Osanaiye, Opeyemi; Chen, Shuo; Yan, Zheng; Lu, Rongxing; Choo, Kim Kwang Raymond; Dlodlo, Mqhele

From Cloud to Fog Computing

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2017.2692960

Published: 01/01/2017

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K. K. R., & Dlodlo, M. (2017). From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework. *IEEE Access*, *5*, 8284-8300. [7896564]. https://doi.org/10.1109/ACCESS.2017.2692960

Received February 13, 2017, accepted April 5, 2017, date of publication April 12, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2692960

From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework

OPEYEMI OSANAIYE[1,2], (Member, IEEE), SHUO CHEN[3], ZHENG YAN[4,5], (Senior Member, IEEE), RONGXING LU[6], KIM-KWANG RAYMOND CHOO[7,1], (Senior Member, IEEE), AND MQHELE DLODLO[2]

[1]Information Assurance Research Group, University of South Australia, Adelaide, SA 5095, Australia
[2]Department of Electrical Engineering, University of Cape Town, Cape Town 7701, South Africa
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798
[4]School of Cyber Engineering, Xidian University, Xi'an 710071, China
[5]Department of Communications and Networking, Aalto University, 02150 Espoo, Finland
[6]Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada
[7]Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249-0631, USA

Corresponding author: Kim-Kwang Raymond Choo (raymond.choo@fulbrightmail.org)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800700, in part by NSFC under Grant 61672410 and Grant U1536202, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Program 2016ZDJC-06, in part by the 111 Project under Grant B16037 and Grant B08038, and in part by the Ph.D. Grant of Chinese Educational Ministry under Grant 20130203110006.

ABSTRACT Fog computing, an extension of cloud computing services to the edge of the network to decrease latency and network congestion, is a relatively recent research trend. Although both cloud and fog offer similar resources and services, the latter is characterized by low latency with a wider spread and geographically distributed nodes to support mobility and real-time interaction. In this paper, we describe the fog computing architecture and review its different services and applications. We then discuss security and privacy issues in fog computing, focusing on service and resource availability. Virtualization is a vital technology in both fog and cloud computing that enables virtual machines (VMs) to coexist in a physical server (host) to share resources. These VMs could be subject to malicious attacks or the physical server hosting it could experience system failure, both of which result in unavailability of services and resources. Therefore, a conceptual smart pre-copy live migration approach is presented for VM migration. Using this approach, we can estimate the downtime after each iteration to determine whether to proceed to the stop-and-copy stage during a system failure or an attack on a fog computing node. This will minimize both the downtime and the migration time to guarantee resource and service availability to the end users of fog computing. Last, future research directions are outlined.

INDEX TERMS Cloud computing, edge computing, fog computing, live VM migration framework, virtualization.

I. INTRODUCTION

Cloud computing can be an efficient alternative to owning and maintaining computer resources and applications for many organizations, particularly small- and medium-sized organizations, due to the pay-as-you-go model and other characteristics (e.g., on-demand, self-service, resource pooling and rapid elasticity) [1]. The continued interest in cloud computing has also resulted in other emerging cloud paradigms, such as fog computing. In fog computing, cloud elastic resources are extended to the edge of the network, such as portable devices, smart objects, wireless sensors and other Internet of Things (IoT) devices [5], [11], [13], [14], [109] to decrease latency and network congestion. IoT devices use interconnected technologies like Radio Frequency Identify (RFID) and Wireless Sensor and Actor Networks (WSAN) to exchange information over the Internet, and are more integrated in our daily life [2]. Smart-home, smart-city and smart-grid are examples of IoT applications, where sets of sensors are used to obtain information to improve the quality of life and quality of experiences. IoT is characterized by widely distributed objects known as "things" with limited storage and processing capacity to guarantee efficiency, reliability and privacy [3]. However, its applications require geo-distribution, mobility support, location-awareness and low latency [4] to efficiently collect and process data from IoT devices. This information is then used to perform detection and prediction for optimization and timely decision-making process.

Cloud and fog computing share overlapping features, but fog computing has additional attributes such as location awareness, edge deployment and a large number of geographically distributed nodes in order to offer a mobile, low latency and real-time interaction [3]. The deployment of both cloud and fog computing is primarily driven by virtualization technology, which introduces a software abstraction between the computer hardware and the operating system (OS) and application running on the hardware [6]. This abstraction layer is also known as a Virtual Machine Monitor (VMM) or hypervisor. The VMM acts as a controller of hardware resources and enables multi-tenancy by allowing multiple OS to co-exist on the same physical hardware and share resources. Despite the benefits afforded by such architecture, cloud services are susceptible to a range of security and reliability risks. Concerns about attacks or risks affecting availability of cloud resources are identified in the literature as one of the factors hindering the general adoption of cloud computing [93].

Therefore, live migration of virtual machines (VM) has been proposed to mitigate malicious attacks, infrastructural and component failures. Live migration involves a dynamic transfer of a VM from one physical machine to another that is transparent to the guest OS, the application running on the OS, and remote users of the VM [6]. Two predominant techniques are pre-copy live migration and post-copy live migration [7], [8], [10]. The former involves the transfer of memory contents of the VM from a source to a target through several iterations before the VM is restarted; whilst the latter only sends the virtual central processing unit (vCPU) and the device state to the target at an initial stage [9]. Subsequent pages are fetched on demand while the VM is running on the target host. The key performance metrics in VM migration are downtime and total migration time [10].

This paper presents a detailed review of fog computing, its architecture and applications. Furthermore, we present the security, privacy and resource availability challenges and propose a conceptual smart pre-copy VM live migration framework to mitigate malicious attacks or failure of physical servers which result in unavailability of services and resources. Specifically, we review existing fog computing literature published between January 2012 and December 2016. The publications were located using keyword search on Google Scholar and other academic databases, such as ScienceDirect, Springer, IEEE Xplore, and ACM digital Library. The keywords we used included "fog computing" "cloud computing", "edge computing" and "VM live migration". The rest of the paper is organized as follows. In Sections II and III, we provide an overview of fog computing and present our taxonomy of fog computing applications, respectively. In Sections IV and V, we discuss several fog computing security and privacy challenges, and resource availability challenges. In Section VI, we present a general discussion, and in Section VII, we present a conceptual framework of smart pre-copy VM live migration approach. Finally, Section VIII concludes the paper and outlines future research opportunities.

II. FOG COMPUTING

The popularity of IoT applications and the increased digitalization of our society where millions to billions of smart devices (e.g., in smart homes, smart cities, smart metering systems, intelligent vehicles and large-scale wireless sensor networks) are constantly exchanging information over the Internet have resulted in large volumes of data that need to be managed and processed. To achieve this, cloud computing is a popular option due to its scalability, storage, computational and other capabilities to support the provisioning or de-provisioning of resources according to user requirements in real-time [5], [122]. However, in recent years, fog computing has been proposed to extend the cloud computing paradigm from the core to the edge of the network. It presents a highly virtualized platform that provides computational, networking and storage services between cloud computing and end devices [11]. For example, Zhu *et al.* [14] describe fog computing as an enabler of smart applications and Internet services (including cloud) for data management and analytics. Song *et al.* [117] construct a system model of fog computing by combining its features and that of graph theory to propose a dynamic load balancing mechanism based on the graph repartitioning.

A. FOG COMPUTING ARCHITECTURE AND FEATURES

Fog computing has a distributed architecture that targets services and applications with widely dispersed deployments [13]. Different fog computing architectures have been proposed in the literature. For example, Sarkar *et al.* [29] described a three-tier architecture where the bottom tier comprises several terminal nodes (TN) (e.g., smart device and wireless sensor nodes) that transmit information to the upper tiers. Tier two is the middle tier (also referred to as the fog computing layer) comprising highly intelligent devices, such as routers, switches and gateways. The third and uppermost tier is referred to as the cloud computing tier that has several high-end servers and data center(s). Shi *et al.* [30] presented a simple fog architecture comprising of fog nodes in between cloud components and end devices. Similar to the architecture presented in [30], Lee *et al.* [31] described a hierarchical fog computing architecture consisting of three components, namely: IoT nodes, fog nodes and back-end Cloud. Zhu *et al.* [22] described the Cisco overview of fog computing architecture by presenting a three-layered approach consisting of distributed intelligence end-point computing (i.e., smart things network, embedded systems and sensors), distributed intelligence fog computing (i.e., multi-service edge and filed area network), and centralized intelligence cloud computing (i.e., data center cloud and core).

Bonomi *et al.* [12] presented a fog computing architecture comprising homogeneous physical resources, fog abstraction layer and a fog service orchestration layer (see Fig. 1). Heterogeneous physical resources consist of components such as servers, edge routers, access points, set-up boxes and end-devices with different storage and memory capacities to support additional functionalities. The platform is hosted on different OSs and software applications, thus having a wide range of software and hardware capabilities.

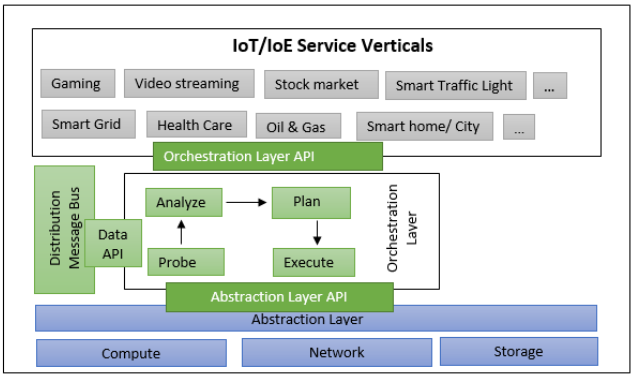

FIGURE 1. Architecture and components of fog computing (adapted from [12]).

The fog abstraction layer provides a generic application programming interface (API) for monitoring resources such as CPU, memory and network by hiding the platforms' heterogeneity and unveiling the uniform and programmable interface for seamless resource management and control – see Fig. 1. It supports virtualization and enables multiple OSs to co-exist on a single physical machine to ensure efficient use of resources. The multi-tenancy feature ensures the isolation of different tenants on the same physical machine.

The orchestration layer provides a dynamic and policy-based life cycle for managing fog services. The orchestration functions by providing a distributed approach as the underlying fog infrastructure and services [12]. The fog orchestration layer consists of a small software agent (hereafter referred to as foglet). Foglet is used to monitor the current state of the deployed fog nodes by presenting a wide range of capabilities using components such as software agent, distributed storage, scalable message bus and distributed policy engine. The orchestration layer API performs four basic functions, namely: probing and application of data, analyzing the retrieved data, managing requests by planning and allocation of resources, and enforcing decision [18]. The fog platform hosts different applications such as smart cities and smart grids.

Fog computing provides an improved quality-of-service (QoS), low latency and location awareness to mobile nodes through edge routers and access points. The latter, for example, can be positioned along highways and tracks to provide resources and services to applications that are latency sensitive (e.g., gaming, video streaming, real-time traffic monitoring systems, and emergency healthcare services). A common characteristics associated with fog computing is its deployment at the "edge of the network" [64], [94], which implies that fog computing has features that make it a non-trivial extension of cloud computing. We highlight some of these key features below:

- Heterogeneity: Fog computing is a virtualized platform that offers computational, networking and storage services between cloud computing and end devices. Its heterogeneity feature serves as a building block as it exists in different forms and can be deployed in wide-ranging environments.
- Geographical distribution: Fog computing has a widely distributed deployment in order to deliver high-quality services to both mobile and stationary end devices.
- Edge location, location awareness and low latency: The emergence of fog computing is partly due to the lack of support for endpoints with quality services at the edge of the network. Examples of applications with low latency requirements are video streaming in real-time closed-circuit television monitoring and gaming.
- Real-time interaction: Various fog applications, such as real-time traffic monitoring systems, demand real-time processing capabilities rather than batch processing.
- Support for mobility: Mobility support is essential for many fog computing applications to enable direct communication with mobile devices using protocols such as Cisco's Locator/ID Separation Protocol that decouples host identity from location identity using a distributed directory system [14]
- Large-scale sensor networks: This is applicable when monitoring the environment or in smart grid using inherently distributed systems that require distributed computing and storage resources.
- Prevalent to wireless access: Wireless access points and cellular mobile gateway are typical examples of a fog network node.
- Interoperability: Fog components must be able to interoperate to ensure support for wide range of services like data streaming.

Su *et al.* [21] proposed a Steiner tree approach based on a caching scheme, where fog servers initially produce a Steiner tree when caching resources to minimize total path, weight and cost, in order to reduce resource caching costs. The comparison between the workings of the Steiner tree in fog computing and the traditional shortest part scheme suggested that the former achieves better efficiency. Zhu *et al.* [22] deployed fog computing to process and transmit video applications and services, ranging from proxy-assisted rate adaptation to intelligent caching for on-demand video streaming. This enhances the quality of experience (QoE) and virtual desktop infrastructure interactive system of real-time video for surveillance cameras. Truong *et al.* [23] proposed a new Vehicular Adhoc Network (VANETs) architecture by combining Software Define Network (SDN) and fog computing to offer an optimized low-latency deployment. Gazis *et al.* [32] presented an industrial context of deploying fog computing by introducing an adaptive operational platform to provide an end-to-end manageability for fog computing infrastructure, according to the operational requirements of the individual process. Femtocloud systems were proposed in [27] to offer a dynamic, self-configuring and multi-device mobile cloud from a cluster of mobile devices to provide cloud services at the edge. The evaluations suggested that the approach can provide reasonably efficient computational capacity. In advanced metering infrastructure, the amount of collected and processed data has increased exponentially, therefore, the centralized cloud approach is no longer adequate. Yan and Su [118] proposed using fog computing in existing smart meter infrastructure to provide a reliable and cost-effective solution.

B. INTERACTION BETWEEN FOG COMPUTING, CLOUD COMPUTING AND INTERNET OF THINGS

Fog computing brings cloud computing closer to Internet of Things (IoT) devices [2]. The advent of IoT has resulted in an increasing number of use cases that generate significant volume of data, compounding the challenges of dealing with big data from a number of geographically distributed data sources [12]. To efficiently analyze these time-sensitive data, fog computing was proposed. To harness the benefits of IoT and speed up awareness and response to events, we require a new set of infrastructures as current cloud models are not designed to handle the specifics of IoT (i.e., volume, variety and velocity of data) [15]. Specifically, billions of previously unconnected devices are now generating over two exabytes of data every day and it has been estimated that by 2020, 50 billion "things" will be connected to the Internet [15]. Therefore, fog computing has been identified as a viable solution.

Sehgal *et al.* [24] proposed a framework that combines IoT, cloud computing, and fog computing for smart human security. This framework provides a wearable computing system by harnessing the pervasive nature of IoT, omnipresence feature of cloud, and the extension of fog computing to provide security cover for people. In a similar vein, Yannuzzi *et al.* [25] integrated fog computing and cloud computing by considering mobility, reliability control and actuation, and scalability to demonstrate that fog computing can be used as the underlying platform for IoT applications. Suciu et al. [26] presented an architecture for secure E-health applications using big data, IoT, and cloud convergence to enable telemonitoring. This approach uses CloudView Exalead as a search platform that offers access to information present in the infrastructural level for search based application online and at the enterprise level. Cirani *et al.* [28] proposed a fog node and IoT hub, distributed on the edge of multiple networks to enhance network capability by implementing border router, cross-proxy, cache, and resource directory. IoT operates at both the link layer and application layer to enable resource discovery and seamless interactions among applications.

Table 1 summarizes the features associated with fog computing, cloud computing, and IoT.

TABLE 1. Summary of fog computing, cloud computing, and IoT features.

Features	Fog computing	Cloud computing	Internet of Things
Target User	Mobile users	General Internet users	Stationary and mobile devices
Number of server nodes	Large	Few	Large
Architecture	Distributed	Centralised	Dense and distributed
Service Type	Localized information service limited to specific deployment location.	Global information collected worldwide	Information specific to the end device
Working Environment	Outdoors (i.e., streets, fields, tracks) or Indoor (i.e., home, malls, restaurants)	Indoors with massive space and ventilation	Outdoor and Indoor
Location awareness	Yes	No	Yes
Real-time interactions	Supported	Supported	Supported
Mobility	Supported	Limited Support	Supported
Big data and duration of storage	Short duration as it transmits big data	Months and years as it manages big data	Transient as it is the source of big data.
Major service provider	Cisco IOx	Amazon, Microsoft, IBM	ARM, Atmel, Bosch

III. PROPOSED FOG COMPUTING APPLICATION TAXONOMY

Different fog computing applications have been suggested in the literature, therefore, in this section, we present a taxonomy of such applications.

Luan *et al.* [16] described fog as a surrogate of cloud that can be used to deliver location-based service application to mobile device users (e.g., showcasing its application in shopping centers, parklands, inter-state bus, and vehicular fog computing networks). Boronmi *et al.* [17] demonstrated the role of fog computing in three scenarios, namely: connected vehicle, smart grid and wireless sensor and actuator networks. Dsouza et al. [18] used the Smart Transport System (STS) as a use case, where STSs are heterogeneous distributed systems designed to constantly monitor traffic activities and transmit data between commuters and smart devices in real-time to pre-empt traffic and safeguard commuters. Dastjerdi *et al.* [19] demonstrated the application of fog computing in healthcare, highly latency intolerant augmented reality domain and its use for improving website performance by caching and pre-processing. Saharan and Kumar [20] identified four areas of fog computing's application, namely: wireless and actuator networks, smart grid, smart traffic lights and connected vehicles, and IoT. Kitanov *et al.* [120] proposed a hybrid environment service orchestration that provides resilient and trustworthy fog computing service beyond the 5G network.

In our taxonomy, we categorize fog computing applications into real-time and near real-time applications – see Fig. 2. Fog computing can also be introduced in a network (for non-real-time application) to reduce the amount of traffic in the core; however, this is beyond the scope of this work.

Real-time applications are low-latency and function within a pre-defined timeframe which user senses as immediate or current. Near real-time applications, on the other hand, are those that are subject to time delay introduced by data processing or network transmission between the moment

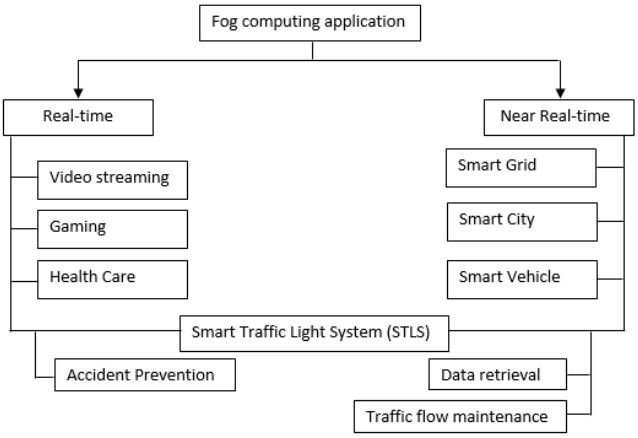

FIGURE 2. **Proposed fog computing application taxonomy.**

an event occurs and the use of the processed data [109]. Near real-time is often determined by subtracting the current time from the processing time that is nearly the time of the live event. In this section, we present popular use cases of both real-time and non-real-time applications.

A. REAL-TIME USE CASES

1) VIDEO STREAMING

Transmissions of video applications and services are more efficient in a fog computing implementation, due to the capability of fog computing to provide location awareness, low latency, mobility, and real-time analytics. Several smart devices support smart surveillance that can be used by law enforcement officers to display live video streams of events of interest. For example, Hong et al. [33] described a video surveillance application that requires a three-level hierarchy system to perform motion detection with smart camera, face recognition with fog computing instances, and identity aggregation with cloud computing instances. Magurawalage et al. [34] proposed Aqua computing, inspired from water cycle, which can take the form of either fog or cloud computing. The proposed architecture consists of clones placed at the edge of the network that serve end users in a video streaming scenario to act as a buffer. Zhu et al. [22] used fog computing to transform video applications and services to support on-demand video delivery. Such an approach enhances interactions in a virtual desktop infrastructure system and provides real-time video analytics for a surveillance camera. Other potential benefits of deploying fog computing to improve video streaming performance such as intelligent caching and adaptive streaming were also highlighted. Foerster *et al.* [35] identified key requirements of fog computing that complements cloud computing to support an intelligent network node. This helps to improve the quality of transmitted video by ensuring an intelligent soft handoff of mobile user and radio-aware resource management.

2) GAMING

The advent of cloud computing has provided a platform for computer gaming without users (players) worrying about hardware requirements. Cloud gaming providers in recent times have been rapidly expanding or leveraging cloud infrastructure to provide game-on-demand (GoD) service to users over the Internet. It is offered remotely by enabling an interactive gaming that can be accessed and decoded by end devices such as smartphones or tablets. Wang and Dey [40] described a cloud server based mobile gaming approach, cloud mobile gaming, where most of the workload for executing the game engine are placed on the cloud server. The mobile device only sends and receives user gaming commands to and from the servers. Zhou *et al.* [37] identified faster response time and higher QoS as key goals to be achieved in ensuring high gaming QoE. Due to the stringent requirements of gaming, cloud gaming is inherently susceptible to latency due to game graphics being rendered online. Lee *et al.* [38] investigated how the response latency in cloud gaming would affect user experience and how it varies between games. Then, a model was developed on how to predict the real-time strictness of a game based on players' input and game dynamics.

Having established the impact of latency on cloud gaming and the inability of cloud to meet the stringent latency requirements, Choy *et al.* [39] proposed a new hybrid platform by extending the existing cloud infrastructure and deploying more diverse geographically distributed devices equipped with specialized resources. To guarantee a high QoE in cloud gaming due to the high popularity of Massively Multiplayer Online Gaming (MMOG), Lin and Shen [41] proposed a lightweight system, which consists of super nodes that extend video games to nearby players to significantly reduce latency and bandwidth consumption. A receiver-driven encoding rate adaptation was also proposed to increase the playback continuity and deadline-driven buffer scheduling strategy. The experimental result obtained from PlanetLab and PeerSim demonstrated the efficiency and effectiveness of the system deployment.

3) HEALTHCARE

IoT applications have provided a structured approach towards improving our health care services. This is achieved by deploying ubiquitous monitoring systems and transmitting the data to fog devices in real-time before sending the information to the cloud for further analysis and diagnosis. Gia *et al.* [46] utilized fog computing as a smart gateway to provide sophisticated techniques and services such as distributed storage and embedded data mining. A case study of electrocardiogram feature extraction that plays a vital role in the diagnosis of cardiac diseases was presented. The experimental result suggested that deploying fog computing achieves a low latency and real-time response with more than 90% bandwidth efficiency. Persuasive health monitoring is one of the key application areas of biomedical big data research for making early predictions to support smart healthcare decision making. Cao *et al.* [47] proposed a real-time fall detection algorithm, U-Fall, which consists of three major modules, front-end, back-end and communication module. Both front-end and back-end make independent detection results. However, a collaborative detection will increase the accuracy and reduce the false alarm rate. An experiment demonstrating the use of the U-Fall algorithm in fog computing that automatically detects pervasive fall during health monitoring to mitigate stroke was presented. Results obtained suggested that a high sensitivity and specificity was achieved. Similar to the work in [47], FAST, a distributed analytics system based on fog computing to monitor and mitigate stroke, was proposed in [48].

In order to facilitate easy access to healthcare service for the elderly, a body sensor network in fog computing was proposed in [49]. The fog computing gateway is used to enhance the system by offering different services such as ECG feature extraction, distributed database and graphical interface to ensure obtained health data are visualized and diagnosed in real time. Aazam and Huh [50] proposed a smartphone-based service, Emergency Help Alert Mobile Cloud (E-HAMC), which uses fog services for pre-processing and offloading purposes to provide an instant way of notifying relevant emergency department (e.g., ambulance) from the stored contact details. This service also sends the incident location to facilitate patient tracing.

Dubey *et al.* [51] proposed and evaluated the use of fog data in carrying out data mining and analytics on raw data collected from different wearable sensors used for telehealth applications. Ahmad *et al.* [52] deployed fog computing as the intermediary layer between cloud and end users in their framework. A security solution, cloud access security broker (CASB), was also introduced as an integral part of health fog to implement certain security policies.

4) SMART TRAFFIC LIGHT SYSTEM (STLS)

Smart traffic lights interact locally with a number of sensor nodes to detect the presence of cyclists, bikers or pedestrians, as well as estimating the speed and distance of approaching vehicles [17]. This information can be used to prevent accidents by sending early warning signals to approaching vehicles. Stojmenovic and Wen [36] described the use of video camera that senses the presence of an ambulance flashing light during an emergency to automatically change street lights and allow the emergency vehicle to pass through traffic. Bonomi *et al.* [12] identified three major goals of STLS, namely: accident prevention, steady traffic flow maintenance, and retrieval of relevant data to evaluate and improve the system. Accident prevention is a real-time process, while traffic flow and data retrieval are regarded as near real-time and batch processes. Wireless access points and smart traffic light units are deployed along the roadside to provide communication such as vehicle-to-vehicle, vehicle to access point, access point to access point (see Fig. 3).

B. NEAR REAL-TIME USE CASES

1) SMART GRIDS

The current call for smart grids can be linked to the fact that the present-day energy demands have outpaced the rate at which energy is generated by conventional methods as well as the need to reduce gas emission to control or curtail climate change [42]. Abdelwahab *et al.* [43] proposed a cloud-assisted remote sensing approach to measure and collect smart grid operational information to enable seamless integration and automation of smart grid components. Cloud computing feature that uses a centralized demand response scheme, where customers and suppliers communicate directly with the cloud has proven to be bandwidth inefficient. Therefore, Stojmenovic [44] proposed a distributed approach by presenting a macro-grid and micro-grid to act as fog devices. Customers communicate with the nearby fog devices rather than the remote cloud. Fog devices, on the other hand, communicate frequently with the customers and occasionally with the cloud. Vatanparvar and Al Faruque [45] presented a Cyber-Physical Energy System (CPES) to improve the efficiency, reliability and performance of power grid by managing demand and supply dynamics intelligently. A prototype of this was implemented in fog computing platform to support interoperability, scalability and remote monitoring.

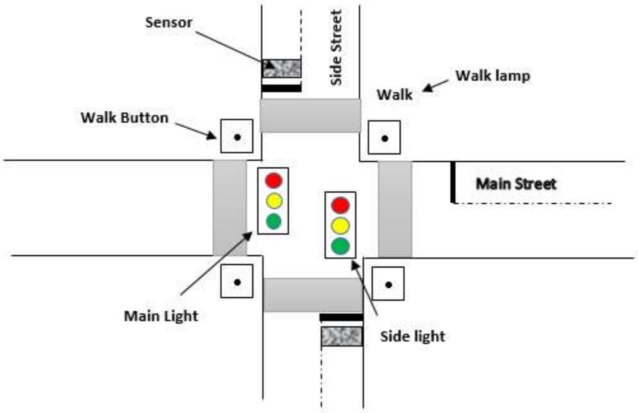

FIGURE 3. Smart Traffic Light System (adapted from [95]).

2) SMART CITIES

A smart city is one key IoT application that ranges from smart traffic management to energy management of buildings, etc. The smart city concept has drawn great interest from both science and engineering sectors, and from both research and practitioner communities, as a means to overcome challenges associated with rapid urban growth. Kitchin [56] described smart city as a city that is vastly controlled and made up of ubiquitous computing whose economy and governance are driven by innovation and creativity. However, some of these IOT applications and devices in a smart city require high computation and storage capacities, and pose interoperability challenges. For example, Byers and Wetterwald [55] identified the complexity associated with a cloud centralized architecture involving smart city that consists of road traffic control, parking lot management and environmental monitoring over a distributed territory.

Yi *et al.* [53] identified fog computing that is close to the edge of the network as the solution as well as integrating all components in a unified platform to enable smart home applications with elastic resources. Smart city was described in [54] as a public space in the edge that optimizes energy consumption and improve the quality of life of citizens. In the work of Tang *et al.* [70], a hierarchical distributed fog computing that supports a huge number of infrastructural component and services for future smart cities was presented. A smart pipeline monitoring system use case was discussed, which is based on fiber optic sensors. Sequential learning algorithm was used to detect events threatening pipeline safety.

3) SMART VEHICLES

The advent of mobile cloud computing has necessitated the study of its agents such as vehicles, robots and humans that interact together to sense the environment, process the data and transmit the results. Lu *et al.* [59] described connected vehicle that communicates with their internal and external environment such as Vehicle-to-Vehicle (V2V), Vehicle-to-Sensor on-board (V2S), Vehicle-to-Road infrastructure (V2R) and Vehicle-to-Internet (V2I). Vehicle cloud has been identified [57] as the leading application that facilitates safe driving, urban sensing, content distribution and intelligent transportation to render benefits such as sensing urban congestion and collaborative reconstruction of footage in a crime scene.

A significant attribute of vehicular cloud as compared to the Internet cloud is its reliance on the sensors they carry, rather than cloud computing resources. Hou *et al.* [116] described a vehicular fog computing that utilizes vehicles as an infrastructure for computing and communication that involves the collaboration of many end-user clients or near-user edge devices. Lee *et al.* [119] described a vehicular fog as the equivalent of Internet cloud in vehicles and the core system environment that will enhance autonomous driving. VANET is a mobile ad-hoc network that uses vehicles as mobile nodes. Truong *et al.* [23] proposed a new architecture for VANET by combing SDN and fog computing to cater for future VANET demands and support surveillance services by considering resource manager and fog orchestration models. Kim *et al.* [58] presented a solution to insufficient parking space as a result of rapidly increasing number of vehicles by proposing a shared parking model in a vehicular network using both fog and cloud environments. Simulation results indicated a high efficiency and reliability in determining vacant parking slot.

IV. FOG COMPUTING SECURITY AND PRIVACY CHALLENGES

Security assessment of fog computing can be guided by the confidentiality, integrity and availability (CIA) triad model [1], which are the critical components that must be considered during the design and deployment of a system. While confidentiality and integrity are closely related to data privacy, availability entails the ability to remotely access resources offered by cloud servers and fog nodes when needed.

Apart from the security challenges fog computing inherited from the cloud, its heterogeneous feature and deployment location(s) at the edge of the network have made it susceptible to some additional challenges. Potential issues likely to be encountered with the deployment of fog computing identified by Yi *et al.* [11] are authentication, access control, intrusion attack and privacy. Vaquero and Rodero-Merino [63] predicted that the current security issues associated with a virtualized environment would be a potential security concern for fog devices hosting applications. Zhanikeev [60] identified challenges associated with hardware and platform standardization required for homogeneity to facilitate federation. Wang *et al.* [61] demonstrated that a man-in-the-middle attack could compromise and replace a genuine gateway before inserting malicious codes into the system.

In this section, we present an overview of security and privacy issues as applicable to the use cases.

A. SECURITY ISSUES IN FOG COMPUTING

The shareability and distributed feature of fog computing have made authentication a key issue when offered to a large number of end devices by front fog nodes. Security solutions proposed for cloud computing will not directly suit fog computing as its working surroundings may face threats that do not exist in a typical cloud deployment. Authentication takes place during the process of establishing a connection to ascertain the accessing rights and identity of a connecting node. Stojmenovic and Wen [36] identified authentication at different levels of the gateways as the main security issue in fog computing. Authentication and authorization issues in the context of smart grid and machine-to-machine communication for fog computing were presented in [62].

Zuo *et al.* [115] presented a chosen ciphertext attack (CCA) on fog computing and proposed a solution by first presenting the CCA security model of OD-ABE (attributed-based encryption with outsourced decryption) prior to describing their CCA-secure OD-ABE scheme. Roman *et al.* [64] presented a threat model by reviewing the scope and nature of potential attacks. They identified the most important asset at the edge, predicted possible attacks that can be directed towards such asset, and categorized potential target into network infrastructure, service infrastructure (edge data center and core infrastructure), virtualized infrastructure and user devices. Different devices and communication elements deployed in fog computing range from wireless to Internet-connected mobile devices, etc. Therefore, the attacker can target any of these components. Denial of Service (DoS), man-in-the-middle attacks, and rogue gateway attacks were identified as possible attacks on network infrastructure, while service infrastructure at the edge data center can be exposed to physical damage, privacy leakage, privilege escalation, sabotage, service manipulation and rogue datacenter, etc. For core service infrastructure, privacy leakage, service

TABLE 2. Threat model distribution for fog computing component (adapted from [64]).

Fog components Security issues	Network Infrastructure	Service Infrastructure (edge datacentre)	Service Infrastructure (core infrastructure)	Virtualization infrastructure	User Devices
DoS	✓			✓	
Man-in-the-middle	✓				
Rogue component (i.e., datacentre, gateway or infrastructure)	✓	✓	✓		
Physical damage		✓			
Privacy leakage		✓	✓	✓	
Privilege escalation		✓		✓	
Service or VM manipulation		✓	✓	✓	✓
Misuse of resources				✓	
Injection of information					✓

manipulation and rogue infrastructure have been identified as possible security threats [64]. Virtualized infrastructure within the core of all edge data center is vulnerable to misuse and exploits associated with DoS, primary leakage, privilege escalation and VM manipulation. Finally, user devices can be subjected to security issues with regards to injection of information and service manipulation. Table 2 summarizes the threat model distribution in fog computing component as identified in [64].

To mitigate some of the security issues presented, strategies such as multicast authentication using Public Key Infrastructure (PKI) [65] and deployment of intrusion detection system (IDS) [4] were suggested. A decoy information technology technique was proposed by Stolfo et al. [66] to withstand malicious insiders by disguising information to prevent attackers from identifying customer's real sensitive data.

B. PRIVACY IN FOG COMPUTING

Shankarwar and Pawar [67] defined privacy as the protection of data-in-transit from passive attacks to ensure sensitive information are not accessed or disclosed to an unauthorized person. Typical of most public remote storage facilities, sensitive and personal information outsourced to or stored in cloud computing could be compromised or leaked. In addition, researchers have also raised concerns about the far-reaching arm of legislation such as the PATRIOT Act for U.S.-based cloud service providers [110]–[112]. Fog computing, on the other hand, presents a higher privacy risk as the deployment is extended to the edge of the network. Yi *et al.* [4] explained that privacy risks such as data privacy, usage privacy and location privacy exist in the fog computing nodes located in the vicinity of the end users, and these nodes are more susceptible to information theft when compared with cloud servers located at the core of the network.

Dong *et al.* [68] identified from existing literature that sensor networks are vulnerable to content-based privacy threats and context-based privacy threats. They then proposed a redundant fog loop to preserve the location privacy of the source node to confuse the adversary from accurately determining the real source node. To mitigate malicious eavesdropping on data-in-transmit, Kulkarni *et al.* [69] proposed a fog friendly framework based on public key encryption with an infrequent key update to avoid high overhead. Lopez *et al.* [54] also proposed the use of attribute-based encryption and deployment of secure middleware for privacy-aware information sharing, with the aims of preventing service providers from accessing users' data without authorization. Privacy issues in smart grid were presented in [71], and a privacy-preserving aggregation scheme using multidimensional data aggregation approach based on homomorphic Paillier cryptography was proposed.

C. INFRASTRUCTURAL FAILURE IN FOG COMPUTING

To ensure availability of fog computing service and resources, the fog architecture must ensure reliability and resilience. Yi *et al.* [11] discussed the reliability improvement of fog computing by periodically carrying out check-pointing to resume after failure and rescheduling failed tasks or replicating to exploit executing in parallel. Due to the dynamic nature of fog computing, check-pointing and rescheduling may not be a good fit as this may introduce some latency and cannot adapt to changes.

V. RESOURCE AVAILABILITY IN FOG COMPUTING

The high availability of cloud and fog computing resources is essential as impending attacks or failure of its infrastructure rely on rule-of–thumb by over-provisioning resources to achieve availability [96]. Fog computing is characterized by geographically distributed nodes that depend on cloud servers, storage and network. Depending on the capacity of the cloud, over-provisioning might be minimal and can have a direct impact on the cost and the performance of other deployed user applications. This could result in a breach of the service level agreement (SLA), a binding agreement between providers and users, if the resource availability drops below the pre-agreed threshold.

Fig. 4 provides a snapshot of the availability distribution in fog and cloud computing according to their capacity.

A. FACTORS AFFECTING AVAILABILITY IN FOG COMPUTING

When determining availability of fog services, security, application failure, and infrastructural failure are three main factors to be considered. Security issues such as malicious

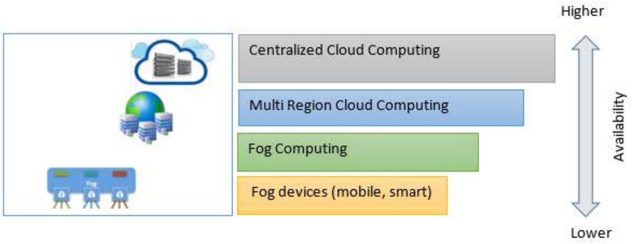
Higher
Centralized Cloud Computing
Multi Region Cloud Computing
Fog Computing
Fog devices (mobile, smart)
Availability
Lower

FIGURE 4. Availability distribution in fog and cloud computing paradigm.

attacks from either an internal or external source can consume significant resources and network bandwidth and disrupt the high availability of fog services to legitimate end users (e.g., successful distributed denial of service attacks [97]). Application and infrastructural failure of cloud and fog components can either be physical, human, and/or operational, which can be a result of system failure, network failure, power cut, design error or software bug.

B. MEASURING AVAILABILITY IN FOG COMPUTING

To measure availability in fog computing, two key reliability metrics can be utilized, namely: mean time to failure (MTTF) and mean time to repair (MTTR). During a component failure, resources and services offered are unavailable for use unless restored. MTTF is the average time estimated by the hardware manufacturer before a failure of the hardware module. For software, MTTF can be determined by multiplying the defect rate with thousands of line codes executed per seconds. MTTF only unveils one side of the coin. To determine the time to repair a failed component, MTTR is used [98]. For a hardware module, the MTTR is the mean time to replace a failed hardware while software MTTR can be determined by computing the time taken to reboot after detecting software fault. Measuring the rate of availability of fog computing can be determined using the following:

$$\text{Availability} = \text{MTTF}/(\text{MTTF} + \text{MTTR}).$$

C. VM MIGRATION IN FOG COMPUTING

The introduction of fog computing in high-performance environment increases the number of deployed nodes, which has a corresponding effect on the number of reported faults [72]. The high availability of fog and cloud resources is essential as an ongoing or successful attack or a failure of infrastructure can be catastrophic to both providers and end users.

One mitigation strategy is VM migration [6]–[8], where VMs are moved from one physical host to another in order to improve performance and reliability. There are different approaches to VM migration. Forsman *et al.* [73] described three different approaches, namely: cold migration, hot migration, and live migration. Cold migration involves shutting down the guest OS before moving the VM to a predetermined host and restarting the system. Hot migration, on the other hand, only suspends the running guest OS rather than shutting it down before it is transmitted and resumed at the predetermined target host. The latter has an advantage over the former as the running applications in the guest OS are not restarted from scratch. Live migration guarantees continuous service of the hosted applications while allowing a VM and its running OS to be moved from one physical host to another [74]. During live migration, a VM and its environment comprising running task, OS, memory, vCPU and sometimes the disk are moved seamlessly between two physical hosts [75]. Other benefits of VM migration include improved load balancing, transparent mobility, pro-active fault tolerance, and green computing [78].

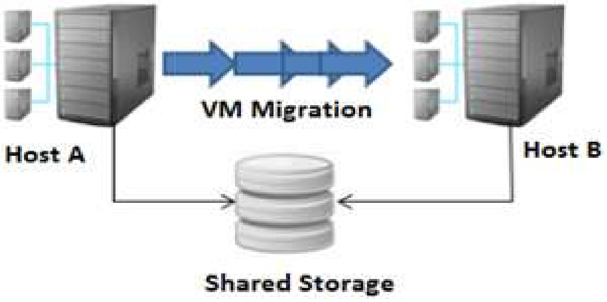

FIGURE 5. VM live migration between two physical machines with shared storage.

VM live migration can, however, be resource intensive as it consumes a large amount of CPU cycles and network bandwidth. Therefore, recent implementations introduced a shared storage (i.e., network attached storage) between the source and the target hosts [76], [77]. With a shared storage, disk storage does not need to be migrated; therefore, only the content of the memory pages that is not available in the shared storage device are transferred (see Fig.5). This immensely reduces the transmission time and downtime time of applications running on the moving VM. During live migration, downtime is the amount of time the migrating VM halts to move from source to target host, while total migration time refers to the total time from the commencement of the migration to the time when the VM is up and running on the target host. Downtime, migration time and amount of dirty pages (data) migrated during VM live migration are some of the key performance metrics [79] used by researchers in their optimization attempts to achieve high availability, load balancing and resilience in a virtualized environment.

Bittencourt *et al.* [113] described a VM migration scenario, similar to handoff procedure in a cellular network, by presenting a general architectural component needed to ensure a location-based VM migration in fog computing. They described a layered fog architecture comprising mobile devices, cloudlets and cloud computing systems. The fog API located in the mobile device layer is a set of pre-defined functions that enables data and computation offloading and migration control. Manzalini *et al.* [114] described a use case for VM migration at the edge of a network by developing a testbed that supports network function migration.

1) VM LIVE MIGRATION TECHNIQUES

During VM live migration, a chunk of the memory state is migrated to a target host even as the source continues to execute. Pre-paging is a method of optimizing memory-constrained disk-based paging systems. It is commonly regarded as a proactive way of pre-fetching from disk, where the memory subsystem attempts to hide the latency of highly-locality page faults by logically sequencing the pre-fetched pages [80]. Due to increasing dynamic random-access memory (DRAM) capacities, recent virtual memory does not often employ pre-paging.

Two designs for live migration were presented in [81], namely: pure stop-and-copy and pure on-demand. The former halts the migrating VM and copies the entire memory to the target host to minimize the total migration time. However, this results in an increase in downtime. The latter, on the other hand, functions by restricting the VM to copy only essential data in the kernel to the target host. The remaining VM address space is transferred when accessed at the target host. Both techniques, however, suffer from poor performance. The pure stop-and-copy technique causes significant service disruption while the pure on-demand technique incurs a longer migration time; hence, necessitating the development of a pre-copy approach to achieve a balance between downtime and migration time [7], [79]. In pre-copy live migration, all memory pages are copied in the first iteration while subsequent iterations transfer the modified pages that occurred during the previous iteration. The iterative process functions by periodically tracking dirty pages that occur in previous iterations, in order to keep migration time and downtime to a minimum.

Post-copy live migration has also been proposed in [82], which stops the VM at the initial stage in order to transfer the vCPU state and device to the target host. The VM is started immediately thereafter and subsequent memory pages are fetched from the source on demand. Hines and Gopalan [80] proposed an adaptive pre-paging to eliminate duplicate page transmission and dynamic self-ballooning to avoid the transfer of free memory pages.

Pre-copy algorithm is the predominant approach used for live migrating VMs, as evident in Xen, VMware and KVM hypervisors [73]. As discussed, the memory pages of the running VM are copied iteratively over several rounds until the modified pages are small enough to temporarily halt the VM at the source and resume on the target host. In the first round, all pages are copied while in subsequent rounds, only modified (i.e., dirty) pages are moved. These modified pages can be tracked using a dirty bitmap maintained by the hypervisor.

Several methods have been proposed in the literature to reduce the amount of data transferred between physical hosts during iterative pre-copy stage, which in turn reduces the total migration time and downtime. Michael and Shen [84] proposed an efficient technique to gradually migrate database connection from source to a target host using a self-adapting algorithm designed to minimize performance impact on the migrating tenant. Only frequently accessed cache contents are sent from the source to the target server. Piao *et al.* [85] proposed a snapshot memory compaction technique based on disk cache and memory. It uses an adaptive downtime control scheme based on the history of VM memory update information (i.e., writable working set) in KVM hypervisor. A live and incremental whole-system approach, three-phase migration, was proposed in [86] to minimize downtime resulting from the migration of a large amount of disk storage data. An incremental migration algorithm is, thereafter used to transfer the VM back to its source in a very short migration time. A compression technique, MECOM, was proposed in [87] that uses memory compression based VM migration approach to ensure fast and stable VM migration. Ruan *et al.* [79] proposed an improved pre-filter copy algorithm to reduce the migration time and bandwidth resource consumption while keeping the downtime constant.

Cerroni and Callegati [88] described the live migration of a virtual network function of an emerging paradigm, cloud-based edge network, and proposed a model that can collectively migrate a group of correlated VMs in a single entity. Clark *et al.* [83] presented six stages of pre-copy migration process between two hosts:

a) Pre-migration – A target host with guaranteed resources is pre-selected for future migration by the source host running the VM.
b) Reservation – Resources on the target host are reserved in anticipation for the incoming migrating VM
c) Iterative pre-copy – During the first iteration, the entire RAM is sent from the source to target host, and subsequent modified dirtied pages are sent in preceding iterations.
d) Stop-and-copy – In this stage, the VM is halted in order to copy its CPU state as well as any remaining inconsistent pages to the target. At the end of this stage, the source and target host have consistent copies of the VM.
e) Commitment – The target host indicates that it has successfully received a consistent VM copy and the source acknowledges the message before discarding the original VM. The target host now becomes the primary host.
f) Activation – The migrated VM is now activated and post-migration codes run to re-attach device drivers on the new machine.

In all six stages, the determinant factor of when to move to the stop-and-copy stage after iterative pre-copy to ensure a minimum migration time and downtime has been the subject of recent research (see [90], [99]). This has a huge impact on the performance of application hosted in the VM. In the case of Xen [81], for example, the stop conditions used for pre-copy algorithms are defined as follows:

a) If less than 50 pages were dirtied during the last pre-copy iteration.
b) If 29 pre-copy iterations have been carried out.

c) If more than 3 times the entire allocated RAM to the VM have been copied from source to the target host during the iterative pre-copy stage.

The first condition ensures a guaranteed minimum downtime as few pages are transferred, while the second and third conditions force the migration process into the stop-and-copy stage irrespective of the amount of modified pages left at the source host. This has a significant impact on the downtime of the application running on the VM.

To further enhance the stop condition after the iterative stage, Zhang *et al.* [89] designed and implemented a VM migration selection method that uses a performance degradation that is sensitive to users. Source codes are analyzed to determine memory size, dirty rate and frequently dirty pages that affect transmission time and downtime. Jo *et al.* [9] used a memory-to-disk mapping in Xen hypervisor to maintain an up-to-date mapping of identical memory pages in the network attached storage. During the iterative pre-copy stage in VM live migration, the memory-to-disk mapping is sent directly to the target host and the contents are fetched directly from the network attached storage. This reduces the total migration time while keeping the downtime to a minimum. Ibrahim *et al.* [90] proposed an algorithm that determines when to switch to the stop-and-copy phase when matched memory pattern does not achieve any significant progress during the iterative phase under different scientific application benchmark.

2) VM LIVE MIGRATION EVALUATION

In order to quantify migration performance during VM live migration, we use downtime and total migration time. Downtime is the overall time a VM is suspended during migration that affects the availability of VM during the migration period [79]. Total migration time, on the other hand, is the total time required to move the VM between a source and the target host. Live migration of VMs in virtualized environments, such as fog computing, is critical to the performance and reliability of the running application. Wu and Zhao [75] presented a model that can predict the migration time, given the application behavior of the migrating VM and the resources available for migration in Xen environment. Nathan *et al.* [91] analyzed existing prediction models in KVM and Xen migration, and their findings indicated a very high error rate due to the non-consideration of writable working set size, a number of pages eligible for skip and the relationship between the number of skipped pages, pages dirty rate, and page transfer rate. To counter this, a comprehensive predictive model that estimates the performance of KVM and Xen live migration was proposed. A study to determine the effect of VM live migration on the performance of the running application inside Xen VM was carried out in [92]. Findings showed that migration overhead is generally acceptable, but it should not be neglected, especially in cases of stringent availability conditions.

TABLE 3. Summary of privacy and security challenges in fog computing.

Reference	Privacy	Security		
		Confidentiality	Integrity	Availability
Stolfo et al. [66]	✓			
Yi et al. [11]	✓	✓	✓	✓
Vaquero et al. [63]	✓			
Wang et al. [61]		✓		
Stojmenovic et al. [36]	✓	✓		
Stojmenovic et al. [62]		✓		
Roman et al. [64]	✓	✓	✓	✓
Dastjerdi et al. [19]	✓	✓		
Ahmad et al. [52]	✓	✓		
Moosavi et al. [108]	✓	✓	✓	✓
Byers et al. [55]		✓		
Garcia Lopez et al. [54]	✓	✓		
Yi et al. [53]	✓	✓	✓	

VI. DISCUSSION

The adoption of fog computing has the potential to enhance QoE for real-time applications that are transmitted within a pre-defined timeframe. Fog computing's interoperability feature ensures wide support for different applications. Its interactions with cloud computing and IoT also ensure that location of fog devices at the edge is close to the source of the data to speed up processes and response to events. The data can be further (pre)processed and subsequently analyzed in the cloud.

In this paper, we categorized the uses of fog computing deployment into real-time and near real-time applications. Batch applications, on the other hand, are handled by cloud computing. Our review shows that fog computing is an emerging research topic. Specifically, from an initial single-digit publication count in 2012 and 2013, the number of academic publications in fog computing have increased in 2015 and 2016, which is an indication of the increase interest in this topic. Typical of any new consumer technologies, security and privacy concerns are two key concerns in fog computing. For example, DDoS attacks while not new [100], [121] are one hard-to-mitigate attacks in fog and cloud computing.

VII. A CONCEPTUAL SMART PRE-COPY VM LIVE MIGRATION IN XEN USING LINEAR REGRESSION

It is clear from the discussions in the preceding sections that both fog and cloud computing are driven by virtualization technology for most of its functions. Most proposed pre-copy live migration methods are designed to reduce both migration time and downtime without recourse to their different benchmark workload. We believe that setting a static stop and copy condition without referring to the current benchmark workload will result in an inefficient migration time and downtime. Therefore, we propose a dynamic approach that uses regression analysis based on the amount of dirty pages in previous iterations to predict the downtime. The predicted downtime will be compared with a predefined downtime threshold to determine whether to move into the stop and copy stage.

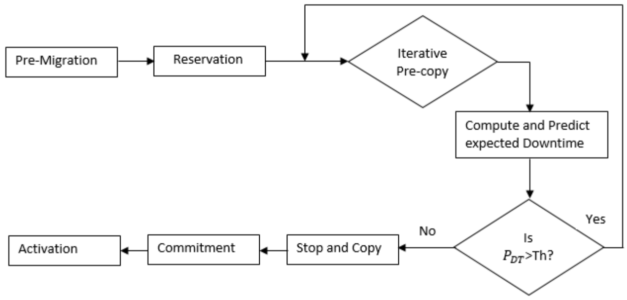

FIGURE 6. **A conceptual live migration framework.**

In this section, we present our conceptual framework, hereafter referred to as the smart pre-copy live migration approach (see Fig.6), which estimates the downtime after each iteration to determine whether to proceed to the stop and copy stage. We now describe the key aspects of this approach.

A. LINEAR REGRESSION APPROACH

Regression, a statistical approach, is useful when prediction is required. Regression estimates the relationship between one or more inputs (which can be an independent variable) to predict a dependent output. When it involves one input, it is referred to as simple regression, while multiple regressions involve two or more inputs. In all cases, the regression relationship can be either linear or nonlinear [101]. In a linear regression, the relationship between variables (i.e., input variable x and output variable y) is a straight line equation.

Several prediction algorithms for VM live migration using linear regression have been proposed in the literature. For example, Farahnakian et al. [101] proposed LiRCUP, a linear regression based CPU usage prediction algorithm, which employs historical information of each host. This is used to approximate the short-time feature of CPU utilization to predict overload or under-load hosts during a live migration process. A modification of multivariate linear regression (MVLR) was proposed in [102], which presented an adaptive algorithm using an ensemble of scaled Fourier analysis, autocorrelation, MVLR, scaling and weighted MVLR to enhance reliability prediction of virtual services. This is achieved by estimating the best prediction value based on the performance of prior predictions at run time to ensure that the SLA is met at a reasonable cost in the cloud environment.

Islam *et al.* [103] proposed an adaptive prediction-based resource measurement and provisioning strategy model by combining neural network and linear regression. This caters for future resource demands by facilitating dynamic and proactive resource management for applications hosted in cloud computing. Beloglazov and Buyya [104] analyzed single VM migration and dynamic VM consolidation problem. An adaptive heuristic based on historical data analysis of resource usage was then proposed for performance and energy efficient dynamic consolidation of VMs using the regression approach. Rybina *et al.* [105] used simple and multiple linear regression models to estimate the time taken to live migrate a VM at run time by considering important parameters, such as CPU instruction retired, dirty memory pages and last level cache line misses that exhibits a strong correlation with migration time. Strunk [106] proposed a lightweight model that estimates the energy cost of live migrating for an idle VM in KVM with respect to the RAM size of the VM and the available network bandwidth using linear regression of recorded data. Huber *et al.* [107] proposed a mathematical model using linear regression to predict the performance of services deployed in a virtualized environment when migrating applications using Citrix XenServer and Vmware.

In our conceptual framework (see Fig. 6), we adapt the pre-copy algorithm in Xen by presenting a smart pre-copy live migration.

B. SMART PRE-COPY LIVE MIGRATION

The pre-copy live migration model in Xen, as discussed in Section V, presents three stop conditions. The first condition (when less than 50 memory pages have been dirtied during the last pre-copy iteration) ensures a low downtime when few pages are dirtied. The rate of the dirty pages increases when a high workload benchmark is involved; therefore, the second stop condition (when 29 pre-copy iterations have been carried out) and the third stop condition (when more than three times of the allocated RAM of the VM has been copied from the source to the target host) are applicable. The second and third conditions force the migrating VM into the stop and copy stage, which eventually has an impact on the downtime of the migrating VM.

Our proposed smart pre-copy live migration for VMs in a virtualized environment that ensures high availability, adds intelligence to the iterative pre-copy stage by estimating the downtime after each iteration using linear regression to determine whether to proceed to the stop and copy stage. This is a function of the benchmark workload involved that can be interpreted as the dirty page rate and the available bandwidth between the source and target physical host. The ''compute and predict expected downtime'' block (see Fig. 6) compares the predicted downtime P_{DT} with a predefined downtime threshold, Then, after each iterative pre-copy round, this will be used to decide whether to proceed to the stop and copy stage or continue with the iteration process until a predefined downtime threshold is met, to achieve a minimum downtime.

Assuming the bandwidth between the source and target host is constant, we use linear regression to estimate the relationship between the independent variable (dirty pages) to predict a dependent variable (downtime).

$$y = mx + b, \tag{1}$$

where y is the dependent variable and x represents the independent variable, and m and b are the regression coefficients.

A measure of fit, that is how well the output variable y is predicted, is measured as the degree of error

$$\varepsilon_i = y_i - \hat{y}_i, \tag{2}$$

where ε_i is the difference between the predicted output $\hat{y}_i$ and the real output y_i in data point i.

To determine the least square criterion, we use:

$$min \sum (y_i - \hat{y}_i)^2, \tag{3}$$

where y_i is the observed value of the dependent variable (downtime) and $\hat{y}_i$ is the estimated (predicted) value of the dependent variable.

From Eqn. (1), we can determine our slope m and intercept b using Eqn. (4) below

$$m = \frac{\sum (x_i - \bar{x})(y_i - \bar{y}_i)}{\sum (x_i - \bar{x})^2} \tag{4}$$

$$\text{and } b = \bar{y} - m\bar{x}, \tag{5}$$

where $\bar{x}$ and $\bar{y}$ are the mean of the dependent and independent variable, and x_i and y_i are values of the dependent and independent variable, respectively.

To predict the number of dirty pages for future iterations using the linear regression approach, we use historical data from previous iterations. We can use Eqn. (1) to determine the function that shows the linear relationship that exists between the workload and downtime. The estimated downtime after each iteration will be compared with the downtime threshold set to decide whether to proceed to the stop and copy stage. This ensures a minimum downtime is achieved during VM live migration between source and target host.

VIII. CONCLUSION AND FUTURE RESEARCH

In this paper, we reviewed academic literature on the paradigm shift from cloud to fog computing published between January 2012 and December 2016. We then presented a taxonomy of different fog computing applications by grouping them into real-time and near real-time. The low-latency requirement of these applications necessitates the extension of the cloud to the edge of the network; thus, resulting in fog computing. Both cloud and fog computing are highly virtualized platforms that provide resources, such as computation, networking and storage. The requirements of high availability by end users motivates the design of the smart pre-copy live migration in Xen presented in this paper. The proposed approach estimates the downtime during the iterative pre-copy stage to determine whether to proceed to the stop and copy stage. This will guarantee a minimum downtime. Future work will include deploying the framework in a real-world or test environment, with the aims of validating and refining the framework.

Fog computing, being in its infancy stage, has a number of challenges due to its architectural design. For example, it is susceptible to trust and authentication issues due to its distributed feature. Cyber attacks such as DDoS attacks can also be detrimental to fog computing's availability as the capacity of each fog node is limited. Therefore, there is a need for more research in the areas of authentication, access control and intrusion detection in fog computing.

Extending cloud to the edge of the network will involve deploying fog nodes close to the end users. This significantly increase the number of devices deployed which results in an increase in energy consumption. Therefore, effort should be expanded into promoting green computing to help reduce global warming.

ACKNOWLEDGEMENTS

The authors thank the three anonymous reviewers for providing constructive and generous feedback.

REFERENCES

[1] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework," *J. Netw. Comput. Appl.*, vol. 67, pp. 147–165, May 2016.

[2] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing," *J. Netw. Comput. Appl.*, vol. 67, 99–117, May 2016.

[3] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Generat. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[4] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. 10th Int. Conf. Wireless Algorithms, Syst., Appl. (WASA)*, Qufu, China, 2015, pp. 685–695.

[5] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for Cloud of Things," in *Proc. IEEE Int. Conf. Future Internet Things Cloud (FiCloud)*, Barcelona, Spain, Aug. 2014, pp. 464–470.

[6] V. Medina and J. M. García, "A survey of migration mechanisms of virtual machines," *ACM Comput. Surv.*, vol. 46, no. 3, 2014, Art. no. 30.

[7] A. Shribman and B. Hudzia, "Pre-copy and post-copy VM live migration for memory intensive applications," in *Euro-Par 2012: Parallel Processing Workshops* (Lecture Notes in Computer Science). New York, NY, USA: Springer, 2012, pp. 539–547.

[8] U. Deshpande, Y. You, D. Chan, N. Bila, and K. Gopalan, "Fast server deprovisioning through scatter-gather live migration of virtual machines," in *Proc. 7th IEEE Int. Conf. Cloud Comput. (CLOUD)*, Anchorage, AK, USA, Jun./Jul. 2014, pp. 376–383.

[9] C. Jo, E. Gustafsson, J. Son, and B. Egger, "Efficient live migration of virtual machines using shared storage," in *Proc. 9th ACM SIGPLAN/SIGOPS Int. Conf. Virtual Execution Environ.*, Houston, TX, USA, 2013, pp. 41–50.

[10] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Commun. Mag.*, vol. 50, no. 9, pp. 34–40, Sep. 2012.

[11] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. ACM Workshop Mobile Big Data*, Hangzhou, China, 2015, pp. 37–42.

[12] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments* (Studies in Computational Intelligence). New York, NY, USA: Springer, 2014, pp. 169–186.

[13] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Gwangju, South Korea, Mar. 2015, pp. 687–694.

[14] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving Web sites performance using edge servers in fog computing architecture," in *Proc. IEEE 7th Int. Symp. Service Oriented Syst. Eng. (SOSE)*, Redwood City, CA, USA, 2013, pp. 320–323.

[15] *Fog Computing and Internet of Things: Extend the Cloud to Where the Things Are*, accessed on Apr. 18, 2017. [Online]. Available: http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

[16] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun. (2015). "Fog computing: Focusing on mobile users at the edge." [Online]. Available: https://arxiv.org/abs/1502.01815

[17] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. ACM 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[18] C. Dsouza, G.-J. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: Preliminary framework and a case study," in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integr. (IRI)*, Redwood City, CA, USA, Aug. 2014, pp. 16–23.

[19] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya. (2016). "Fog computing: Principles, architectures, and applications." [Online]. Available: https://arxiv.org/abs/1601.02752

[20] K. P. Saharan and A. Kumar, "Fog in comparison to cloud: A survey," *Int. J. Comput. Appl.*, vol. 122, no. 3, pp. 10–12, 2015.

[21] J. Su, F. Lin, X. Zhou, and X. Lu, "Steiner tree based optimal resource caching scheme in fog computing," *China Commun.*, vol. 12, no. 3, pp. 161–168, 2015.

[22] X. Zhu, D. S. Chan, H. Hu, M. S. Prabhu, E. Ganesan, and F. Bonomi, "Improving video performance with edge servers in the fog computing architecture," *Intel Technol. J.*, vol. 19, no. 3, pp. 202–224, 2015.

[23] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Ottawa, ON, Canada, May 2015, pp. 1202–1207.

[24] V. K. Sehgal, A. Patrick, A. Soni, and L. Rajput, "Smart human security framework using Internet of Things, cloud and fog computing," in *Intelligent Distributed Computing* (Advances in Intelligent Systems and Computing), New York, NY, USA: Springer, vol. 321. 2015, pp. 251–263.

[25] M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog computing, cloud computing, and more Fog computing," in *Proc. 19th IEEE Int. Workshop Comput.-Aided Modeling Design Commun. Links Netw. (CAMAD)*, Athens, Greece, Dec. 2014, pp. 325–329.

[26] G. Suciu *et al.*, "Big data, Internet of Things and cloud convergence—An architecture for secure E-health applications," *J. Med. Syst.*, vol. 39, no. 11, pp. 1–8, 2015.

[27] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto clouds: Leveraging mobile devices to provide cloud service at the edge," in *Proc. 8th IEEE Int. Conf. Cloud Comput. (CLOUD)*, New York, NY, USA, Jun./Jul. 2015, pp. 9–16.

[28] S. Cirani, G. Ferrari, N. Iotti, and M. Picone, "The IoT hub: A fog node for seamless management of heterogeneous connected smart objects," in *Proc. 12th Annu. IEEE Int. Conf. Sens., Commun., Netw.-Workshops (SECON Workshops)*, Seattle, WA, USA, Jun. 2015, pp. 1–6.

[29] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, to be published.

[30] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," in *Proc. 2nd IEEE Int. Symp. Future Inf. Commun. Technol. Ubiquitous HealthCare (Ubi-HealthTech)*, Beijing, China, May 2015, pp. 1–5.

[31] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, "On security and privacy issues of fog computing supported Internet of Things environment," in *Proc. 6th IEEE Int. Conf. Netw. Future (NOF)*, Montreal, QC, Canada, 2015, pp. 1–3.

[32] V. Gazis, A. Leonardi, K. Mathioudakis, K. Sasloglou, P. Kikiras, and R. Sudhaakar, "Components of fog computing in an industrial Internet of Things context," in *Proc. 12th Annu. IEEE Int. Conf. Sens., Commun., Netw.-Workshops (SECON Workshops)*, Seattle, WA, USA, Jun. 2015, pp. 1–6.

[33] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the Internet of Things," in *Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput.*, Hong Kong, 2013, pp. 15–20.

[34] C. S. Magurawalage, K. Yang, and K. Wang. (2015). "Aqua computing: Coupling computing and communications." [Online]. Available: https://arxiv.org/abs/1510.07250

[35] J. Foerster *et al.*, "Towards realizing video aware wireless networks," *Intel Technol. J.*, vol. 19, no. 3, pp. 6–25, 2015.

[36] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. IEEE Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Warsaw, Poland, Sep. 2014, pp. 1–8.

[37] Z. Zhao, K. Hwang, and J. Villeta, "Game cloud design with virtualized CPU/GPU servers and initial performance results," in *Proc. 3rd ACM Workshop Sci. Cloud Comput.*, Delft, The Netherlands, 2012, pp. 23–30.

[38] Y.-T. Lee, K.-T. Chen, H.-I. Su, and C.-L. Lei, "Are all games equally cloud-gaming-friendly? An electromyographic approach," in *Proc. 11th IEEE Annu. Workshop Netw. Syst. Support Games (NetGames)*, Venice, Italy, Nov. 2012, pp. 1–6.

[39] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "Edge cloud: A new hybrid platform for on-demand gaming," School Comput. Sci., Univ. Waterloo, Waterloo, ON, USA, Tech. Rep. CS-2012–19, 2012.

[40] S. Wang and S. Dey, "Cloud mobile gaming: Modelling and measuring user experience in mobile wireless networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 16, no. 3, pp. 10–21, 2012.

[41] Y. Lin and H. Shen, "Cloud fog: Towards high quality of experience in cloud gaming," in *Proc. 44th IEEE Int. Conf. Parallel Process. (ICPP)*, Beijing, China, Sep. 2015, pp. 500–509.

[42] S. Galli, A. Scaglione, and Z. Wang, "For the grid and through the grid: The role of power line communications in the smart grid," in *Proc. IEEE*, vol. 99, no. 3, pp. 998–1027, Jun. 2011.

[43] S. Abdelwahab, B. Hamdaoui, M. Guizani, and A. Rayes, "Enabling smart cloud services through remote sensing: An Internet of everything enabler," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 276–288, Jun. 2014.

[44] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proc. Telecommun. Netw. Appl. Conf. (ATNAC)*, Southbank, VIC, Australia, 2014, pp. 117–122.

[45] K. Vatanparvar and M. A. Al Faruque, "Demo abstract: Energy management as a service over fog computing platform," in *Proc. ACM/IEEE Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Seattle, WA, USA, Apr. 2015, pp. 1–2.

[46] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog Computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secure Comput. Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM)*, Liverpool, U.K., Oct. 2015, pp. 356–363.

[47] Y. Cao, P. Hou, D. Brown, J. Wang, and S. Chen, "Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing," in *Proc. ACM Workshop Mobile Big Data*, Hangzhou, China, 2015, pp. 43–48.

[48] Y. Cao, S. Chen, P. Hou, and D. Brown, "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Boston, MA, USA, Aug. 2015, pp. 2–11.

[49] T. N. Gia *et al.*, "Fog computing in body sensor networks: An energy efficient approach," in *Proc. IEEE Int. Body Sensor Netw. Conf. (BSN)*, Cambridge, MA, USA, Jan. 2015, pp. 1–7.

[50] M. Aazam and E.-N. Huh, "E-HAMC: Leveraging fog computing for emergency alert service," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, St. Louis, MI, USA, Mar. 2015, pp. 518–523.

[51] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, "Fog data: Enhancing telehealth big data through fog computing," in *Proc. ACM ASE BigData Social Inform.*, Kaohsiung, Taiwan, 2015, p. 14.

[52] M. Ahmad, M. B. Amin, S. Hussain, B. H. Kang, K. Taechoong, and S. Lee, "Health fog: A novel framework for health and wellness applications," *J. Supercomput.*, vol. 72, no. 10, pp. 3677–3695, 2016.

[53] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Washington, DC, USA, Nov. 2015, pp. 73–78.

[54] L. P. Garcia *et al.*, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 3, pp. 37–42, 2015.

[55] C. C. Byers and P. Wetterwald, "Fog computing distributing data and intelligence for resiliency and scale necessary for IoT: The Internet of Things (ubiquity symposium)," *ACM Ubiquity Mag.*, vol. 2015, p. 4, Nov. 2015.

[56] R. Kitchin, "The real-time city? Big data and smart urbanism," *GeoJournal*, vol. 79, no. 1, pp. 1–14, 2014.

[57] M. Gerla, "Vehicular cloud computing," in *Proc. 11th IEEE Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Ayia Napa, Cyprus, Jun. 2012, pp. 152–155.

[58] O. T. T. Kim, N. D. Tri, N. N. Tran, and C. S. Hong, "A shared parking model in vehicular network using fog and cloud environment," in *Proc. 17th IEEE 17th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Busan, South Korea, Aug. 2015, pp. 321–326.

[59] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[60] M. Zhanikeev, "A cloud visitation platform to facilitate cloud federation and fog computing," *Computer*, vol. 48, no. 3, pp. 80–83, 2015.

[61] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proc. 39th IEEE Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Taichung, Taiwan, Jul. 2015, pp. 53–59.

[62] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 10, pp. 1–15, 2015.

[63] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[64] R. Roman, J. Lopez, and M. Mambo. (2016). "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges." [Online]. Available: https://arxiv.org/abs/1602.00484

[65] Y. W. Law, M. Palaniswami, G. Kounga, and A. Lo, "WAKE: Key management scheme for wide-area measurement systems in smart grid," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 34–41, Jan. 2013.

[66] S. J. Stolfo, M. Ben Salem, and A. D. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in *Proc. IEEE Symp. Secur. Privacy Workshops (SPW)*, San Francisco, CA, USA, May 2012, pp. 125–128.

[67] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," in *Proc. 3rd Int. Conf. Frontiers Intell. Comput., Theory Appl. (FICTA)*, Odisha, India, 2014, pp. 1–11.

[68] M. Dong, K. Ota, and A. Liu, "Preserving source-location privacy through redundant fog loop for wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secure Comput. Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM)*, Liverpool, U.K., Oct. 2015, pp. 1835–1842.

[69] S. Kulkarni, S. Saha, and R. Hockenbury, "Preserving privacy in sensor-fog networks," in *Proc. 9th IEEE Int. Conf. Internet Technol. Secured Trans. (ICITST)*, London, U.K., Dec. 2014, pp. 96–99.

[70] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proc. ACM ASE BigData Social Inform.*, Kaohsiung, Taiwan, 2015, p. 28.

[71] R. Lu, X. Liang, X. Li, X. Lin, and X. S. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 1621–1631, Sep. 2012.

[72] C. Wang, F. Mueller, and C. Engelmann, and S. L. Scott, "Proactive process-level live migration and back migration in HPC environments," *J. Parallel Distrib. Comput.*, vol. 72, no. 2, pp. 254–267, 2012.

[73] M. Forsman, A. Glad, L. Lundberg, and D. Ilie, "Algorithms for automated live migration of virtual machines," *J. Syst. Softw.*, vol. 101, pp. 110–126, Mar. 2015.

[74] R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai, and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," *J. Netw. Comput. Appl.*, vol. 52, pp. 11–25, Jun. 2015.

[75] Y. Wu and M. Zhao, "Performance modeling of virtual machine live migration," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Washington, DC, USA, Jul. 2011, pp. 492–499.

[76] C. Jo, E. Gustafsson, J. Son, and B. Egger, "Efficient live migration of virtual machines using shared storage," *ACM SIGPLAN Notices*, vol. 48, no. 3, pp. 41–50, 2013.

[77] S. Das, S. Nishimura, D. Agrawal, and A. El Abbadi, "Albatross: Lightweight elasticity in shared storage databases for the cloud using live data migration," in *Proc. VLDB Endowment*, vol. 4, no. 3, pp. 494–505, 2011.

[78] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in *Proc. 18th ACM Int. Symp. High Perform. Distrib. Comput.*, Munich, Germany, 2009, pp. 101–110.

[79] Y. Ruan, Z. Cao, and Z. Cui, "Pre-filter-copy: Efficient and self-adaptive live migration of virtual machines," *IEEE Syst. J.*, vol. 10, no. 4, pp. 1459–1469, Dec. 2016.

[80] M. R. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in *Proc. ACM SIGPLAN/SIGOPS Int. Conf. Virtual Execution Environ.*, Washington, DC, USA, 2009, pp. 51–60.

[81] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Predicting the performance of virtual machine migration," in *Proc. IEEE Int. Symp. Modeling, Anal. Simulation Comput. Telecommun. Syst. (MASCOTS)*, Miami Beach, FL, USA, Aug. 2010, pp. 37–46.

[82] M. R. Hines, U. Deshpande, and K. Gopalan, "Post-copy live migration of virtual machines," *ACM SIGOPS Oper. Syst. Rev.*, vol. 43, no. 3, pp. 14–26, 2009.

[83] C. Clark *et al.*, "Live migration of virtual machines," in *Proc. 2nd USENIX Conf. Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, 2005, pp. 273–286.

[84] N. Michael and Y. Shen, "Downtime-free live migration in a multi-tenant database," in *Proc. 6th TPC Technol. Conf. Traditional to Big Data (TPCTC)*, Hangzhou, China, 2014, pp. 130–155.

[85] G. Piao, Y. Oh, B. Sung, and C. Park, "Efficient pre-copy live migration with memory compaction and adaptive VM downtime control," in *Proc. 4th IEEE Int. Conf. Big Data Cloud Comput. (BdCloud)*, Sydney, NSW, Australia, Dec. 2014, pp. 85–90.

[86] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and incremental whole-system migration of virtual machines using block-bitmap," in *Proc. IEEE Int. Conf. Cluster Comput.*, Tsukuba, Japan, Sep./Oct. 2008, pp. 99–106.

[87] H. Jin, L. Deng, S. Wu, X. Shi, H. Chen, and X. Pan, "MECOM: Live migration of virtual machines by adaptively compressing memory pages," *Future Generat. Comput. Syst.*, vol. 38, pp. 23–35, Sep. 2014.

[88] W. Cerroni and F. Callegati, "Live migration of virtual network functions in cloud-based edge networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 2963–2968.

[89] W. Zhang *et al.*, "Performance degradation-aware virtual machine live migration in virtualized servers," in *Proc. 13th IEEE Int. Conf. Parallel Distrib. Comput., Appl. Technol. (PDCAT)*, Beijing, China, Dec. 2012, pp. 429–435.

[90] K. Z. Ibrahim, S. Hofmeyr, C. Iancu, and E. Roman, "Optimized pre-copy live migration for memory intensive applications," in *Proc. ACM Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Seattle, WA, USA, 2012, p. 40.

[91] S. Nathan, U. Bellur, and P. Kulkarni, "Towards a comprehensive performance model of virtual machine live migration," in *Proc. 6th ACM Symp. Cloud Comput.*, Kohala Coast, HI, USA, 2015, pp. 288–301.

[92] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *Proc. 1st Int. Conf. Cloud Comput.*, Beijing, China, 2009, pp. 254–265.

[93] F. Machida, D. S. Kim, and K. S. Trivedi, "Modeling and analysis of software rejuvenation in a server virtualized system," *Perform. Eval.*, vol. 70, no. 3, pp. 212–230, 2013.

[94] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric Internet of Things services," in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Madrid, Spain, Jun. 2015, pp. 1–2.

[95] W. E. Sulistiono and S. Liu, "Applying SOFL to constructing a smart traffic light specification," in *Proc. 3rd Int. Workshop Struct. Object-Oriented Formal Lang. Method (SOFL+MSVL)*, Queenstown, New Zealand, 2013, pp. 166–174.

[96] J. Varia, "Best practices in architecting cloud applications in the AWS cloud," in *Cloud Computing: Principles and Paradigms*. 2011, pp. 459–490.

[97] O. Osanaiye and M. Dlodlo, "TCP/IP header classification for detecting spoofed DDoS attack in cloud environment," in *Proc. 16th IEEE Int. Conf. Comput. Tool IEEE (EUROCON)*, Salamanca, Spain, Sep. 2015, pp. 1–6.

[98] F. Longo, R. Ghosh, V. K. Naik, and K. S. Trivedi, "A scalable availability model for infrastructure-as-a-service cloud," in *Proc. 41st IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Hong Kong, Jun. 2011, pp. 335–346.

[99] H. Jin, W. Gao, S. Wu, X. Shi, X. Wu, and F. Zhou, "Optimizing the live migration of virtual machine by CPU scheduling," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1088–1096, 2011.

[100] O. A. Osanaiye, "IP spoofing detection for preventing DDoS attack in Cloud Computing," in *Proc. 18th IEEE Int. Conf. Intell. Next Generat. Netw. (ICIN)*, Paris, France, Feb. 2015, pp. 139–141.

[101] F. Farahnakian, P. Liljeberg, and J. Plosila, "LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers," in *Proc. 39th IEEE EUROMICRO Conf. Softw. Eng. Adv. Appl. (SEAA)*, Santander, Spain, Sep. 2013, pp. 357–364.

[102] I. J. Davis, H. Hemmati, R. C. Holt, M. W. Godfrey, D. M. Neuse, and S. Mankovskii, "Regression-based utilization prediction algorithms: An empirical investigation," in *Proc. CASCON*, 2013, pp. 106–120.

[103] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generat. Comput. Syst.*, vol. 28, no. 1, pp. 155–162, 2012.

[104] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput., Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[105] K. Rybina, W. Dargie, S. Umashankar, and A. Schill, "Modelling the live migration time of virtual machines," in *Proc. OTM Confederated Int. Conf. Move Meaningful Internet Syst.*, Oct. 2015, pp. 575–593.

[106] A. Strunk, "A lightweight model for estimating energy cost of live migration of virtual machines," in *Proc. 6th IEEE Int. Conf. Cloud Comput.*, Santa Clara, CA, USA, Jun./Jul. 2013, pp. 510–517.

[107] N. Huber, M. von Quast, M. Hauck, and S. Kounev, "Evaluating and modeling virtualization performance overhead for cloud environments," in *Proc. CLOSER*, 2011, pp. 563–573.

[108] S. R. Moosavi *et al.*, "End-to-end security scheme for mobility enabled healthcare Internet of Things," *Future Generat. Comput. Syst.*, vol. 64, pp. 108–124, Nov. 2016.

[109] A. Antonić, M. Marjanović, K. Pripužić, and I. P. Žarko, "A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things," *Future Generat. Comput. Syst.*, vol. 56, pp. 22–607, Mar. 2016.

[110] K.-K. R. Choo and R. Sarre, "Balancing privacy with legitimate surveillance and lawful data access," *IEEE Cloud Comput.*, vol. 2, no. 4, pp. 8–13, Jul./Aug. 2015.

[111] K. K. R. Choo, "Legal issues in the cloud," *IEEE Cloud Comput.*, vol. 1, no. 3, pp. 94–96, May 2014.

[112] K. K. Choo, "Challenges in dealing with politically exposed persons," *Trends Issues Crime Criminal Justice*, vol. 386, pp. 1–6, Feb. 2010.

[113] L. F. Bittencourt, M. M. Lopes, I. Petri, and O. F. Rana, "Towards virtual machine migration in fog computing," in *Proc. 10th IEEE Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (3PGCIC)*, Kraków, Poland, Nov. 2015, pp. 1–8.

[114] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, and A. Campi, "Clouds of virtual machines in edge networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 63–70, Jul. 2013.

[115] A. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Generat. Comput. Syst.*, to be published.

[116] X. Hou, L. Yong, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 3860–3873, Jun. 2016.

[117] N. Song, C. Gong, X. An, and Q. Zhan, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *IEEE Netw. Technol. Appl.*, vol. 13, no. 3, pp. 156–164, Mar. 2016.

[118] Y. Yan and W. Su, "A fog computing solution for advanced metering infrastructure," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Expo. (T&D)*, May 2016, pp. 1–4.

[119] E. K. Lee, M. Gerla, G. Pau, U. Lee, and J. H. Lim, "Internet of Vehicles: From intelligent grid to autonomous cars and vehicular fogs," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 9, pp. 1–14, 2016.

[120] S. Kitanov, E. Monteiro, and T. Janevski, "5G and the fog—Survey of related technologies and research directions," in *Proc. 18th IEEE Medit. Electrotech. Conf. (MELECON)*, Limassol, Cyprus, Apr. 2016, pp. 1–6.

[121] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, p. 130, Dec. 2016.

[122] P. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Change-point cloud DDoS detection using packet inter-arrival time," in *Proc. 8th Comput. Sci. Electron. Eng. Conf. (CEEC)*, Essex, U.K., 2016, pp. 204–209.

OPEYEMI OSANAIYE (M'14) received the bachelor's degree in electrical engineering from the University of Ilorin, Nigeria, in 2007, the master's degree in telecommunications engineering from the University of Sunderland, U.K., in 2011, and the Ph.D. degree in electrical engineering from the University of Cape Town, South Africa, in 2016. He was a Lecturer with the Telecommunications Department, Federal University of Technology, Minna, Nigeria, and also a part time Lecturer with the Cape Peninsular University of Technology, Cape Town, South Africa. He was with the University of South Australia on a research exchange visit from 2015 to 2016. His research interests include, computer networks, cloud computing, wireless sensor network, fog computing, network security, voice over internet protocol technology, and cloud computing security. He is a registered COREN member.

SHUO CHEN received the M.E. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interests include network security, applied cryptography, and trust management in VANET.

ZHENG YAN (SM'14) received the B.Eng. degree in electrical engineering and the M.Eng. degree in computer science and engineering from Xi'an Jiaotong University, Xi'an, China, in 1994 and 1997, respectively, the M.Eng. degree in information security from the National University of Singapore, Singapore, in 2000, and the Licentiate of Science and Doctor of Science in Technology degrees in electrical engineering from the Helsinki University of Technology, Helsinki, Finland, in 2005 and 2007, respectively. She is currently a Professor with Xidian University, Xi'an, China, and also a Visiting Professor with Aalto University, Espoo, Finland. She authored over 150 publications and solely authored two books. She is the Inventor of over 50 patents and patent applications. Her research interests are in trust, security, and privacy, social networking, cloud computing, networking systems, and data mining. She serves as an Organization and Program Committee Member for numerous international conferences and workshops. She is also an Associate Editor or a Guest Editor of many reputable journals, such as *Information Sciences*, *ACM TOMM*, *Information Fusion*, the IEEE Systems Journal, the IEEE Access, the IEEE IoT Journal, JNCA, MONET, SCN, and FGCS.

RONGXING LU was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2013 to 2016. He has been an Assistant Professor with the Faculty of Computer Science, University of New Brunswick, Canada, since 2016. His research interests include applied cryptography, privacy enhancing technologies, and IoT-big data security and privacy. He received the 2016-17 Excellence in Teaching Award, FCS, UNB. He currently serves as the Secretary of the IEEE ComSoc CIS-TC.

KIM-KWANG RAYMOND CHOO (SM'15) holds the Cloud Technology Endowed Professorship with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio. He is also an Adjunct Associate Professor with the University of South, and a fellow of the Australian Computer Society. He is a recipient of various awards, including the ESORICS 2015 Best Paper Award, the Winning Team of the Germany's University of Erlangen-Nuremberg Digital Forensics Research Challenge 2015, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. He serves on the Editorial Board of *Cluster Computing*, *Digital Investigation*, the IEEE Cloud Computing, *Future Generation Computer Systems*, the *Journal of Network and Computer Applications*, and *PLoS ONE*. He also serves as the Special Issue Guest Editor of the *ACM Transactions on Embedded Computing Systems* (2017; DOI: 10.1145/3015662), *ACM Transactions on Internet Technology* (2016; DOI: 10.1145/3013520), *Digital Investigation* (2016; DOI: 10.1016/j.diin.2016.08.003), *Future Generation Computer Systems* (2016; DOI: 10.1016/j.future.2016.04.017), IEEE Cloud Computing (2015; DOI: 10.1109/MCC.2015.84), IEEE Network (2016; DOI: 10.1109/MNET.2016.7764272), *Journal of Computer and System Sciences* (2017; DOI: 10.1016/j.jcss.2016.09.001), *Multimedia Tools and Applications* (2017; DOI: 10.1007/s11042-016-4081-z), and *Pervasive and Mobile Computing* (2016; DOI: 10.1016/j.pmcj.2016.10.003).

MQHELE DLODLO received the Ph.D. degree from the Delft University of Technology in 1996, the master's degree in electrical engineering from Kansas State University in 1989, and the bachelor's degrees in electrical engineering and mathematics and independent studies in engineering management from Geneva College, Beaver Falls, PA, USA, in 1980 and 1983, respectively. He has designed curricula, taught, consulted, researched, and headed departments and faculties at NUST and the Bulawayo Polytechnic over 21 years. In 2003, he became a Fulbright Senior African Scholar with Virginia Tech, USA, where he was involved in studying Learning and Information Technology Initiatives for academic leadership in research, learning, and fundraising. He has been an Associate Professor in telecommunications with the Electrical Engineering Department, University of Cape Town, Cape Town, South Africa, since 2005. His research interests are in wireless communication systems and applications.