From cloud to fog computing: A review and a conceptual live VM migration framework

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

***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, which estimates 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. Lastly, 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

O. Osanaiye was with Information Assurance Research Group, University of South Australia, South Australia 5095, Australia and the Department of Electrical Engineering, University of Cape Town, South Africa.

M. Dlodlo is with the Department of Electrical Engineering, University of Cape Town, South Africa.

S. Chen is with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, Singapore.

Z. Yan is with the School of Cyber Engineering, Xidian University, Xi'an 710071, China; Department of Communications and Networking, Aalto University, Espoo 02150, Finland.

R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada.

KKR. Choo was with the Information Assurance Research Group, University of South Australia, SA 5095, Australia. He is now with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249-0631, USA. (e-mail: Raymond.choo@fulbrightmail.org)

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 novel smart pre-copy VM live migration conceptual framework to cater for malicious attacks or failure of physical servers which result in unavailability of services and resources. In this paper, 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 tier 1 is the bottom tier comprising of 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 of 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 centre 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.

Fog abstraction layer provides a generic 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.

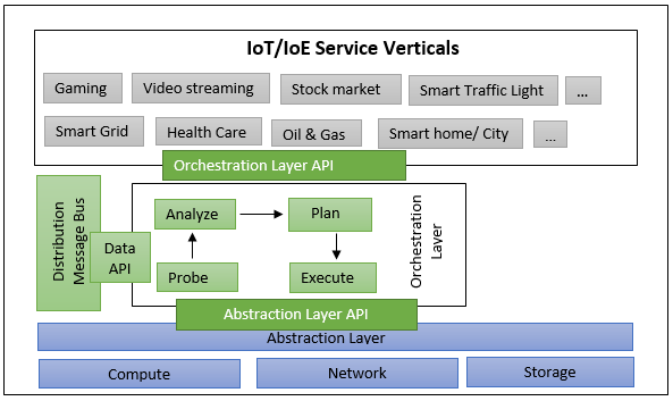

Fig. 1. Architecture and components of fog computing (adapted from [12])

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 comprises of a small software agent- herewith 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 concept of fog computing was borne from 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 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: Most 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.

For the deployment of fog computing, 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 demonstrated that the former achieved 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 called FSDN 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 centralize cloud approach is no longer sufficient under such big data explosion. Yan and Su [118], therefore propose fog computing that enhance the 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's nomenclature was borne from the fact that fog is a cloud close to the ground, intending to bring cloud computing closer to Internet of Things (IoT) devices [2]. The advent of IoT has resulted in a number of use cases that generates a 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 and fog computing for smart human security. This framework provides a wearable computing 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 a novel telemonitoring architecture. This approach uses CloudView Exalead as a search platform that offers access to information present in the infrastructural level for search based application in both online and 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 in both the link layer and application layer to enable resource discovery and seamless interactions among applications.

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

TABLE I. SUMMARY OF FEATURES OF FOG COMPUTING, CLOUD COMPUTING AND IOT

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. Luan et al. [16] described fog as a surrogate of cloud that can be used to deliver location-based service application to mobile 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] in their work 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. Furthermore, 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.

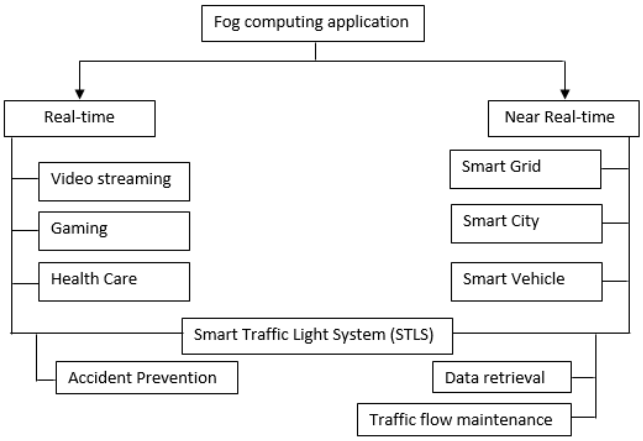

Fig. 2. Proposed fog computing application taxonomy

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

A. Real-time use cases

1) Video streaming

Transmission of video applications and services have harnessed some of the key benefits of fog computing which include location awareness, low latency, support for mobility and real-time analytics. Several smart devices support smart surveillance application that can be used by law enforcement officers to display video streams of suspicious events in real-time. Hong et al. [33] described a video surveillance application that requires three level hierarchy systems 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 serves end users in a video streaming scenario to act as a buffer. Zhu et al. [22] showcased the use of fog computing to transform video applications and services for supporting on-demand video delivery to enhance interactions in virtual desktop infrastructure system and 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 complemented 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 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 is 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 to ensure high QoE gaming. 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. Thereafter, 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 and Wong [39] proposed a new hybrid platform called EdgeCloud (i.e., fog computing) by extending the existing cloud infrastructure. This was achieved by 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 called Cloudfog. Cloudfog 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 Cloudfog 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 advanced 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 at the edge of the network 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 care 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. For emergency situations, Aazam and Huh [50] proposed a smartphone-based service, Emergency Help Alert Mobile Cloud (E-HAMC) that uses fog services for pre-processing and offloading purposes to provide an instant way of notifying relevant emergency department (i.e., ambulance) from the stored contact details. This service also sends the incidence location for easy trace.

Debey 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 and Amin [52] proposed a framework of health fog by deploying fog computing as the intermediary layer between cloud and end users. 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)

The smart traffic light was envisioned by smart connected vehicle and advanced transport system [12]. It interacts locally with a number of sensor nodes to detect the presence of cyclists, bikers or pedestrians and also estimates 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 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 such as 3G, Wi-Fi and smart traffic light units are deployed along the roadside to ensure communication such as vehicle-to-vehicle, vehicle to access point, access point to access point. (see Fig. 3).

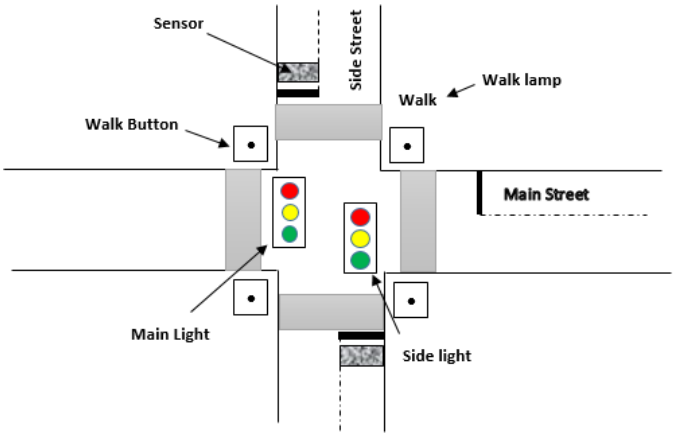

Fig. 3. Smart Traffic Light System (adapted from [95])

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 traditional 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 et al. [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.

2) Smart Cities

A smart city is one key IoT application that ranges from smart traffic management to energy management of buildings. 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 cities that are vastly controlled and made up of ubiquitous computing whose economy and governance is being propelled by innovation and creativity enacted by smart people. The drive for the deployment of smart devices and sensors to homes and cities was initially powered by the repaid development of IoT applications. However, some of these applications and devices present high computation and storage capacity requirement and interoperability issues. Byers and Patrick [55] identified the complexity associated with 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] proposed 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 guarantees 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 Internet cloud is its reliance on the sensor they carry, rather than cloud computing resources. Hou et al. [116] described a vehicular fog computing that utilize 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 offer their complementary features. This caters for future VANET demands and supports 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 tailored towards data privacy, availability entails the ability to remotely access resources offered by cloud servers and fog nodes when needed.

Apart from the inherent 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 novel 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] identified a sophisticated attack, chosen ciphertext attacks (CCA) on fog computing and proposed a solution by first presenting the CCA security model of OD-ABE (attributed-based encryption with outsourced decryption) before presenting the first 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 are deployed in fog computing which range from wireless to Internet-connected mobile devices. Therefore, the attacker can target any of these components. Denial of Service (DoS), man-in-the-middle attack 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 datacentre. For core service infrastructure, privacy leakage, service 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 II summarizes the threat model distribution in fog computing component as identified in [64].

To mitigate against 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] have been suggested. A decoy information technology technique was proposed by Salvatore et al. [66] to withstand malicious insiders by disguising information to prevent attackers from identifying customer's real sensitive data.

TABLE 1I. 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					✓

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, scholars 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's ability to determine the real source node. To mitigate against 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 by providing a resilient system. Yi et al. [11] discussed the reliability improvement of fog computing by periodically carrying out check-pointing to resume after failure and also 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 limited 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.

Figure 3 provides a snapshot of the availability distribution in fog and cloud computing according to their capacity.

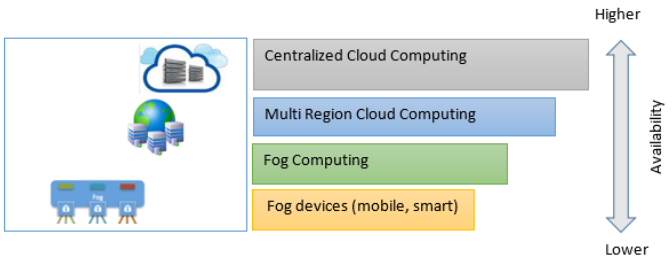

Fig. 4 Availability distribution in fog and cloud computing paradigm

A. *Factors affecting availability in fog computing*

When accessing availability of fog services, security, application failure, and infrastructural failure are the three main factors to be considered. Security issues such as malicious 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, mean time to repair (MMTR) measurement 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 by dividing the available service time by the total time.

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

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 attack or 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 notable benefits of VM migration include improved load

balancing, transparent mobility, pro-active fault tolerance and green computing [78].

VM live migration can, however, be resource intensive as it consumes a large amount of CPU cycles and network bandwidths. 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.

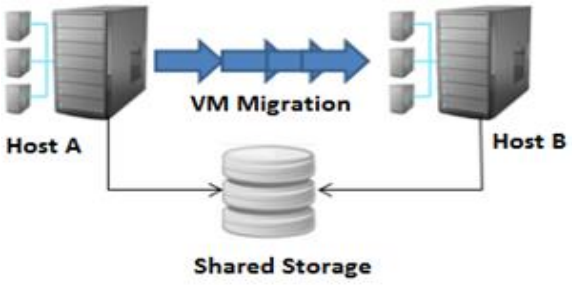

Fig. 5 VM live migration between two physical machines with shared storage

In fog computing, 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 application programming interface (API) located in the mobile device layer are set of pre-defined functions that enables data and computation offloading and migration control. Manzalini et al [114] described a use case of VM migration at the edge of a network by developing a testbed that supports network function migration.

1) State-of-art in VM live migration techniques

During VM live migration, a chunk of the memory state is migrated to a target host even as the source still 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 DRAM capacities, recent virtual memory does not often employ pre-paging.

Akoush et al. [81] identified two earlier proposed designs for live migration, 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 stopping 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 at al. [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. It analyzes source codes 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 which 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 cloud and 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.

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 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.

TABLE III. 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]	✓	✓	✓	

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.

In this section, we present our conceptual framework called smart pre-copy live migration approach (see Fig.6) that 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.

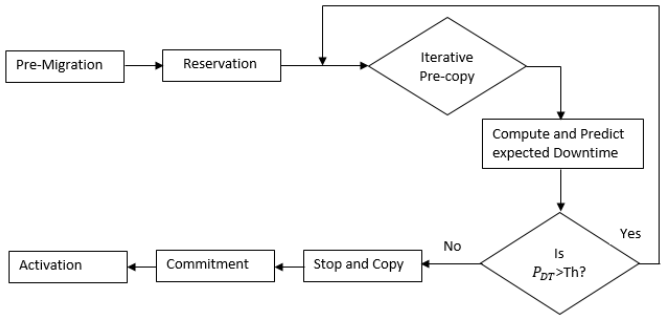

Fig.6. A conceptual live migration framework

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. 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 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 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 energy cost of live migrating 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. The proposed model was able to predict the energy cost of migrating an idle VM with 90% accuracy. Huber et al. [107] proposed a mathematical model using linear regression to predict the performance of services deployed in 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. First is when less than 50 memory pages have been dirtied during the last pre-copy iteration. This condition ensures a low downtime when few pages are dirtied.

The rate of the dirty pages' increase when a high workload benchmark is involved; therefore, the 2nd stop condition (when 29 pre-copy iterations have been carried out) and the 3rd 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 2nd and 3rd 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 (i.e., fog and cloud computing), 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, Th, 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).

$$y = mx + b \quad , \qquad (1)$$

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

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

$$\mathcal{E}_i = y_i - \hat{y}_i \quad , \qquad (2)$$

where $\mathcal{E}_i$ 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:

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

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

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

and $b = \bar{y} - m\bar{x}$, (5)

where $\bar{x}$ and $\bar{y}$ are the mean of the dependent and independent variable, while 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 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 determine whether to proceed to the stop and copy stage. This ensures that 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 has necessitated 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 have necessitated our proposed smart pre-copy live migration in Xen that 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 increases 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.

REFERENCES

[1] Osanaiye. O**.,** Choo K-KR., and Dlodlo. M. "Distributed Denial of Service (DDoS) Resilience in Cloud: Review and Conceptual Cloud DDoS Mitigation Framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147-165, 2016.

[2] Díaz M, Martín C, Rubio B. State-of-the-art, challenges, and open issues in the integration of Internet of Things and Cloud Computing. *Journal of Network and Computer Applications*, 2016 [In Press]

[3] Botta A, de Donato W. Persico V. Pescapé A. Integration of cloud computing and Internet of things: a survey. *Futur Gener Comp Syst* 2016; 56:684-700.

[4] Yi S, Qin Z, Li Q. Security and privacy issues of fog computing: A survey. In: Proceedings of 10th International Conference on Wireless Algorithms, Systems, and Applications, (WASA 2015), Qufu, China; 2015. p. 685-695.

[5] Aazam M, Huh E.-N. Fog computing and smart gateway based communication for Cloud of Things. In: Proceedings of IEEE International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain; 2014. p. 464–470.

[6] Medina V, García JM. A survey of migration mechanisms of virtual machines. ACM Comput Surv (CSUR) 2014; 46(3):30: 1-33.

[7] Shribman A, Hudzia B. Pre-Copy and post-copy VM live migration for memory intensive applications. In: Series of Lecture Notes in Computer Science in Parallel Processing Workshop (Euro-Par); 2012. p. 539-547.

[8] Deshpande U, You Y, Chan D, Bila N, Gopalan K. Fast server deprovisioning through scatter-gather live migration of virtual machines. In: Proceedings of 7th IEEE International Conference on Cloud Computing (CLOUD), Anchorage, Alaska; 2014. p. 376-383.

[9] Jo C, Gustafsson E, Son J, Egger B. Efficient live migration of virtual machines using shared storage. In: Proceeding of the 9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environment, Houston, United State; 2013. p. 41-50.

[10] Mishra M, Das A, Kulkarni P, Sahoo A. Dynamic resource management using virtual machine migrations. IEEE Commun Mag 2012; 50(9):34-40.

[11] Yi S, Li C, Li Q. A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 ACM Workshop on Mobile Big Data, Hangzhou, China; 2015. p. 37-42.

[12] Bonomi F, Milito R, Natarajan P, Zhu J. Fog computing: A platform for Internet of Things and analytics. In: The Series Studies in Computational Intelligence, Big Data and Internet of Things: A Roadmap for Smart Environments; 2014. p. 169-186.

[13] Aazam M, Huh EN. Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT. In: Proceedings of IEEE 29th International Conference on Advanced Information Networking and Applications (AINA), Gwangju, South Korea; 2015. p. 687-694.

[14] Zhu J, Chan DS, Prabhu MS, Natarajan P, Hu H, Bonomi F. Improving web sites performance using edge servers in fog computing architecture. In: Proceedings of IEEE 7th International Symposium on Service Oriented System Engineering (SOSE), Redwood City, California; 2013. p. 320-323.

[15] [Online] Fog Computing and Internet of Things: Extend the Cloud to Where the Things Are. (http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf)

[16] Luan TH, Gao L, Li Z, Xiang Y, Sun L. Fog computing: Focusing on mobile users at the edge 2015. arXiv preprint arXiv:1502.01815.

[17] Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the Internet of Things. In: Proceedings of the ACM first edition of the MCC workshop on Mobile cloud computing, 2012. p. 13-16.

[18] Dsouza C, Ahn GJ, Taguinod M. Policy-driven security management for fog computing: Preliminary framework and a case study. In Proceedings of IEEE 15th International Conference on Information Reuse and Integration (IRI), Redwood City, California; 2014. p. 16-23.

[19] Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R. Fog Computing: Principals, Architectures, and Applications 2016; arXiv.

[20] Saharan KP, Kumar A. Fog in Comparison to Cloud: A Survey. Int J Comput Appl T 2015; 122(3): 10-12.

[21] Su J, Lin F, Zhou X, Lu X. Steiner tree based optimal resource caching scheme in fog computing. China Commun 2015; 12(8): 161-168.

[22] Zhu X, Chan DS, Hu H, Prabhu MS, Ganesan E, Bonomi F. Improving Video Performance with Edge Servers in the Fog Computing Architecture. Intel Tech J 2015; 19(1): 202-224.

[23] Truong NB, Lee GM, Ghamri-Doudane Y. Software defined networking-based vehicular Adhoc Network with Fog Computing. In: Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, Canada; 2015. p. 1202-1207.

[24] Sehgal V K, Patrick A, Soni A, Rajput L. Smart Human Security Framework Using Internet of Things, Cloud and Fog Computing. In: Series of Intelligent Distributed Computing 2015; 321: 251-263.

[25] Yannuzzi M, Milito R, Serral-Gracià R, Montero D, Nemirovsky M. Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. In: Proceedings of 19th IEEE International Workshop on Computer-Aided Modelling and Design of Communication Links and Networks (CAMAD), Athens, Greece; 2014. p. 325-329.

[26] Suciu G, Suciu V, Martian A, Craciunescu R, Vulpe, A, Marcu I, Halunga S, Fratu O. Big Data, Internet of Things and Cloud Convergence–An Architecture for Secure E-Health Applications. J Med Syst 2015; 39(11):1-8.

[27] Habak K, Ammar M, Harras K.A, Zegura E. Femto Clouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge. In: Proceedings of 8th IEEE International Conference on Cloud Computing (CLOUD), New York City, USA; 2015. p. 9-16.

[28] Cirani S, Ferrari G, Iotti N, Picone M. The IoT hub: a fog node for seamless management of heterogeneous connected smart objects. In: Proceedings of 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops), Seattle, USA; 2015. p. 1-6.

[29] Sarkar, Subhadeep, Subarna Chatterjee, and Sudip Misra. Assessment of the Suitability of Fog Computing in the Context of Internet of Things. IEEE Trans Cloud Comput 2015 ;(99):1

[30] Shi Y, Ding G, Wang H, Roman HE, Lu S. The fog computing service for healthcare. In: Proceedings of 2nd IEEE International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), Beijing, China; 2015. p. 1-5.

[31] Lee K, Kim D, Ha D, Rajput U, Oh H. On security and privacy issues of fog computing supported Internet of Things environment. In: Proceedings of 6th IEEE International Conference on the Network of the Future (NOF), Montreal, Canada; 2015. p. 1-3.

[32] Gazis V, Leonardi A, Mathioudakis K, Sasloglou K, Kikiras P, Sudhaakar R. Components of fog computing in an industrial Internet of things context. In: Proceedings of 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops), Seattle, USA; 2015. p. 1-6.
[33] Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B. Mobile fog: A programming model for large-scale applications on the Internet of things. In: Proceedings of the 2nd ACM SIGCOMM workshop on Mobile cloud computing, Hong Kong, China; 2013. p. 15-20.
[34] Magurawalage CS, Yang K, Wang K. Aqua Computing: Coupling Computing and Communications 2015; arXiv preprint arXiv: 1510.07250.
[35] Foerster J, Ott D, Oyman O, Liao Y, Somayazulu S, Zhu X, Chan D S, Neisinger C. Towards Realizing Video aware Wireless Networks. Intel Tech J 2015; 19(1): 6-25.
[36] Stojmenovic I, Wen S. The Fog computing paradigm: Scenarios and security issues. In: Proceedings of IEEE Federated Conference on Computer Science and Information Systems (FedCSIS), Warsaw, Poland; 2014. p. 1-8.
[37] Zhao Z, Hwang K, Villeta, J. Game cloud design with virtualized CPU/GPU servers and initial performance results. In: Proceedings of the 3rd ACM workshop on Scientific Cloud Computing, Delft, Netherlands; 2012. p. 23-30.
[38] Lee Y, Chen K, Su H, Lei C. Are all games equally cloud-gaming-friendly? an electromyography approach. In: Proceedings of 11th IEEE Annual Workshop on Network and Systems Support for Games (NetGames), Venice, Italy; 2012. p. 1-6
[39] Choy S, Wong B, Simon G, Rosenberg C. Edge cloud: A new hybrid platform for on-demand gaming 2012; University of Waterloo, Tech. Rep.
[40] Wang S, Dey S. Cloud mobile gaming: modelling and measuring user experience in mobile wireless networks. ACM SIGCOMM Comp Commun Rev 2012; 16(1): 10-21.
[41] Lin Y, Shen H. Cloud Fog: Towards High Quality of Experience in Cloud Gaming. In: Proceedings of 44th IEEE International Conference on Parallel Processing (ICPP), Beijing, China; 2015. p. 500-509.
[42] Galli S, Scaglione A, Wang Z. For the grid and through the grid: The role of power line communications in the smart grid. In: Proceedings of the IEEE 2011; 99(6): 998-1027.
[43] Abdelwahab S, Hamdaoui B, Guizani M, Rayes A. Enabling smart cloud services through remote sensing: An Internet of everything enabler. IEEE Internet of Things Journal, 2014; 1(3): 276-288.
[44] Stojmenovic I. Fog computing: a cloud to the ground support for smart things and machine-to-machine networks. In: Proceedings of Telecommunication Networks and Applications Conference (ATNAC), Southbank, Australia; 2014.p. 117-122.
[45] Vatanparvar K, Al Faruque M A. Demo Abstract: Energy Management as a Service over Fog Computing Platform. In: Proceedings of ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Seattle, USA; 2015.p. 1-2.
[46] Gia T N, Jiang M, Rahmani A M, Westerlund T, Liljeberg P, Tenhunen H. fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction. In: Proceedings of IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, UK; 2015. p. 356-363.
[47] Cao Y, Hou P, Brown D, Wang J, Chen S. Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing. In: Proceedings of the ACM 2015 Workshop on Mobile Big Data, Hangzhou, China; 2015. p. 43-48.
[48] Cao Y, Chen S, Hou P, Brown D. FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In: Proceedings of IEEE International Conference on Networking, Architecture and Storage (NAS), Boston, USA; 2015. p. 2-11.
[49] Gia TN, Jiang M, Rahmani AM, Westerlund T, Mankodiya K, Liljeberg P, Tenhunen H. Fog Computing in Body Sensor Networks: An Energy Efficient Approach. In: Proceedings of IEEE International Body Sensor Networks Conference (BSN'15), Cambridge. USA; 2015. p. 1-7.
[50] Aazam M, Huh EN. E-HAMC: Leveraging Fog computing for emergency alert service. In: Proceedings of IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), St Louis, USA; 2015. p. 518-523.
[51] Dubey H, Yang J, Constant N, Amiri AM, Yang Q, Makodiya K. Fog Data: Enhancing Telehealth Big Data Through Fog Computing. In: Proceedings of the ACM ASE BigData & Social Informatics, Kaohsiung, Taiwan; 2015. p. 14.
[52] Ahmad M, Amin MB, Hussain S, Kang BH, Cheong T, Lee S. Health Fog: a novel framework for health and wellness applications. The Journal of Supercomputing 2016: 1-19.
[53] Yi S, Hao Z, Qin Z, Li Q. Fog Computing: Platform and Applications. In: proceedings of 3rd IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington DC, USA; 2015. p. 73-78.
[54] Garcia Lopez P, Montresor A, Epema D, Datta A, Higashino T, Iamnitchi A, Barcellos M, Felber P, Riviere E. Edge-centric Computing: Vision and Challenges. ACM SIGCOMM Comp Commun Rev 2015; 45(5): 37-42.
[55] Byers C, Wetterwald P. Fog Computing Distributing Data and Intelligence for Resiliency and Scale Necessary for IoT: The Internet of Things (Ubiquity symposium) ACM Ubiquity Magazine 2015:4.
[56] Kitchin R. The real-time city? Big data and smart urbanism. GeoJournal 2014; 79(1):1-14.
[57] Gerla M. Vehicular cloud computing. In: Proceedings of the 11th IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Ayia Napa, Cyprus; 2012. p. 152-155.
[58] Kim OT, Tri ND, Tran NH, Hong CS, A shared parking model in vehicular network using fog and cloud environment. In: Proceedings of 17th IEEE 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, South Korea; 2015. p. 321-326.
[59] Lu N, Cheng N, Zhang N, Shen X, Mark, JW. Connected vehicles: solutions and challenges. IEEE Internet of Things Journal 2014; 1(4): 289-299.
[60] Zhanikeev M. A cloud visitation platform to facilitate cloud federation and fog computing. Computer 2015; 48(5): 80-83.
[61] Wang Y, Uehara T, Sasaki R. Fog Computing: Issues and Challenges in Security and Forensics. In: Proceedings of 39th IEEE Annual Computer Software and Applications Conference (COMPSAC), Taichung, Taiwan; 2015. p. 53-59.
[62] Stojmenovic I, Wen S, Huang X, Luan H. An overview of Fog computing and its security issues. Concurr Comput- Pract E 2015:1-15.
[63] Vaquero LM, Rodero-Merino L Finding your way in the fog: Towards a comprehensive definition of fog computing. ACM SIGCOMM Comp Commun Rev 2014; 44(5): 27-32.
[64] Roman R, Lopez J, Mambo M. Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges. arXiv preprint arXiv:1602.00484 2016.
[65] Law YW, Palaniswami M, Kounga G, Lo A. WAKE: Key management scheme for wide-area measurement systems in smart grid. IEEE Commun Mag 2013; 51(1): 34-41.
[66] Stolfo S, Salem M, Keromytis A. Fog computing: Mitigating insider data theft attacks in the cloud. In: Proceedings of IEEE Symposium on Security and Privacy Workshops (SPW), San Francisco, USA; 2012. p. 125-128.
[67] Shankarwar MU, Pawar AV. Security and Privacy in Cloud Computing: A Survey. In: Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), Odisha, India: 2014. p. 1-11.
[68] Dong M, Ota K, Liu A. Preserving Source-Location Privacy through Redundant Fog Loop for Wireless Sensor Networks. In: Proceedings of IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, UK: 2015. p. 1835-1842.
[69] Kulkarni S, Saha, S, Hockenbury R. Preserving privacy in sensor-fog networks. In: Proceedings of 9th IEEE International Conference for Internet Technology and Secured Transactions (ICITST), London, UK: 2014. p. 96-99.
[70] Tang B, Chen Z, Hefferman, G, Wei T, He H, Yang Q. A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities. In: Proceedings of the ACM ASE BigData & Social Informatics, Kaohsiung, Taiwan: 2015.p.28.
[71] Lu R., Liang X, Li X, Lin X, Shen XS. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. IEEE Trans Parallel Distrib Syst 2012; 23(9): 1621-1631.
[72] Wang C, Mueller F, Engelmann C. Scott SL. Proactive process-level live migration and back migration in HPC environments. J Parallel Distrib Comput 2012; 72(2):254-267.
[73] Forsman M, Glad A, Lundberg L, Ilie D. Algorithms for automated live migration of virtual machines. J of Syst Softw 2015; 101:110-126.
[74] Ahmad RW, Gani A, Hamid SH, Shiraz M, Yousafzai A, Xia F. A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J Netw Comput Appl 2015; 52:11-25.
[75] Wu Y, Zhao M. Performance modeling of virtual machine live migration. In: Proceedings of the IEEE International Conference on Cloud Computing (CLOUD), Washington DC, USA; 2011.p. 492-499.

[76] Jo C, Gustafsson E, Son J, Egger B. Efficient live migration of virtual machines using shared storage. In: ACM Sigplan Notices 2013; 48(7): 41-50.
[77] Das S, Nishimura S, Agrawal D, El Abbadi A. Albatross: lightweight elasticity in shared storage databases for the cloud using live data migration. In: Proceedings of the VLDB Endowment 2011; 4(8): 494-505.
[78] Liu H, Jin H, Liao X, Hu L, Yu C. Live migration of virtual machine based on full system trace and replay. In: Proceedings of the 18th ACM international symposium on High performance distributed computing, Munich, Germany; 2009. p. 101-110.
[79] Ruan Y, Cao Z, Cui Z. Pre-Filter-Copy: Efficient and Self-Adaptive Live Migration of Virtual Machines. IEEE System Journal 2014; (99): 1-11.
[80] Hines MR, Gopalan, K. Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. In: Proceedings of the ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, Washington DC, USA; 2009.p. 51-60.
[81] Akoush S, Sohan R, Rice A, Moore AW, Hopper A. Predicting the performance of virtual machine migration. In: Proceedings of IEEE International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), Miami, USA; 2010. p. 37-46.
[82] Hines MR, Deshpande U, Gopalan K. Post-copy live migration of virtual machines. ACM SIGOPS operating systems review 2009; 43(3): 14-26.
[83] Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, Pratt I, Warfield A. Live migration of virtual machines. In: Proceedings of the 2nd USENIX conference on Symposium on Networked Systems Design & Implementation (NSDI), Boston, USA; 2005.p. 273-286.
[84] Michael N, Shen Y. Downtime-Free Live Migration in a Multitenant Database. In Performance Characterization and Benchmarking. In: The Traditional to Big Data: 6th TPC Technology Conference (TPCTC), Hangzhou, China; 2014. p. 130-155.
[85] Piao G, Oh Y, Sung B, Park C. Efficient Pre-copy Live Migration with Memory Compaction and Adaptive VM Downtime Control. In: Proceedings of 4th IEEE International Conference on Big Data and Cloud Computing (BdCloud), Sydney, Australia; 2014. p. 85-90.
[86] Luo Y, Zhang B, Wang X, Wang Z, Sun Y, Chen H., 2008. Live and incremental whole-system migration of virtual machines using block-bitmap. In: Proceedings of IEEE International Conference on Cluster Computing, Tsukuba, Japan; 2008. p. 99-106.
[87] Jin H, Deng L, Wu S, Shi X, Chen H, Pan X. MECOM: Live migration of virtual machines by adaptively compressing memory pages. Futur Gener Comp Syst 2014; 38: 23-35.
[88] Cerroni W, Callegati F. Live migration of virtual network functions in cloud-based edge networks. In: Proceedings of IEEE International Conference Communications (ICC), Sydney, Australia; 2014.p. 2963-2968.
[89] Zhang W, Zhu M, Gong T, Xiao L, Ruan L, Mei Y, Sun Y, Ji X. Performance degradation-aware virtual machine live migration in virtualized servers. In: Proceedings of 13th IEEE International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Beijing, China; 2012.p. 429-435.
[90] Ibrahim KZ, Hofmeyr S, Iancu C, Roman E. Optimized pre-copy live migration for memory intensive applications. In: Proceedings of 2011 ACM International Conference for High Performance Computing, Networking, Storage and Analysis, Seattle, USA; 2012. p. 40.
[91] Nathan S, Bellur U, Kulkarni P. Towards a comprehensive performance model of virtual machine live migration. In: Proceedings of the Sixth ACM Symposium on Cloud Computing, Hawaii, USA; 2015.p. 288-301.
[92] Voorsluys W, Broberg J, Venugopal S, Buyya R. Cost of virtual machine live migration in clouds: A performance evaluation. In: Proceedings of the 1st International Conference on Cloud Computing, Beijing, China; 2009.p. 254-265.
[93] Machida F, Kim DS, Trivedi KS. Modelling and analysis of software rejuvenation in a server virtualized system with live VM migration. Perform Eval 2013; 70(3): 212-230.
[94] Datta SK, Bonnet C, Haerri J. Fog Computing architecture to enable consumer-centric Internet of Things services. In: Proceedings of IEEE International Symposium on Consumer Electronics (ISCE), Madrid, Spain; 2015. p. 1-2.
[95] Sulistiono WE, Liu S. Applying SOFL to constructing a smart traffic light specification. In: Proceedings of the 3rd International Workshop Structured Object-Oriented Formal Language and Method (SOFL + MSVL), Queenstown, New Zealand; 2013.p. 166-174.
[96] Varia J. Best practices in architecting cloud applications in the AWS cloud. Cloud Computing. Principles and Paradigms 2011:459-490.
[97] Osanaiye O, Dlodlo M. TCP/IP header classification for detecting spoofed DDoS attack in Cloud environment. In: Proceedings of 16th IEEE International Conference on Computer as a Tool (EUROCON 2015), Salamanca, Spain; 2015.p. 1-6.
[98] Longo F, Ghosh R, Naik VK, Trivedi KS. A scalable availability model for infrastructure-as-a-service cloud. In: Proceedings of the 41st IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), Hong Kong, China; 2011. p. 335-346.
[99] Jin H, Gao W, Wu S, Shi X, Wu X, Zhou F. Optimizing the live migration of virtual machine by CPU scheduling. J Netw Comput Appl 2011; 34(4):1088-1096.
[100] Osanaiye O. IP spoofing detection for preventing DDoS attack in Cloud Computing. In: Proceedings of the 18th IEEE International Conference on Intelligence in Next Generation Networks (ICIN), Paris, France; 2015. p. 139-141.
[101] Farahnakian F, Liljeberg P, Plosila, J. LiRCUP: Linear regression-based CPU usage prediction algorithm for live migration of virtual machines in data centers. In: Proceedings of 39th IEEE EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Santander, Spain; 2013. p. 357-364.
[102] Davis IJ, Hemmati H, Holt RC, Godfrey MW, Neuse D, Mankovskii S. Regression-based utilization prediction algorithms: an empirical investigation. CASCON 2013: 106-120.
[103] Islam S, Keung J, Lee K, Liu A. Empirical prediction models for adaptive resource provisioning in the cloud. Futur Gener Comp Syst 2012; 28(1):155-162.
[104] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr Comput-Pract Exp 2012; 24(13): 1397-1420.
[105] Rybina, K., Dargie, W., Umashankar, S. and Schill, A., 2015, October. Modelling the Live Migration Time of Virtual Machines. In On the Move to Meaningful Internet Systems: OTM 2015 Conferences (pp. 575-593).
[106] Strunk A. A lightweight model for estimating energy cost of live migration of virtual machines. In: Proceedings of the 6th IEEE International Conference on Cloud Computing, Santa Clara, USA; 2013. p. 510-517.
[107] Huber N, von Quast M, Hauck M, Kounev S. Evaluating and Modelling Virtualization Performance Overhead for Cloud Environments. In: CLOSER; 2011.p. 563-573.
[108] Moosavi SR, Gia TN, Nigussie E, Rahmani AM, Virtanen S, Tenhunen H, Isoaho J. End-to-end security scheme for mobility enabled healthcare Internet of Things. Futur Gener Comp Syst 2016 [In press]
[109] Antonić A, Marjanović M, Pripužić K, Žarko IP. A mobile crowd sensing ecosystem enabled by CUPUS: cloud-based publish/subscribe middleware for the Internet of Things. Futur Gener Comp Syst. 2016; 56:607-22.
[110] Choo KK, Sarre R. Balancing Privacy with Legitimate Surveillance and Lawful Data Access. IEEE Cloud Comp. 2015; 2(4):8-13.
[111] Choo KK, Legal Issues in the Cloud. IEEE Cloud Comp. 2014; 1(1): 94-96.

[112] Choo KK, Challenges in dealing with politically exposed persons. Trends & Issues in Crime and Criminal Justice. 2010; 386: 1–6.

[113] Bittencourt L, Petri I, Rana, O, Towards virtual machine migration in fog computing. In the proceedings of the 10th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland; 2015, pp. 1-8.

[114] Manzalini A, Minerva R, Callegati F, Cerroni W, Campi A. Clouds of virtual machines in edge networks. IEEE Commun Mag. 2013; 51(7):63-70.

[115] Zuo A, Shao J, Wei G, Xie M, Ji M. CCA-secure ABE with outsourced decryption for fog computing. Futur Gener Comp Syst 2016; 1-9. [IN PRESS].
[116] Hou X, Yong L, Chen M, Wu D, Jin D, Chen S. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures. IEEE Trans on Vehic Tech 2016; 65(6): 3860-3873.
[117] Song N, Gong C, An X, Zhan Q. Fog Computing Dynamic Load Balancing Mechanism Based on Graph Repartitioning. IEEE Network Technology and Application 2016; 156 – 164.
[118] Yan Y, Su W. A fog computing solution for advanced metering infrastructure. In: Proceedings of IEEE/PES Transmission and Distribution Conference and Exposition (T&D),2016, pp. 1-4.
[119] Lee EK, Gerla M, Pau G, Lee U, Lim JH. Internet of Vehicles: From intelligent grid to autonomous cars and vehicular fogs. International Journal of Distributed Sensor Networks. 2016;12(9):1-14.
[120] Kitanov S, Monteiro E, Janevski T. 5G and the Fog—Survey of related technologies and research directions. In: Proceedings of the 18th IEEE

Mediterranean Electrotechnical Conference (MELECON), Limassol, Cyprus; 2016. p. 1-6.
[121] Osanaiye. O, Cai. H., Choo K-KR., Dehghantanha. A., Xu. Z' and Dlodlo. M. "Ensemble-based Multi-Filter Feature Selection Method for DDoS Detection in Cloud Computing," EURASIP Journal for Wireless and Communications Network, no. 1, pp. 1-10, 2016.
[122] Osanaiye. O. Choo. K-KR. and Dlodlo. M. Change-Point Cloud DDoS Detection using Packet Inter-Arrival time. In: Proceedings of the 8th Computer Science & Electronic Engineering Conference (CEEC'16), Essex, United Kingdom; 2016.p. 204-209.