of hosts. Therefore, resource ratio value serves as an indication of overall resource utilization level, incorporating both processing capacity and memory. By enforcing this constraint, the algorithm ensures that VMs are allocated only to hosts with sufficient available resources to accommodate them. This helps to maintain stability, prevent resource contention, and enhance the overall performance of the datacenter. Consequently, D-Ph algorithm aims to achieve resource efficiency and performance optimization by efficiently allocating VMs based on resource ratios while respecting the optimization constraint to avoid overloading hosts. The objective equations reflect the key performance metrics of energy consumption, CPU utilization, and memory utilization, ensuring an effective and balanced allocation strategy in the data center. While we focus on two distinct objectives, we measure the power consumption of hosts and switches to verify that the proposed algorithm is capable of reducing the total power

consumed. On the other hand, we measure response time and QoS violation rate to study the impact on network performance.

V. PROPOSED SYSTEM MODEL

A. SYSTEM ARCHITECTURE

Figure 2 displays the overall architecture of our proposed system. Inspired by fuzzy systems, our system takes application requests as ambiguous data, without prior prioritization or classification. However, different requests often contain a random number of VMs with varying specifications. Such specifications may include the type of VM, processing core and capacity, storage size, and memory. Additionally, flow specification is determined by factors such as source, destination, and bandwidth. These specification is typically derived from commercialized cloud providers. The challenge, therefore, lies in providing adequate service for different types of application requests, while reducing overall energy consumption. As datacenter power consumption is increasing exponentially in order to meet the demands of cloud resources, thus, we focus on hosts due to their direct impact on power usage. This power usage is based on their power mode which can be either 'on', 'off', or 'idle'. We presume that 'mode-on' hosts consume more energy than that are 'off' or 'idle', primarily due to computational and storage requests. Considering hosts utilization and performance levels, our system aims to minimize the number of 'mode-on' hosts while avoiding over-utilization. This is crucial because overloaded hosts can result in degraded network performance and increase QoS violation, both of which are indicators of poor system management. Hence, we also consider VM resource utilization in order to maintain network performance.

To simulate fuzzy logic process, the system obtains the available resources of hosts and VM requirements in the form of imprecise numerical data. Subsequently, the fuzzification phase is initiated to assign weights to the hosts based on their available CPU and RAM. These weights aim to differentiate between high-utilized and low-utilized hosts, and they will be used to sort hosts accordingly. Thus, the assigned weights reflect the degree of the utilization of the hosts. The weights are calculated as the ratio of processing capacity including processing elements (Pes) and (MIPS) to memory, which is determined by RAM capacity. The value of this ratio indicates the level of utilization of different resources. Then phase two, the de-fuzzification process begins by extraction the actual resource capacity of the most utilized host based on their assigned weights, and re-calculating its resource ratio. This is executed with the process of inference rule, which performs a comparison with the VM's requested resources ratio to assess the suitability of the candidate host to accommodate the requested VM.

The allocation will be conducted if the resource ratio of selected host is higher than the ratio of demanding resource by requested VM, represented in equation (1b) as optimization constraint. If this condition is not met, the system will check the capabilities of the subsequent host and repeat the process until a suitable host is found.

TABLE 2. List of symbols.

Symbol	Description
H	Set of hosts in the datacenter
$h(i)$	i^{th} host $\in H$
$P_{(Hosti)}$	Power consumption of host I (kW)
P_{idle}	Idle power consumption of host (kW)
P_{peak}	Peak power consumption of host (kW)
P_{port}	Power consumption of each port on switch (kW)
$P_{(Switchi)}$	Power consumption of switch i (kW)
P_{static}	Power consumption of switch without traffic (kW)
q_i	The number of active ports on switch i
r_v	QoS violation rate
t_{era}	Response time of a workload measured from ER algorithm (second)
t_x	Response time of a workload measured from the deigned algorithm (second)
u_i	CPU utilization percentage of host i
VM	Set of virtual machines
$Vm(j)$	The jth virtual machine on host i
$\|VM\|$	The total number of VMs in the datacenter
W	Set of workloads
w	Workloads with requested *vms*
w_v	Workload v $\in$ W
$\|W\|$	Total number of workloads

B. DUAL-PHASE RESOURCE ALGORITHM

Our proposed Dual-Phase (D-Ph) resource allocation algorithm functioning can be break down as following: The first step of the algorithm involves localizing hosts based on edge connections. It aims to minimize network traffic and reduce latency in data transmission. This localization strategy can lead to improved response times and enhanced user experience. Then, the algorithm uses fuzzy logic to develop resource ratios for the most power-consuming resources, memory, and CPU. Fuzzy logic allows for flexible representation of resource allocation decisions, enabling the algorithm to consider a range of factors and uncertainties that might affect VM placement. The fuzzy logic-based resource ratios guide the allocation of multiple virtual machines (VMs) into hosts efficiently. By optimizing the allocation based on memory and CPU ratios, the algorithm aims to pack as many VMs as possible into each host, reducing the overall number of active hosts needed in the data center. However, while maximizing VM density per host, the algorithm also ensures that host overutilization is avoided. The fuzzy logic-based resource ratios set constraints that prevent hosts from becoming overloaded, ensuring that each host operates within its resource capacity.

In order to find the best candidate host for the requested VM, our algorithm considers factors including network

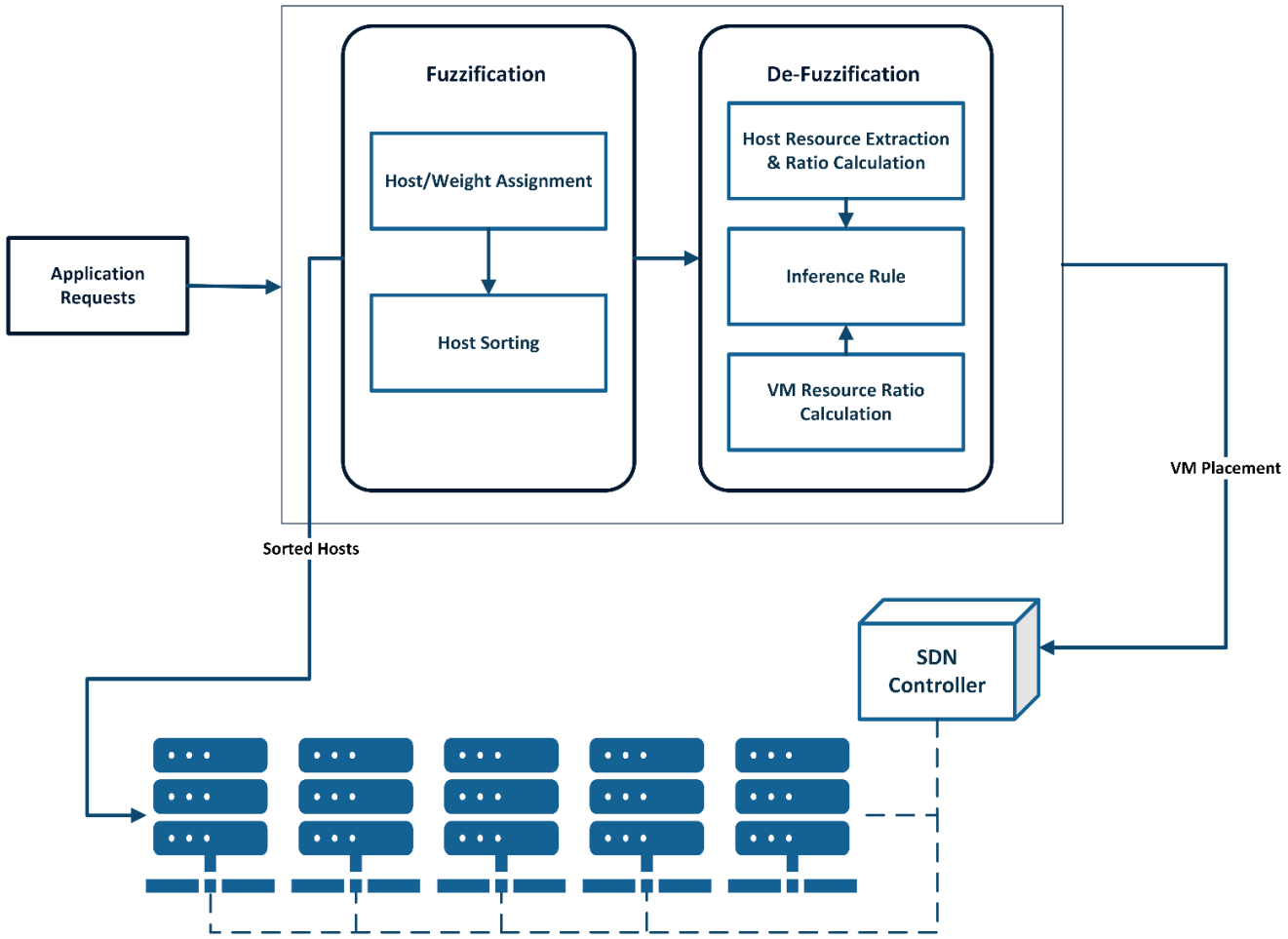

FIGURE 2. System architecture of fuzzy-inspired resource VM allocation.

topology, physical specification, VM resource demands, available host resources, hosts. locality, network performance, and power consumption. Consequently, adjacent hosts are grouped according to their edge connection in an attempt to minimize network traffic between them. which leads to enhanced performance.

Then, the algorithm extracts each host's available resources, which are used as input to calculate the ratios based on processing capacity and memory. These ratios are used as weights and assigned to their respective hosts. After that, the hosts are sorted based on their assigned weights, from highest to lowest. The host with highest weight indicating the most-utilization, is the first nominated host to accommodate the requested VM.

However, before allocation process occurs, we need to ensure the suitability of the nominated host. Hence, our algorithm calculates the resource ratio based on the VM's resource demands to guarantee that sufficient resources are available for each VM to operate efficiently on the selected host. Similarly, to ensure that the VM's resource request doesn't exceed the current utilization level of the host. This management in turn, ensures the prevention of the performance degradation by the host. The resource ratio of the host and VM is extracted as follow:

$$\text{ProcessingCapacity} = \text{PEs} \cdot \text{MIPS} \tag{2}$$

$$\text{ResourceRatio} = \text{ProcessingCapacity/RAM} \tag{3}$$

Thus, resource ratio is utilized to approximate a host's level of utilization, as our algorithm primarily targets the most-utilized hosts in order to maintain the number of high utilized-hosts at lowest possible level. This approach ensures that the remaining hosts are set to idle, which will eventually lead to power saving. As a result, hosts are sorted by their assigned weights, from the most to the least utilized.

However, D-Ph algorithm also considers network performance. Therefore, resource requirements requested by each VM are assigned with a ratio similar to those of the hosts. This enables the identification of whether a particular host is capable of handling a specific VM by comparing the two values.

The resource ratio of first host in the sorted queue, which represents most-utilized host, is compared to the resource

Algorithm 1 Dual-Phase VM Allocation (D-Ph)

Input: *vm*: virtual topology consisting of VMs to be placed.
Input: *w*: Workloads with requested VMs
Input: *H*: host list
Input: *vmDemands*: resource demands by VM
Output: VM allocation
1: $Host_{\text{adjacent}} \leftarrow$ groups of adjacent hosts based on edge connection
2: **for each** $h \in Host_{\text{adjacent}}$ **do**
3: *hostAVBLR* $\leftarrow h$ (PES, MIPS, RAM)
4: h*RRatio* $\leftarrow$ *ResourceRatio*(*hostAVBLR*)
5: **end for**
6: Sort $Host_{\text{adjacent}}$ based on *hRRatio* in ascending order
7: *queueHostCapacity* $\leftarrow$ sorted $Host_{\text{adjacent}}$
8: **for each** $h \in$ *queueHostCapacity* **do**
9: *hostResources* $\leftarrow$ *hostAVBLR* (*h*)
10: *HostRRatio* $\leftarrow$ *ResourceRatio(hostResource)*
11: *vmRRatio* $\leftarrow$ *ResourceRatio (vmDemands))*
12: **if** *vmRRatio* $\leq$ *HostRRatio* **then**
13: *placed* $\leftarrow$ true
14: update *hostResources* of *h*
15: **else**
16: *examine the following* $h \in$ queueHostCapacity
17: **end if**
18: **end for**

demands ratio of the requested VM. Through this comparison, we evaluate the suitability of the candidate host to the particular VM, leading to one of the two outcomes: VM allocation, followed by an update of available resourcse and resorting, or the examining of the next host in the queue.

The detailed workings of the D-Ph algorithm are explained in algorithm 1.

The effectiveness of this algorithm can be attributed to its combination of localization, fuzzy logic-based resource ratio development, and intelligent VM allocation strategies. For instance, by localizing hosts based on edge connections, the algorithm reduces data transmission distances, leading to lower network traffic and reduced latency. This results in reduced communication overhead and enhances user responsiveness.

Additionally, the integration of adaptive resource ratios using fuzzy logic enables the algorithm to dynamically adjust resource allocation based on various factors, such as workload patterns, traffic fluctuations, and resource availability. This adaptability allows the algorithm to efficiently handle varying conditions and achieve efficient VM allocation.

Moreover, the fuzzy logic-based resource ratios enhance resource utilization, ensuring that VMs are allocated in a manner that maximizes the utilization of power-consuming resources (memory and CPU) without exceeding host capacities. As a result, wasted resources are minimized and overall data center efficiency is increased. The algorithm also achieves high VM density by efficiently allocating multiple VMs into a single host, which reduces the number of active hosts required. This not only reduces operational costs but also improves resource consolidation, leading to better power utilization and reduced physical space requirements.

Furthermore, fuzzy logic allows the algorithm to adapt to dynamic changes in workload and resource availability. As the load on hosts fluctuates, the resource ratio membership can be dynamically updated, ensuring that VMs are allocated to hosts based on their current suitability. This dynamic allocation helps prevent host overutilization or underutilization. As well, we designed our VM allocation algorithm with practicality and scalability in mind aiming to improve network performance. Therefore, the real-time evaluation and dynamic nature of fuzzy logic-based VM allocation minimize the overhead associated with frequent recalculations and adjustments. The algorithm can efficiently assess the load status and make informed decisions without placing undue computational burden on the system. By focusing on simplicity and real-time responsiveness, our system becomes applicable to dynamic and large-scale data center environments, where the ability to adapt quickly to workload changes is crucial.

These benefits significantly contribute to the algorithm's scalability, as its adaptability and localization strategy make it well-suited for large-scale data centers while maintaining effective resource allocation. Additionally, the algorithm's real-time performance, typical of fuzzy logic-based approach, ensures faster decision-making compared to complex optimization methods. This real-time capability is crucial for handling dynamic workloads and ensuring responsiveness in distributed environments.

VI. SIMULATION EXPERIMENT

A. SIMULATION SETTINGS

The proposed algorithm is implemented using CloudSimSDN simulation environment designed to evaluate resource allocation policies for SDN-Cloud environment. It is extended from CloudSim simulation toolkit. CloudSim is focused on simulating general cloud computing environments, while CloudSimSDN serves as an extension or integration of CloudSim with SDN concepts.

The architecture of CloudSimSDN leverages the capabilities of CloudSim for cloud computing simulation and incorporates additional components and modules related to SDN. As a result, researchers can investigate the dynamic interplay between SDN and cloud computing within virtualized data centers. CloudSimSDN offers specialized modules for simulating SDN-based network management, programmable forwarding rules, and network resource provisioning, complementing the traditional cloud computing simulation features of CloudSim.

Figure 3 illustrates the architecture of CloudSimSDN, wherein users provide topology configurations comprising user code and scenarios, along with physical and virtual topology configurations. Additionally, workload descriptions, including submission times, job processing sizes, and

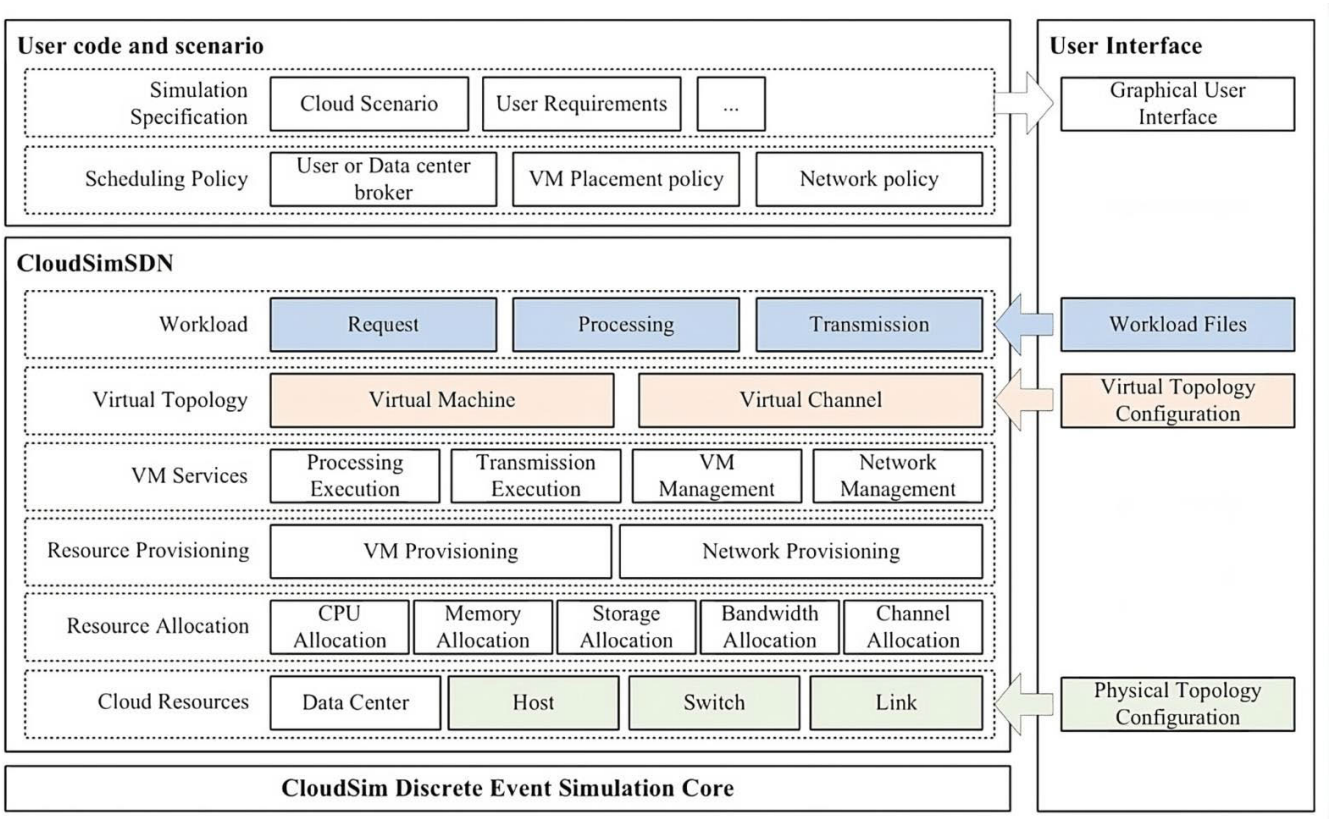

FIGURE 3. CloudSimSDN architecture [31].

TABLE 3. Simulation parameters.

Topology	8-Pod Fat Tree
Core Switches	16
Aggregation Switches	32
Edge Switches	32
Number of Hosts	128
Dataset	Wikipedia Workload
Workload Model	Three-Tier Application Model [30]
Total Workload	$n \approx 46{,}369$

traffic data sizes, are supplied. Users also specify scheduling policies, such as VM placement algorithms and network policies. Brokers can simulate end-users or data centers, and users have the option to use built-in policies or create their own. The VM Services layer manages VMs and the network, while the Resource Provisioning layer includes VM Provisioning and Network Provisioning modules. The Resource Allocation layer handles resource allocation as specified in the Cloud Resources layer. This architecture facilitates comprehensive simulation and study of cloud and SDN interactions [31].

In this experiment, an 8-pod fat tree topology is used for the evaluation, which comprises of total 16 core switches, 32 aggregation switches, 32 edge switches and 128 hosts.

In order to assess the effectiveness of the proposed algorithm, we employ real-world dataset: Wikipedia workload. This workload is based on a three-tier application model [30] with the intent to provide a practical solution reflecting real-world scenarios with total workload approximately $n \approx 46{,}369$.

D-Ph algorithm is compared to Priority-Aware VM Allocation (PAVA) algorithm [20]. PAVA algorithm is prioritization-based VM allocation developed specifically for SDN-Cloud datacenter. It processes requests based on their assigned priority status which can be either 0 or 1. Critical applications are given a higher priority during the allocation process, with the priority status of these requests set to 1. This indicates that those applications ought to be processed first compared to regular applications, which are assigned a priority value of 0. However, PAVA is only triggered for high priority request. This means that when a request enters the system with priority status value of 1, it is treated as critical application and is consequently processed by PAVA algorithm. Meanwhile, other VM requests, namely those of a lower priority, are managed through First Fit Decreasing algorithm (FFD).

PAVA operates by allocating VMs to hosts based on the concept of hosts co-localization and facilitates placement by considering groups of hosts with the highest computing resources. Therefore, the allocation decision is made

Algorithm 2 Priority-Aware VM Allocation (PAVA)

Input: *vm*: VM to be placed.
Input: *rd*: Resource demand of vm;
Input: *app*: Application information of vm.
Input: H: List of all hosts in data center.
Output: VM placement map.
$H_{group} \leftarrow$ Group H based on edge connection;
$Q_H \leftarrow$ Empty non-duplicated queue for candidate hosts;
placed $\leftarrow$ false;
if *app* is a higher-priority application **then**
$H_{app} \leftarrow$ list of hosts allocated for other VMs in *app*;
if H_{app} is not empty **then**
  Q_H.enqueue(H_{app});
  for each h_a in H_{app} do
  $H_{edge} \leftarrow$ A host group in H_{group} where h_a is included;
  Q_H.enqueue(H_{edge});
end for
for each h_a in H_{app} **do**
    $H_{pod} \leftarrow$ Hosts in the same pod with h_a;
    Q_H.enqueue(H_{pod});
end for
end if
sort H_{group} with available capacity, high to low;
Q_H.enqueue(H_{group});
while Q_H is not empty **and** *placed* = false **do**
  $h_q = Q_H$.dequeue()
  $C_h \leftarrow$ free resource in host h_q;
  if $rd < C_{hq}$ then
    Place *vm* in h_q;
    $C_h \leftarrow C_h$ - *rd;*
    placed $\leftarrow$ true;
    end if
  end while
end if
if *placed* = false **then**
Use FFD algorithm to place vm;
end if

based on the most substantial amount of available resources among groups of hosts. However, there is no association is established between available host resources and VM resources. While PAVA significantly improves network performance, however, power consumption was maintained. The pseudocode of PAVA algorithm is detailed in algorithm 2. As PAVA is designed exclusively for critical applications, we presumed during this experiment all requests have high priority in order to trigger PAVA and ensure fairness comparison.

The experiment intends to meet objectives related to power consumption and network performance. Therefore, we measure response time and power consumption of hosts and switches to compare with PAVA algorithm.

Furthermore, given D-Ph algorithm aims to ensure high network performance while minimizing power consumption, we consider the average response time, network and CPU serve time, and QoS violation rate as an indicators of overall network performance.

B. EXPIRMENTAL RESULTS

We compare D-Ph and PAVA using the identical number of workloads and simulation settings. However, since PAVA is designed expressly for critical applications, we assume that all requests have high priority. We calculate the energy consumption, which denotes the energy consumed by a resource to complete a workload's execution. High energy consumption typically signifies that a substatinal workload is being processed. Accordingly, we measure the specific power consumption of hosts and switches to evaluate whether D-Ph consumes more power than PAVA, based on the power models of hosts [33] and switches [34]. The power model of the hosts is determined by the percentage of processing capacity utilization:

$$\mathrm{P(Host_i)} = \begin{cases} P_{\mathrm{idle}} + \left(P_{\mathrm{peak}} - P_{\mathrm{idle}}\right) \cdot u_i & \text{if} |\mathrm{VM}| > 0 \\ 0 & \text{if} |\mathrm{VM}| = 0 \end{cases} \tag{4}$$

where u_i represents the percentage of processing capacity and the power consumption of idle hosts has a constant factor. This consumption will be reduced once the host is powered off. The power consumption of switch i is calculated based on the active ports as follow:

$$\mathrm{P(Switch}_i) = \begin{cases} \mathrm{P}_{static} + \mathrm{P}_{port} . q_i & \text{if switch}_i \text{ is on} \\ 0 & \text{if switch}_i \text{ is off} \end{cases} \tag{5}$$

where q_i represents active ports in switch i.

The following subsections represents the results regarding D-Ph algorithm impact on power consumption and network performance together with an analysis of the algorithm complexity.

1) ANALYSIS OF POWER CONSUMPTION

Figure 4 provides a detailed analysis of power consumption in the datacenter, including the power consumption of hosts and switches. Comparing the D-Ph algorithm with the PAVA algorithm, we observe slightly lower reduction in power consumption by switches with the D-Ph algorithm. This reduction can be attributed to the D-Ph algorithm's effective implementation of co-localization, which reduces traffic between connected hosts. By considering VM placement based on host consolidation, the D-Ph algorithm minimizes data traffic passing through switches, resulting in fewer active ports and decreased power consumption. Although the D-Ph algorithm does not operate at the link level, it still successfully reduces power consumption of switches.

In terms of power consumption by hosts, the D-Ph algorithm achieves a significant reduction compared to the PAVA algorithm. The D-Ph algorithm focuses on the utilization of the most power-consuming resources, such as processing capacity and memory. It also takes into consideration other factors for VM allocation, including VM

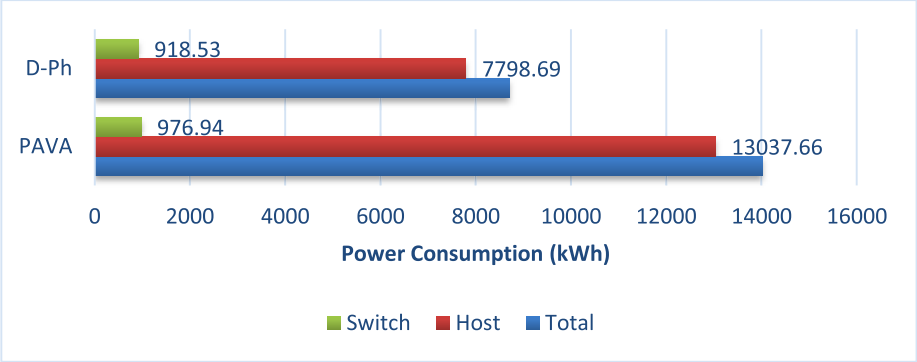

FIGURE 4. Detailed power consumption of hosts and switches.

resource demands and the relationship between available and requested resources. By optimizing resource utilization and minimizing the number of hosts used, the D-Ph algorithm effectively reduces power consumption at the host level.

Overall, the D-Ph algorithm results in a noticeable decrease in total power consumption compared to the baseline algorithm, PAVA. This is because the D-Ph algorithm is designed to optimize the operational cost of the datacenter by maximizing resource utilization and minimizing the number of hosts, based on resource ratios. Consequently, the D-Ph algorithm achieves an approximate 40% reduction in total power consumption compared to PAVA, as evidenced by obtained results.

The behavior of the D-Ph algorithm in reducing power consumption is attributed to its co-localization concept, VM placement based on host consolidation, utilization of power consuming resources, and considerations for VM allocation. The D-Ph algorithm successfully reduces power consumption at both hosts and switches levels, leading to a significant drop in total power consumption compared to existing algorithms like PAVA. This highlights the effectiveness and efficiency of the D-Ph algorithm in optimizing datacenter power consumption and operational cost.

Analysis of Network Performance like PAVA. This highlights the effectiveness and efficiency of the D-Ph algorithm in optimizing datacenter power consumption and operational cost.

2) ANALYSIS OF NETWORK PERFORMANCE

Figure 5 displays the average response time achieved by both the PAVA and D-Ph algorithms. Although there is not a significant difference in average response time between the two algorithms, the average response time achieved by D-Ph is slightly lower than that of PAVA. This difference can be attributed to the hosts grouping step is performed by both algorithms, which also contributes to the reduction of power consumption in switches.

Furthermore, the hosts grouping stage plays a role in decreasing transmission across networks between hosts, thereby improving network performance. We also analyze the

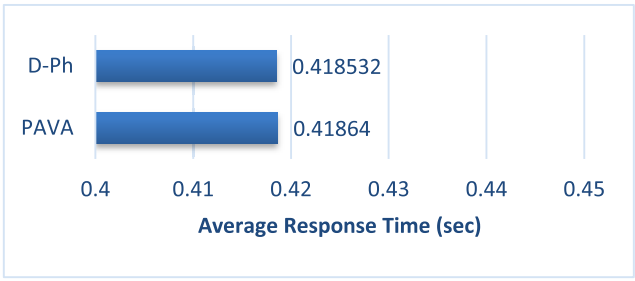

FIGURE 5. Average response time.

average response results in Figure 6, focusing on variations based on the serving time of the network and CPU.

Upon analysis, we observe that the average serve time of the network and CPU in both D-Ph and PAVA is similar, supporting the findings illustrated in Figure 6 where D-Ph exhibited a similar effect on overall network performance as PAVA. However, it is important to note that when the network is overloaded with complex applications, PAVA typically outperforms in improving network performance. Despite this, our conclusion is that D-Ph excels in maintaining high performance in overload-traffic scenarios within large-scale datacenters.

This conclusion is supported by existing research paper [20] that have explored the behavior of different algorithms in datacenters. These study have demonstrated that PAVA succeeds in minimizing response time compared to well-known algorithms such as FFD, Random algorithm, and Dynamic Flow algorithm. Accordingly, D-Ph can effectively maintain high performance even under heavy network traffic. This further validates the capability of the D-Ph algorithm in enhancing network performance and supporting the efficient operation of large-scale datacenters.

Further, to ensure that the overall performance is not negatively affected by D-Ph, we take into account the QoS violation rate of both algorithms. This violation rate is computed on the basis of the response time obtained by Exclusive Resource Algorithm (ER), which provides dedicated hosts for each VM. Given that ER provides devoted hosts for every

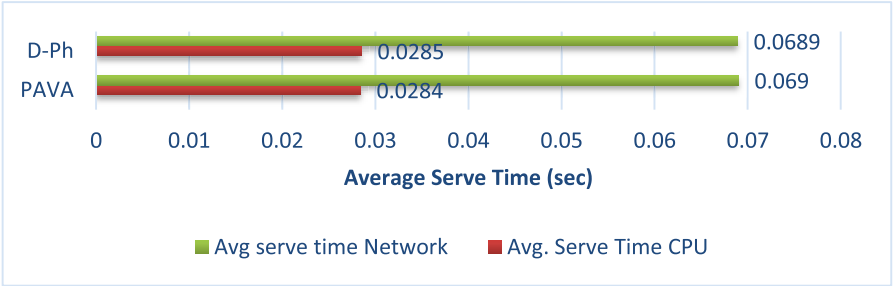

FIGURE 6. Average serve time of Network and CPU.

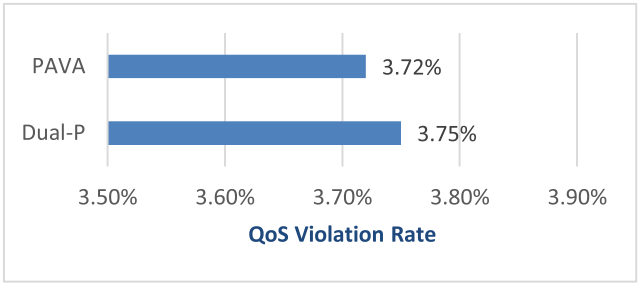

FIGURE 7. QoS violation rate percentage.

VM, all resources offered by the host are fully utilized by a single VM application, resulting in the lowest possible response time.

The QoS violation rate is determined using the response time obtained from ER. We calculate it by comparing total number of workloads whose response time exceeds the value set by ER algorithm, divided by the total number of workloads. The following equation represents the formula for calculating the QoS violation rate:

$$r_v = \frac{|\{w_v \in W \mid t_x(w_v) > t_{era}(w_v)\}|}{|W|} \tag{6}$$

- t_x : response time of a workload (w_v) measured from D-Ph & PAVA algorithm.
- t_{era} : response time of a workload (w_v) measured from ER algorithm.
- W : workload sets.

The results of the QoS violation rate, as shown in Figure 7, support our conclusion that D-Ph does not negatively impact the network since it achieves similar results to PAVA that succeeds in reducing QoS rate in overloaded-traffic network within a large scale datacentre.

C. ANALYSIS OF ALGORITHM COMPLEXITY

The process of creating the collection of adjacent hosts based on edge connections may have a time complexity of O(n^2) in the worst case if all hosts are adjacent to each other. This is because, in the worst case, each host needs to be compared with every other host to determine adjacency.

The calculation of available resources for each host and sorting the adjacent hosts based on resource ratio has a time complexity of O(n log n). Hence, sorting the hosts takes O(n log n) time, where n is the number of hosts. The iteration over the sorted adjacent hosts has a time complexity of O(n) since it involves iterating over each host once.

Overall, the time complexity of the algorithm can be approximated as O(n^2) for the worst-case scenario where all hosts are adjacent, or O(n log n) for the average case when considering the sorting operation.

The space complexity of the algorithm is relatively low, primarily depending on the storage of the input data and a few auxiliary variables. It does not involve any significant data structures that grow with the input size. Therefore, the space complexity can be considered as O(1), indicating constant space requirements.

TABLE 4. List of abbreviation.

Abbreviations	Definitions
API	application programming interface
CPU	Central Processing Unit
D-Ph	Dual-Phase VM Allocation Algorithm
ER	Exclusive Resource Algorithm
EQVMP	Energy Efficient and QoS-Aware Virtual Machine Placement Algorithm
FFD	First Fit Decreasing Algorithm
GA	Genetic Algorithm
IaaS	Infrastructure as a Service
MIPS	Millions of Instructions per Second
NIST	National Institute of Standard Technology
OVS	Open vSwitch
PaaS	Platform as a Service
PAVA	Priority Aware VM Allocation Algorithm
PM	Physical Machine
PSO	Particle Swarm Optimization
QoS	Quality of Service
RAM	Random Access Memory
SaaS	Software as a Service
SDN	Random Access Memory
SDN-Cloud	Software-Defined Networks
SLA	Service Level Agreement
VM	Virtual Machine

Therefore, the time complexity is O(n^2) or O(n log n) depending on the adjacency structure, and the space complexity is O(1).

VII. CONCLUSION

In this paper, we developed a two-phase VM allocation algorithm that leverages fuzzy logic for SDN-Cloud heterogeneous networks in cloud datacenter. The proposed algorithm focuses on multiple aspects like the utilization of both PM and VM and takes into account multiple resources, such as processing capacity and memory for host-VM mapping. The first phase incorporates a fuzzification step, wherein adjacent hosts are grouped, and resources (processing capacity and memory) are extracted. A ratio for each host is calculated and the hosts are sorted based on this assigned ratio. This is followed by a defuzzification phase, in which the process of selecting the best fir host begins by calculating the requested resource ratio of the requested VM. An inference rule is built by comparing the resource ratio of the VM to the nominated host. The available resources of the host are then updated if the allocation successfully is performed. We have evaluated our work in CloudSimSDN with an overloaded network using complex applications, such as Wikipedia workloads, to represent real-world traffic in large scale datacenters. We measure response time as an indication of network performance in conjunction with serve time of the network and CPU. QoS violation rate and power consumption for both hosts and switches are also considered. Results reveal that our proposed algorithm succeeds in reducing power consumption by 40% while preserving network performance in heavy-loaded networks, compared to the baseline algorithm. Hence, D-Ph algorithm manages to achieve a balance between multiple performance metrics, including energy consumption, CPU utilization, and memory utilization. Thus, our proposed algorithm demonstrates its ability in minimizing energy usage, which leads to cost savings and environmental benefits, while maximizing CPU and memory utilization to ensure efficient resource usage without compromising network performance.

ACKNOWLEDGMENT

The author Amirah H. Alomari sincerely grateful to King Khalid University for their scholarship that enabled her academic journey and the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), for their support.

REFERENCES

[1] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems*. Harlow, U.K.: Addison-Wesley, 2012.
[2] L. F. Bitrencourt, E. R. M. Madeira, and N. L. S. da Fonseca, "Resource management and scheduling," in *Cloud Services Networking and Management*. Hoboken, NJ, USA: Wiley, 2015, pp. 243–267.
[3] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN," *Queue*, vol. 11, no. 12, pp. 20:20–20:40, Dec. 2013, doi: 10.1145/2559899.2560327.
[4] N. Almezeini and A. Hafez, "Review on scheduling in cloud computing," *Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 2, pp. 108–111, 2018.
[5] A. Alomari, S. K. Subramaniam, N. Samian, R. Latip, and Z. Zukarnain, "Resource management in SDN-based cloud and SDN-based fog computing: Taxonomy study," *Symmetry*, vol. 13, no. 5, p. 734, Apr. 2021, doi: 10.3390/sym13050734.
[6] R. Czabanski, M. Jezewski, and J. Leski, "Introduction to fuzzy systems," in *Theory and Applications of Ordered Fuzzy Numbers*. Cham, Switzerland: Springer, 2017, pp. 23–43.
[7] N. M. Gonzalez, T. C. M. D. B. Carvalho, and C. C. Miers, "Cloud resource management: Towards efficient execution of large-scale scientific applications and workflows on complex infrastructures," *J. Cloud Comput.*, vol. 6, no. 1, Dec. 2017, p. 13.
[8] D. C. Marinescu, "Cloud energy consumption," in *Encyclopedia of Cloud Computing*, May 2010, pp. 301–314, doi: 10.1002/9781118821930.ch25.
[9] N. T. Nguyen, C. P. Lim, L. C. Jain, and V. E. Balas, "Theoretical advances and applications of intelligent paradigms," *J. Intell. Fuzzy Syst.*, vol. 20, nos. 1–2, pp. 1–2, 2009, doi: 10.3233/IFS-2009-0410.
[10] M. I. Tariq, S. Tayyaba, M. W. Ashrf, M. Imran, E. Pricop, O. Cangea, N. Paraschiv, and N. A. Mian, "An analysis of the application of fuzzy logic in cloud computing," *J. Intell. Fuzzy Syst.*, vol. 38, no. 5, pp. 1–15, 2020.
[11] G. J. Klir, B. Yuan, and R. L. O. Ceurvorst, Eds., *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.
[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[13] P. M. Mell and T. Grance, "SP 800-145. The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-145, 2011.
[14] D. C. Marinescu, *Cloud Computing Theory and Practice*. Waltham, MA, USA: Elsevier, 2013.
[15] *OpenStack Administrator Guide*, OpenStack Found., Ussuri, Austin, TX, USA, May 2020.
[16] P. Culver, *Software Defined Networks*, 2nd ed. San Mateo, CA, USA: Morgan Kaufmann, 2016.
[17] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014, doi: 10.1109/SURV.2014.012214.00180.
[18] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven SDN controller architecture," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Sydney, NSW, Australia, 2014, pp. 1–6, doi: 10.1109/WoWMoM.2014.6918985.
[19] N. L. S. da Fonseca and R. Boutaba, "Openflow and SDN for clouds," in *Cloud Services, Networking, and Management*. Piscataway, NJ, USA: IEEE, 2015, pp. 129–152. [Online]. Available: https://www.opendaylight.org/what-we-do/odl-platform-overview, doi: 10.1002/9781119042655.ch6.
[20] J. Son and R. Buyya, "Priority-aware VM allocation and network bandwidth provisioning in software-defined networking (SDN)-enabled clouds," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 17–28, Jan. 2019, doi: 10.1109/TSUSC.2018.2842074.
[21] S.-H. Wang, P. P.-W. Huang, C. H.-P. Wen, and L.-C. Wang, "EQVMP: Energy-efficient and QoS-aware virtual machine placement for software defined datacenter networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Feb. 2014, pp. 220–225.
[22] Z. Ding, Y.-C. Tian, M. Tang, Y. Li, Y.-G. Wang, and C. Zhou, "Profile-guided three-phase virtual resource management for energy efficiency of data centers," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2460–2468, Mar. 2020, doi: 10.1109/TIE.2019.2902786.
[23] R. Wang, R. Esteves, L. Shi, J. A. Wickboldt, B. Jennings, and L. Z. Granville, "Network-aware placement of virtual machine ensembles using effective bandwidth estimation," in *Proc. 10th Int. Conf. Netw. Service Manage. (CNSM) Workshop*, Rio de Janeiro, Brazil, Nov. 2014, pp. 100–108, doi: 10.1109/CNSM.2014.7014146.
[24] R. Wang, J. A. Wickboldt, R. P. Esteves, L. Shi, B. Jennings, and L. Z. Granville, "Using empirical estimates of effective bandwidth in network-aware placement of virtual machines in datacenters," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 2, pp. 267–280, Jun. 2016.
[25] R. Wang, S. Mangiante, A. Davy, L. Shi, and B. Jennings, "QoS-aware multipathing in datacenters using effective bandwidth estimation and SDN," in *Proc. 12th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2016, pp. 342–347.

[26] N. K. Sharma and G. R. M. Reddy, "Multi-objective energy efficient virtual machines allocation at the cloud data center," *IEEE Trans. Services Comput.*, vol. 12, no. 1, pp. 158–171, Jan./Feb. 2019, doi: 10.1109/TSC.2016.2596289.
[27] D.-M. Zhao, J.-T. Zhou, and K. Li, "An energy-aware algorithm for virtual machine placement in cloud computing," *IEEE Access*, vol. 7, pp. 55659–55668, 2019, doi: 10.1109/ACCESS.2019.2913175.
[28] A. Jumnal and S. M. D. Kumar, "Optimal VM placement approach using fuzzy reinforcement learning for cloud data centers," in *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV)*, Feb. 2021, pp. 29–35, doi: 10.1109/ICICV50876.2021.9388424.
[29] S. Alanazi and B. Hamdaoui, "Energy-aware resource management framework for overbooked cloud data centers with SLA assurance," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647884.
[30] R. Shaw, E. Howley, and E. Barrett, "A predictive anti-correlated virtual machine placement algorithm for green cloud computing," in *Proc. IEEE/ACM 11th Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2018, pp. 267–276, doi: 10.1109/UCC.2018.00035.
[31] J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon, and R. Buyya, "CloudSimSDN: Modeling and simulation of software-defined cloud data centers," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 475–484, doi: 10.1109/CCGrid.2015.87.
[32] D. Ersoz, M. S. Yousif, and C. R. Das, "Characterizing network traffic in a cluster-based, multi-tier data center," in *Proc. 27th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2007, p. 59, doi: 10.1109/ICDCS.2007.90.
[33] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. Van Gilder, "Understanding and abstracting total data center power," in *Proc. Workshop Energy-Efficient Design (WEED)*, 2009.
[34] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "CARPO: Correlation-aware power optimization in data center networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1125–1133.

AMIRAH H. ALOMARI received the bachelor's degree in computer science from Universiti Kebangsaan Malaysia (UKM), in 2014, and the master's degree in computer science from King Saud University (KSU), Saudi Arabia, in 2017. She is currently pursuing the Ph.D. degree in parallel and distributed systems with Universiti Putra Malaysia (UPM). She is also a Lecturer with King Khalid University, Saudi Arabia. Her research interests include computer networks, artificial intelligence, and network management.

SHAMALA K. SUBRAMANIAM (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in computer science from Universiti Putra Malaysia (UPM), in 1996, 1999, and 2002, respectively. She is currently a Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, UPM. Her research interests include computer networks, simulation and modeling, and scheduling and real-time systems.

NORMALIA SAMIAN (Member, IEEE) received the Diploma degree in engineering and the B.Sc. degree in computer science from Universiti Putra Malaysia (UPM), the M.Sc. degree in computer science with a specialization in wireless networks security from Universiti Teknologi Malaysia (UTM), in 2010, and the Ph.D. degree in cooperation in wireless multihop networks from UPM. She is currently an Assistant Professor with the Faculty of Computer Science and Information Technology, UPM. Her research interests include ad hoc networks security, trust management in MANETs, and game theoretical approaches for wireless multihop networks. She received the Best UPM Student Award, in 2006, and nominated for the Best Academic Award Candidate.

ROHAYA LATIP (Member, IEEE) received the bachelor's degree in computer science from Universiti Teknologi Malaysia, in 1999, and the M.Sc. degree in distributed systems and the Ph.D. degree in distributed database from Universiti Putra Malaysia (UPM). From 2011 to 2012, she was the Head of the HPC Section, UPM. She consulted the Campus Grid Project and the Wireless for Hostel in Campus UPM Project. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology, UPM. She is also the Head of the Department of Communication Technology and Networks and a Co-Researcher with the Institute for Mathematic Research (INSPEM). Her research interests include big data, cloud and grid computing, network management, and distributed database.

ZURIATI AHMED ZUKARNAIN (Member, IEEE) received the bachelor's and master's degrees in physics and education from Universiti Putra Malaysia (UPM), in 1997 and 2000, respectively, and the Ph.D. degree in quantum computing and communication from the University of Bradford, U.K., in 2005. Since 2001, she has been an Academic Staff with the Faculty of Computer Science and Information Technology, UPM. From 2006 to 2011, she was the Head of the Department of Communication Technology and Networks. From 2012 to 2015, she was also the Head of the Section of High-Performance Computing, Institute of Mathematical Research, UPM. At the faculty, she taught several courses for bachelor's students, such as data communication and networks, distributed systems, mobile and wireless, network security, computer architecture, and assembly language. For master's students, she taught a few courses, such as the advanced distributed and research method. Her research interests include computer networks, distributed systems, mobile and wireless networks, network security, quantum computing, and quantum cryptography. She is a member of the IEEE Computer Society.