DOI: http://dx.doi.org/10.26483/ijarcs.v15i1.7047

ISSN No. 0976-5697

Volume 15, No. 1, January-February 2024

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Optimizing Edge Computing through Virtualization: A Comprehensive Review

Nisha Saini
Department of Computer Science and Engineering
Deenbandhu Chhotu Ram University of Science and Technology
Murthal (Sonepat), India

Jitender Kumar
Department of Computer Science and Engineering
Deenbandhu Chhotu Ram University of Science and Technology
Murthal (Sonepat), India

Abstract: The adoption of virtualization in edge computing promotes the use of numerous applications and operating systems on a single physical server. The technological environment is increasingly efficient, economical, and hassle-free by virtue of this technology. This study explores the potential of virtualization in edge computing environments by creating virtual versions of physical resources and services. The primary objective of this study is to reduce hardware and operational expenses by consolidating multiple systems into one physical system. Various deployment strategies for virtual machines on edge nodes, including containers, hypervisors, and virtual machines, are discussed. The article also delves into a virtualization-based framework to address challenges in edge computing. Additionally, through historical research on virtualization techniques, this article provides insights into the benefits and limitations of virtualization in edge computing. Moreover, the integration of virtualization and edge computing is examined, highlighting open challenges that need to be addressed for optimal utilization of virtualization in edge computing environments. The findings contribute to a deeper understanding of the possibilities and challenges associated with virtualization in edge computing environments, laying the foundation for further research in this field.

Keywords: Edge Computing; Virtualization; Virtual Machines; Containers; Hypervisors; Virtualized Environments; Optimization

I. INTRODUCTION

Virtualization in an edge computing environment creates virtual versions of physical resources and services, such as networks, servers, storage devices and applications. Virtualization enables organizations to reduce costs and increase efficiency by consolidating multiple systems into one or more physical systems. By doing so, organizations can deploy the same application on various machines while maintaining a single instance of the application instead of replicating it across those machines. This reduces hardware requirements and operational expenses associated with managing separate copies of software licenses for each machine. Moreover, virtualization allows an organization to quickly provision resources without purchasing hard-copy hardware or software licenses.

Virtual Machine (VM) technology is the foundation for many edge computing solutions that enable organizations to access cost savings through increased operational efficiencies in resource utilization while minimizing upfront hardware costs [1]. VMs are self-contained operating system environments that execute on top of an underlying host operating system (OS). VMs provide complete isolation from other guest OSs running on the same host OS by allowing all processes associated with any given host OS to run independently from all other processes running on different hosts. This makes it possible for multiple VMs to operate within a single physical server without any performance degradation or security threats due to resource contention between them. As such, a single VM can be used in place of several dedicated servers, which reduces the upfront capital costs and maintenance-related operational expenses over time since only one copy needs to be maintained for all its potential users instead of several distinct copies [2].

The combination of virtual machine technology with modern network infrastructure provides a platform that enables organizations to rapidly provision instances within seconds via automation tools like infrastructure as code (IaC) frameworks such as cloud formation or terraform that allow development teams greater flexibility in their operations when compared against manual configuration methods traditionally used up until recently. Furthermore, IaC offers improved scalability due to sources being easily created/deleted/resized according to demand [4]. This was not possible under traditional methods since manual changes would have taken too long compared to what now takes seconds through automation tools like IaC frameworks, which are designed specifically for developing complex infrastructures quickly & easily at the scale & granularity level desired by users. Securely connecting these resources across regions has become much more accessible than advancements made networking capabilities alongside less complexity associated with managing these connections due to orchestration tools available today like Kubernetes, which handle most complications behind the scenes with little effort required surface layer users. The basic structure of virtualization is depicted in Figure 1.

Moreover, some cloud service providers offer virtual servers combined with secure virtual private network (VPN) connections that allow remote access from anywhere in the world through an encrypted tunnel connection provided via their service provider's VPN service that will enable employees working away from home or office locations safe access securely connect internal corporate networks data centres conveniently securely over the internet regardless physical distance between two points connection [5].

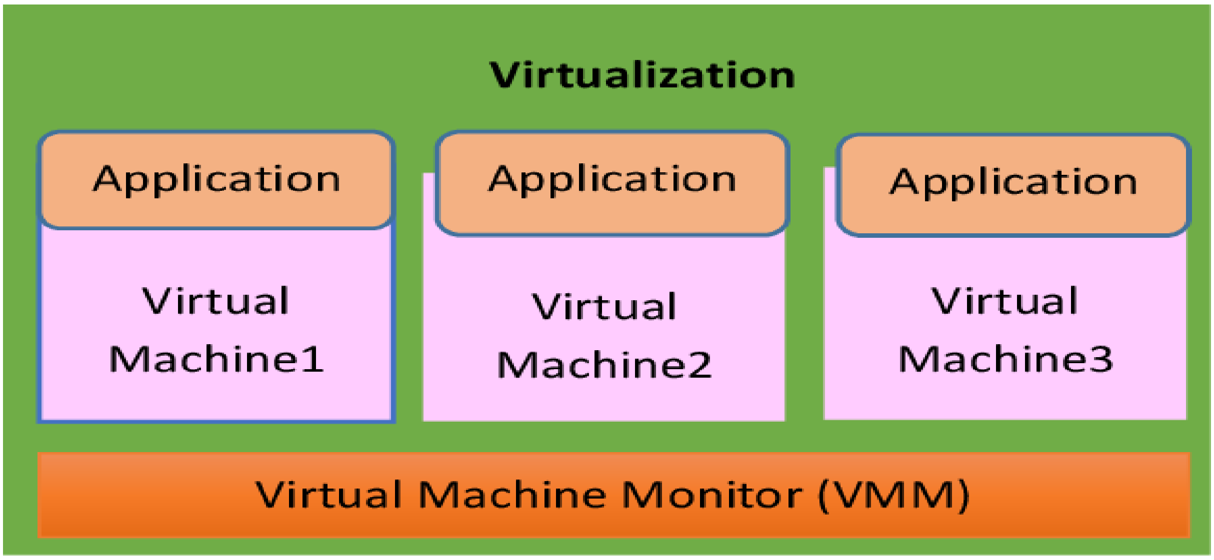

Fig. 1 Virtualization architecture [3]

The significant contribution of this article is as follows:

- This paper focuses on the most important and relevant benefits and challenges of virtualization in edge computing.
- By consolidating multiple operating systems and applications on a single physical server, virtualization can help reduce hardware costs and improve resource utilization in edge computing environments.
- This study evaluates several virtualization frameworks that are utilized in edge computing systems.
- Virtualization supports a variety of deployment approaches on edge nodes, such as containers, hypervisors, and virtual machines. This flexibility allows organizations to choose the best strategy for their needs and requirements.
- This paper also analyzes obstacles in fostering virtualization and edge computing and emphasizes several unresolved issues for optimal utilization in edge computing environments.

This article will delve into several topics, beginning with Section 2, which explores the related work. Section 3 will provide a discussion section. Finally, concluding thoughts and future directions will be explored in Section 4.

II. RELATED LITERATURE

The agility, flexibility, and efficiency of the technology infrastructure are all improved by the widespread adoption of virtualization technologies in cloud computing. It renders it possible to segregate metaphorical computing resources from tangible resources, allowing numerous virtualized environments to be created on a single physical server. Significant perks associated with this technology include reduced expenses for hardware, easier management, and greater scalability of technological assets. However, virtualization poses issues such as quality of services (QoS), infrastructure integration, integrity, cost-cutting, and automation. Experts have studied the potential of virtualization in edge computing environments and its challenges. This review will summarize some relevant research on virtualization in edge computing environments.

One key area of research is ensuring high-quality service in virtualized environments. To achieve this, Al-Asaly et al. [6] proposed a deep learning and machine learning-based approach to monitor the status of cloud computing services in real time. The suggested approach can identify and categorize abnormalities based on severity, allowing immediate remedial measures. Numerous tests on various datasets have demonstrated that the technique is hugely competent in finding and categorizing abnormalities. Researchers have tackled many issues of virtualization in edge computing configurations. One problem arises from integrating network function virtualization (NFV) technologies with existing physical premises. Zhang et al. [7] presented a hybrid cloud framework that accomplishes the issue while enhancing network reliability, lowering delay, and enhancing security. Their assessment results revealed that the proposed methodology surpassed various other approaches.

Another concern that arises in virtualized surroundings is security. Virtualization opens up the possibility of novel avenues for attack that weren't previously possible in conventional physical setups. Researchers have also tackled this issue. For instance, Sun et al. [8] suggested a virtualized security framework that includes a system for detecting intrusions, a privilege monitoring system, and a data encoding technique. Their system comprises a complete security framework that safeguards cloud computing infrastructures against cyber threats. Using a large-scale dataset, the contributors demonstrated that their technique outperformed the current security mechanisms in terms of security. Another intriguing field of investigation for virtualized surroundings is automation, which may mitigate the deployment and upkeep expenditures. Chen et al. [9] presented an automated technique for rapidly deploying virtualized environments that combines cloud computing and container technologies. The infrastructure has an automated administration procedure that instinctively deploys and customizes VMs.

Huang et al. [10] utilized the full virtualization technique to optimize resource utilization. Zhuo et al. [15] emphasized performance improvement using the full virtualization technique. Suranegara et al. [19] tested the system on enormous data sets and discovered that it considerably lowered costs, enhanced flexibility, and optimized performance. Storck et al. [23] used the network function virtualization (NFV) technique to improve QoS and network resource utilization. Several academics [25-28] have investigated the prospective use of virtualization techniques for edge computing systems and proposed suggestions for the issues. These solutions have improved the practice and shown novel perspectives regarding the use of virtualization more effectively. Table 1 summarizes the study outcomes in the historical virtualization setting in edge computing.

Table I. Comparative analysis of virtualization techniques

Year	*References*	*Virtualization Technique*	*Simulation Tool*	*Pros*	*Cons*
2011	Huang et al. [10]	Full Virtualization	CloudSim	Improved resource utilization	High migration overhead
2012	Belbekkouche et al. [11]	Network Virtualization	NS-3	Improved network performance	Limited scalability
2013	Gupta et al. [12]	Full Virtualization	SimGrid	Efficient resource allocation	High overhead
2014	Wang et al. [13]	Hardware-assisted Virtualization	Gem5	Improved performance	Limited hardware support
2015	Song et al. [14]	Para-Virtualization	CloudSim	Improved load balancing	Limited security
2016	Zhou et al. [15]	Full Virtualization	CloudAnalyst	Improved performance	Limited scalability
2017	Du et al. [16]	Para-Virtualization	SimGrid	Efficient resource utilization	Limited hardware support
2017	Pahl et al. [17]	Containerization	OpenShift	Reduced service deployment time and improved resource utilization	Limited scalability for complex services
2018	Alam et al. [18]	Containerization	CloudSim	Reduced resource consumption	Limited isolation
2018	Suranegara et al. [19]	Live Migration	Mininet	Improved VM migration efficiency and network performance	Limited scalability and complexity of implementation
2019	Zeng et al. [20]	Full Virtualization	NS-3	Improved resource allocation	High overhead
2019	Shen et al. [21]	HAV, PV, and Containerization	CloudSim	Improved resource utilization and scalability for different workloads	Limited kernel support for some Virtualization techniques, resource overhead
2020	Lin et al. [22]	VNF	CloudSim	Reduced latency and improved resource utilization	Complex network management and increased overhead
2021	Storck et al. [23]	NFV	NS-3	Improved QoS and network resource utilization	Limited scalability and performance
2022	Kuai et al. [24]	NFV	Customized Simulator	Improved fairness and acceptance ratio	Limited computation capacity.
2023	Attaoui et al. [25]	VNF and CNF	CloudSim	Improved flexibility and reduced resource cost.	Limited work is done in decentralizing the scheduling decisions to multiple edge servers.
2023	Yang et al. [26]	SDN	Visual Studio 2019 C++	User-centric, amended resource sharing and utilization	The proposed methodology was not applicable in real time.
2024	Dogani et al. [27]	Container based virtulaization	Customized Simulator	Enhanced scalability	The proposed technique did not cope with the dynamic heterogeneity of

					clients.
2024	Guo et al. [28]	NFV	Simulink	Reduced latency	SFC assembly induces high latency, network congestion and difficulties in SFC parallelization.

NS-3: Network Simulator, HAV: Hardware-assisted Virtualization, PV: Para-Virtualization, VNF: Virtual Network Function, NFV: Network Function Virtualization, CNF: Container Network Function, SDN: Software-Defined Networking, SFC: Service Function Chains

III. DISCUSSION

In this section, we first provide a comprehensive overview of existing virtualization frameworks applicable to edge computing systems. Subsequently, we examine several virtualization techniques employed within these frameworks. Finally, the discussion culminates in analyzing critical challenges that must be addressed to successfully implement virtualization solutions in the dynamic context of edge computing.

3.1 Virtualization-based frameworks

Edge computing innovations allow organizations to conveniently utilize virtualized resources from nearby or remote servers, allowing them greater oversight and versatility than conventional cloud approaches. It offers organizations a new way to manage their IT resources and applications, using technologies such as virtualization-based frameworks such as the OpenStack platform, KubeEdge, and OpenEdge.

3.1.1 OpenStack

It is a liberated framework that enables sharing computing resources among multiple users in a single instance while delivering cost savings on hardware investments. It allows users to create custom configurations based on either public or private clouds within the same example by utilizing its various components, such as Nova compute nodes for server resource management, Neutron networking nodes for setting up virtual networks, Cinder block storage nodes for managing block storage; Swift object storage nodes for handling objects like files or images; heat orchestration engine for automating application deployment operations; Glance image service node used by VM creators while building their VMs from scratch; keystone identity service node which is used by user authentication processes [29]. As part of its core functionality, OpenStack also provides features like high availability through redundant components across instances and data centres to ensure uninterrupted services even when one component may fail due to natural disasters like floods or earthquakes. In addition, it also offers monitoring capabilities so users can keep track of all their services at once without having real-time access to each one individually.

It is an open-source infrastructure as a service (IaaS) platform [30] that facilitates deploying various applications on multiple hypervisor technologies, including XenServer, KVM, VMware vSphere ESXi and Microsoft Hyper-V. It provides networking through Neutron, object storage through Swift, image services through Glance, and block storage through Cinder for information storage. Additionally, it enables resource scalability up or down according to demand and high availability for all services. An organization embracing OpenStack can easily and swiftly deploy applications in a public or private cloud environment. Its expandable, modular architecture can cater to the organization's needs. It is a cloud computing platform that operates on open-source technology. It provides infrastructure as a Service (IaaS) in a virtualized environment. Businesses can use it to create private clouds that offer the same benefits as traditional cloud computing. It comprises several modular components, allowing users to manage cybernetics, storage, and networking resources efficiently. The core functionality of OpenStack includes:

- *Cybernetics:* It is the foundational element of the OpenStack assortment of products, allowing users access to virtual machines (VMs) for use in software and service applications. Compute supports on-demand deployment and scaling of VMs in response to changing user demand or application requirements. It also allows users to migrate VMs between physical hosts across public or private clouds using live migration capabilities provided by OpenStack's Nova compute service.

- *Storage:* It provides persistent data storage services such as block storage (Cinder), object storage (Swift), and file systems (Manila). Block storage offers persistent block-level disks attached directly to the VMs, while object storage stores data objects in distributed clusters across multiple nodes for highly available cloud-scale applications such as media streaming services or content delivery networks. File systems provide shared file systems running on top of block devices for distributed file systems such as GlusterFS for large-scale projects requiring high availability across multiple racks or regions.

- *Networking:* It provides highly scalable networking support with access control lists (ACLs), allowing fine-grained control over traffic within user networks through features like security groups and firewalls along with configurable IP address ranges defined by customizable subnets called VPN. Networking also offers quality service (QoS) guarantees when configuring network load balancers supported by OpenStack's Neutron API service used in public clouds where multiple tenants share physical infrastructure resources like web hosting platforms.

- *Orchestration & Automation:* It provides IT operational automation tools such as Heat templates enabling automated deployment, configuration management, software updates, etc., along with integration with external provisioning tools like Chef

Puppet Ansible, etc., allowing network administrators greater control over operations. It also offers built-in support from popular DevOps platforms like Jenkins, allowing developers/operations teams to foster better collaboration when deploying new apps/services into production.

- *Identity & Access Management:* It enforces authentication authorization policy through identity federation standards (SAML/OAuth2), enabling single sign-on experiences when managing multiple accounts across different domains while ensuring user access remains secure through a token-based authentication process at each stage. It is essential when dealing with multi-tenant deployments where proper segregation must be enforced amongst users accessing shared resources without compromise or security lapse.

3.1.2 KubeEdge

It is an open-source framework for managing data and workload in an edge computing environment. It enables the dynamic orchestration of distributed applications across clusters of edge nodes and clouds to provide low-latency services to end users [31]. It leverages the existing platforms from popular cloud providers like Amazon Web Services (AWS) and Google Cloud Platform (GCP). This allows developers to benefit from these platforms' scalability, reliability, and security features while still having a unified control plane for managing their edge computing deployments. It provides powerful cloud-native platform capabilities such as application scheduling, service discovery, monitoring and logging, resource management, deployment & configuration management, which are required to develop complex distributed applications on heterogeneous environments. Applications can be orchestrated in real-time across multiple clusters located at different geographical locations without deploying hardware or software components on each node separately. This makes it easier for enterprises to manage their edge deployments with reduced operational overhead.

It also enables application development using a containerized [33] microservices-based architecture that allows developers to rapidly iterate through development cycles by leveraging server-less functions like AWS Lambda or Google Cloud Functions, which are triggered by events occurring at the edge nodes or by external systems such as IoT devices sending telemetry data into the cluster. These serverless functions can pre-process data before it is sent back into the cloud for further processing via machine learning algorithms hosted in public clouds such as Azure Machine Learning Service or Amazon SageMaker. By decoupling application code from infrastructure details, developers have much more flexibility when developing large-scale applications. They can quickly deploy updates across multiple edges whenever needed instead of needing manual intervention each time an update is required.

Overall, KubeEdge provides an ideal solution when deploying applications across distributed edges while leveraging existing public cloud services. It enables organizations to reduce costs associated with traditional centralized model architectures while maintaining a high availability level due to its fault-tolerant nature. It also opens up opportunities for organizations who wish to leverage advanced technologies like machine learning since they now have access to powerful processing capabilities even on devices located remotely using Kubernetes running in proximate environments without having any hardware-specific constraints imposed on them. Figure 2 illustrates the fundamental architectural principle of KubeEdge, which aims to establish interfaces that align with Kubernetes on both the cloud and edge domains. To facilitate the implementation of a highly effective edge computing system, KubeEdge is comprised of an assortment of components, namely Edged, EdgeHub, CloudHub, EdgeController, EventBus, DeviceTwin, and MetaManager [34].

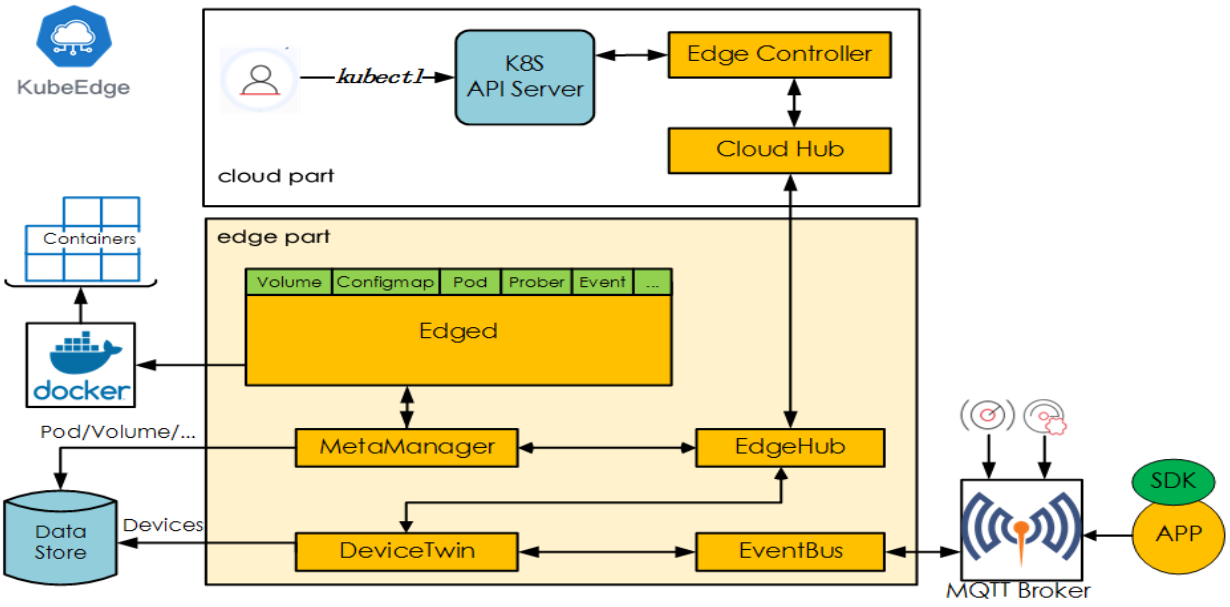

Fig. 2 Architecture of KubeEdge [32]

3.1.3 OpenEdge

Virtualization-based frameworks, like OpenEdge in edge computing environment, are designed to provide users with a cost-effective, secure and highly scalable cloud computing platform. The OpenEdge platform [35] enables consumers to rapidly establish and administer software services through little effort and expense. The framework includes a variety of capabilities, notably the potential to construct virtual machines on consumer demand, automatic resource escalating depending on consumption, high reliability, and assistance for different operating systems, including Windows, Ubuntu, Mac, OS X, and FreeBSD. It also stipulates the simple installation of outside vendors into the infrastructure, such as Amazon Web Services (AWS) or Microsoft Azure, and the installation of applications in minutes using assembled layouts and customized configurations. Customers using OpenEdge can automatically replicate their data for disaster recovery across different geographical locations within the cloud computing infrastructure.

Additionally, it provides advanced monitoring capabilities so that users can quickly identify performance issues in their applications before they become a problem. Furthermore, its automation capabilities allow developers to develop applications rapidly without manually configuring each component separately [36]. This will enable them to focus on development instead of spending time configuring the infrastructure themselves. The main benefit of using OpenEdge's edge computing environment is that it is cost-effective compared to other cloud providers while still providing all the same features. It also makes it easier for organizations looking to leverage cloud computing technologies without having to invest in expensive hardware or software solutions upfront since all necessary components are included in the platform itself, which simplifies deployments significantly compared to traditional methods but ensures reliable performance across all use cases regardless of their size or nature at any given point in time making sure maximum value is delivered from every service instance running within this context thus enabling organizations worldwide realize their desired results from deploying applications within this platform.

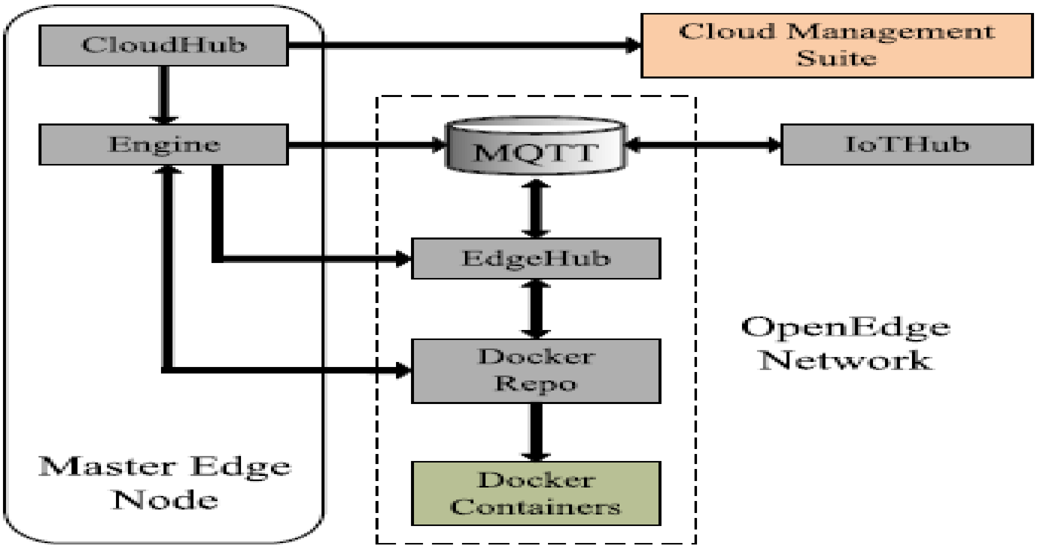

Fig. 3 Architecture of OpenEdge [37]

The OpenEdge platform has a containerized and modular architectural framework. The OpenEdge platform consists of two fundamental mechanisms: a central controller process and loadable modules functioning as plugins. These components operate at the edge of the system [38]. OpenEdge extraneously supports two distinct operating modes, notably the native process and the docker container mode. In the initial phase, applications are executed within docker containers, while the subsequent approach employs native Linux processes. Figure 3 illustrates the essential components of OpenEdge when functioning in the docker container approach.

The grey modules depicted in Figure 3 symbolize the OpenEdge components, in contrast to the orange modules, which indicate the cloud components. Initially, the engine component on the primary edge node acquires IoT signals from the IoTHub component by utilizing the Message Queuing Telemetry Transport (MQTT) message queue. The Engine segment facilitates the interaction between the EdgeHub and Docker Repo components, enabling the handling of events and execution of relevant computations within Docker containers. The Engine component has the capability to establish communication with the Cloud Management Suite component for different scenarios, such as synchronizing the master edge node with the cloud and fetching Docker images that are not cached.

3.2 *Virtualization techniques*

The potential of virtualization in cloud computing is immense. It enables organizations to leverage the power of the cloud to meet their needs while also allowing them to reduce costs and increase agility. Virtualization enables enterprises to quickly deploy and configure applications on cloud infrastructure without tedious tasks or protracted wait times. This makes it more intuitive for organizations to manage convoluted applications that demand adaptability, versatility, and integrity while remaining economically viable. Organizations can adopt virtualization techniques to lower their overall infrastructure expenditures as they no longer need to handle hardware and software autonomously. These techniques are becoming more prevalent in edge computing. This advanced technology offers more versatility and efficacy than conventional cloud computing solutions by enabling several VMs to run on one tangible server. Utilizing a single server to host multiple VMs can enhance computational capabilities and minimize operational expenses. Additionally, this approach streamlines application and service management, enabling faster deployment and configuration of new services without requiring additional hardware or reconfiguration [39]. Users can access many cloud services by leveraging these techniques within edge systems, elevating their experience and streamlining their operations. For instance, a company can deploy VMs with various OSs tailored to their unique requirements and implement load balancers to distribute traffic evenly across all servers, achieving performance.

Furthermore, data can be seamlessly distributed between plenty of VMs using scrambled tunnels or storage platforms such as Amazon S3 buckets, which are protected from unauthorized access or intruders while remaining readily accessible to people in the organization. In addition to increased scalability and performance benefits, virtualization-based frameworks offer cost savings compared to traditional cloud infrastructure solutions due to their higher resource utilization rates (i.e., fewer physical servers) [40]. Furthermore, organizations can quickly scale up or down based on their current needs without worrying about over-provisioning or under-provisioning resources, avoiding costly mistakes that *could* occur when manually setting up traditional infrastructure. Finally, this framework allows organizations greater flexibility when it comes time to migrate away from a current system due to its compatibility with existing applications and code bases, making transitions smoother overall. In edge computing, numerous virtualization techniques enable applications that utilize various operating systems to coexist on the same host, including:

3.2.1 Container-based Virtualization

Containerization, a system virtualization, allows multiple isolated containers to run on the same physical hardware infrastructure. Each container can contain a separate application or service with its resources and dependencies. It efficiently manages different environments within a single OS instance and allows users to spin up new instances quickly. It is a virtualization technology that enables applications to be deployed and securely isolated from one another in the same physical or cloud environment. It differs from traditional virtualization [41], which uses a hypervisor layer to separate multiple operating systems and applications running on the same hardware.

The representation of container-based virtualization is illustrated in Figure 4. A container is a standard unit of software that packages up code and all its dependencies so the application can run quickly and reliably from one computing environment to another. Container virtualization technologies offer an efficient way to rapidly package and deploy applications using lightweight, stand-alone containers that contain their own operating system, libraries, files, configuration settings, and any other software needed for execution. The containers can then be deployed on any machine regardless of architecture as long as the underlying host supports it.

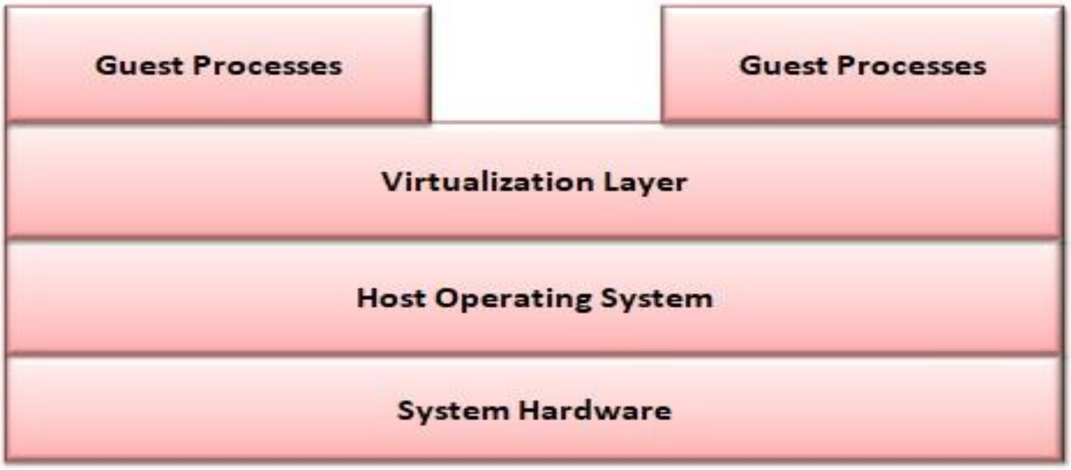

Fig.4 Architecture of container-based virtualization

Containers provide great portability because they are portable across different environments and platforms such as Windows, Linux or Mac operating systems; public clouds like AWS or Google Cloud Platform; bare metal servers; private data centres; and edge devices such as IoT sensors or mobile phones. Another advantage of container virtualization is that it offers greater security by isolating processes in each container so resources cannot be accessed outside its containerized environment. This ensures data integrity regardless of where they are deployed across various infrastructures while providing more secure isolation between workloads, even if they have been developed with different coding languages or libraries used within them. Containers also reduce overheads associated with traditional server-level virtualization, such as installation costs, due to their ability to scale exceptionally across multiple hosts with minimal manual intervention required for deployment tasks.

3.2.2 Hardware Virtualization

Hardware virtualization (a machine or hypervisor-based virtualization) is a technology that allows one physical server to act as several separate virtual machines. It is also called hardware virtual machine (HVM) and server virtualization. Regardless of what it's called, this technology is incredible. It enables multiple operating systems, applications, and services to run simultaneously on the same physical server. It means more tasks can be done with fewer resources. To do this, hardware virtualization runs a hypervisor on the physical server. They act as a middleman between the hardware and various systems running on different VMs. Since each VM has its resources, like memory, CPU cores, and storage space from the underlying hardware, they can all be optimized for the task at hand without resource contention. Besides optimizing its security features, HVM protects individual VMs from threats caused by other VMs on the same host machine [42]. Figure 5 depicts the representation of hypervisor-based virtualization.

The benefits of using HVM include increased efficiency since it eliminates redundant servers and reduces power consumption in data centres. It doesn't affect other VMs running on the same host. It also offers reduced cost compared to maintaining multiple physical servers, greater scalability due to increased speed when provisioning new VMs, enhanced security due to isolation between individual VMs, simpler management through centralized control over a more significant number of machines all at once and easier migration since moving applications between different hosts only requires copying files instead of rebuilding entire servers.

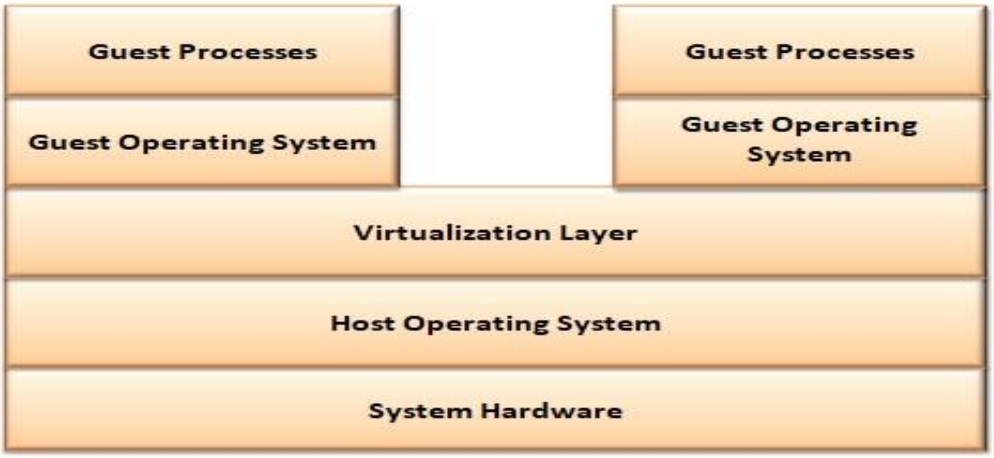

Fig. 5 Architecture of hypervisor-based virtualization

3.2.3 Software Virtualization

Software virtualization creates a different version of an OS, application, server, or storage system on a host computer. Because of this, organizations can run multiple applications and OS at the same time in the same environment. It makes it easy to use different versions of the same software without affecting the other apps. It also makes it easier to move them between systems. It can be used for cloud computing purposes, such as running different web servers on one machine or deploying legacy applications on new platforms without needing to install and configure them individually [43]. In addition, it allows organizations to instantly scale their infrastructure resources with minimal effort by spinning up new servers when needed and quickly shutting them down when they are no longer required.

It provides many benefits compared with traditional physical systems, such as improved resource utilization, more efficient IT operations management, better control over cost and performance requirements due to its more flexible deployment options, and increased security by isolating workloads within a single environment. However, some potential disadvantages associated with software virtualization include increased complexity in managing multiple domains and advanced networking and storage requirements due to numerous OS's sharing resources across networks. This can reduce overall system performance if not properly managed.

3.2.4 Network Function Virtualization (NFV)

NFV is a new approach to deploying network services. The functions and services traditionally supported by dedicated hardware are instead supported through software running on standard, off-the-shelf servers. NFV enables telecom providers to become more agile in their operations and services and reduce costs. NFV enables networks with multiple functions like routing/switching/firewall [44] capabilities, etc., running over common off-the-shelf data centre infrastructure; this helps reduce costs associated with buying specialized routers/switches, etc., while still maintaining performance levels required by enterprises today when deploying complex network topologies over public cloud environments. Table 2 depicts different virtualization techniques in edge computing systems. The benefits of NFV include:

- *Reduced upfront capital expenditure (CAPEX):* Providers can save significantly by replacing expensive dedicated hardware with virtualized servers.

- *Improved operational flexibility and scalability:* Service providers can quickly increase capacity when required or move resources to different areas of their network as needed by virtualization or pooling resources.

- *Increased speeds for deploying new services and features:* The ability to deploy virtualized platforms can facilitate faster rollouts of new services compared to traditional deployments that require more manual intervention from an engineering team.

- *Ability to support ever-increasing data demands while minimizing total cost of ownership (TCO):* By using commodity hardware rather than proprietary systems, operators can take advantage of economies of scale and reduce costs over time by leveraging existing infrastructure investments such as data centres and cloud computing capabilities without having to invest in additional hardware or software licenses.

Table II. Virtualization techniques in edge computing

Virtualization	*Category*	*Techniques Used*	*Key Characteristics*
Container	Server	OS-level Virtualization and resource isolation	Automated deployments allow for scaling and support portability across multiple platforms.
Hardware	Platform	Hypervisors	Allows for the virtualization of physical infrastructure and the consolidation of resources.
Software	Application layer	Emulation and sandboxing	It helps developers to test applications on different operating systems, improves scalability, and reduces costs.
Network Function	Network	Abstraction of network components, disaggregation of hardware and software resources	It enhances agility and flexibility in network operations and provides abstraction between network functions and hardware.

3.3 Open issues

Virtualization in an edge environment faces various issues, including managing multiple heterogeneous data sets, ensuring end-to-end QoS, integrating existing physical environments with new NFV solutions, and ensuring security and cost reduction for large-scale deployments. In addition, ensuring security and compliance with regulations is challenging. However, service providers must address several critical challenges to realize these benefits, as shown in Figure 6.

3.3.1 Ensuring end-to-end Quality of Service (QoS)

Ensuring end-to-end QoS across the entire network architecture when transitioning from physical appliances to virtualized ones is one of the leading open challenges that need to be addressed for NFV solutions to become more widely adopted. To obtain QoS, the network engineers must have a good grip on the implemented virtualization technologies and consider all possible scenarios. Even though it may seem obvious, performance testing should be conducted at every step of deployment and maintenance to guarantee a consistent level of service.

3.3.2 Integration of existing physical environments with new NFV solutions

A big challenge that needs to be addressed in the world of NFV solutions is integrating them into a physical environment. One way to do this is by migrating legacy systems onto cloud-based infrastructure or managing compatibility issues between vendor solutions. But it's not as easy as it sounds. With these processes, we have to ensure that the quality of the service provided by existing infrastructure stays the same, making it even more challenging. Compatibility issues make it necessary for an open ecosystem for interoperability between vendors to carry out these integration processes without sacrificing quality in the process.

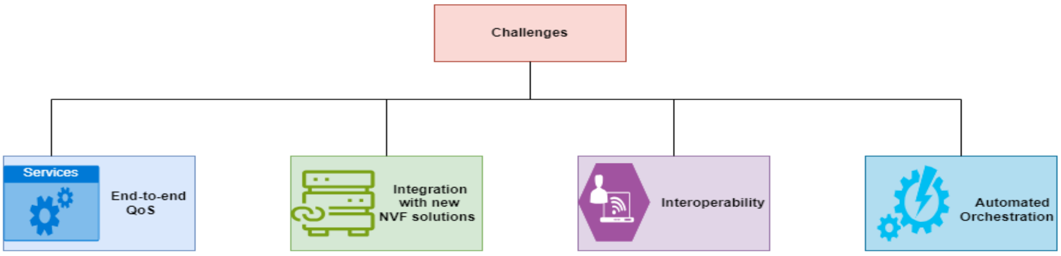

Fig. 6 Challenges associated with the adoption of virtualization in edge computing

3.3.3 Building an open ecosystem for interoperability between vendors

Building an open interoperability ecosystem between vendors in edge computing virtualization is challenging. It requires vendor cooperation to ensure seamless communication, making their applications compatible and secure for cloud platforms. The challenge also requires careful planning and implementation of standards to ensure the security and reliability of data.

3.3.4 Developing automated orchestration processes for fast deployment times

Developing automated orchestration processes for rapid deployment times in virtualization within edge computing environments is also necessary when implementing NFV solutions since manual configurations are time-consuming and prone to errors that could significantly impact performance levels. Automated orchestration can ensure consistent deployments across different VMs and considerably reduce time spent on deployments compared with manual methods, thus making it an essential process within any successful Virtualization project or migration towards NFV services.

3.3.5 Security

Virtualization in an edge computing environment presents security challenges due to networks' increased complexity and distributed nature. Cloud providers must ensure that virtual machines are adequately isolated from each other and protect against malicious attacks on their systems. Enforcing access control policies is vital to protect sensitive data stored in the cloud.

3.3.6 Cost Efficiency

The ability to optimize costs is critical when operating an edge environment because it allows consumers to get the most possible from their expenditures in facilities and amenities while still maintaining superior performance levels for their users' experiences. However, this necessitates meticulous planning when selecting hardware assets to maximize cost savings without compromising consistency or safety concerns.

IV. CONCLUSION AND FUTURE VISIONS

Virtualization has transformed the way in which organizations implement and manage their technology systems in edge computing environments by providing a reliable and cost-effective platform for hosting services and applications. This article highlights the significance of virtualization in edge computing. It explores the potential benefits such as reduced cost, efficient resource utilization, and improved scalability for organizations using virtualization in edge computing. Additionally, it describes the virtualization framework, techniques, and tools utilized in edge computing. Moreover, it also discusses the flexibility of virtualization in supporting different deployment approaches like containers, hypervisors, and virtual machines on edge nodes. The article highlights some issues that need to be resolved to utilize the full potential of virtualization in edge computing. These challenges include integration with cloud computing, ensuring quality of service, addressing security concerns, and optimizing cost efficiency. It is essential to implement virtualization in edge computing to overcome these challenges.

Future research focuses on the crucial development of standardized virtualization architectures to facilitate seamless integration within edge computing environments. This endeavour necessitates further investigations to address the outlined challenges. Additionally, the importance of raising awareness and actively engaging stakeholders regarding the promising benefits of virtualization in edge computing scenarios cannot be overstated.

Overall, this study emphasizes the pivotal role of virtualization in edge computing, advocating for continued research efforts, standardized architectural frameworks, and active stakeholder engagement to harness its transformative potential fully within this domain.

V. REFERENCES

[1] Q. Zhang, C. Li, Y. Huang, and Y. Luo, "Effective multi-controller management and adaptive service deployment strategy in multi-access edge computing environment," Ad Hoc Netw., vol. 138, p. 103020, 2023.

[2] M. Hussain, L.-F. Wei, A. Lakhan, S. Wali, S. Ali, and A. Hussain, "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing," Sustain. Comput. Inform. Syst., vol. 30, p. 100517, 2021.`

[3] A. Banushri and D. R. A. Karthika, "Implementation levels of Virtualization and security issues in cloud computing," Int. J. Eng. Technol., vol. 7, no. 3.3, p. 678, Jun. 2018, doi: 10.14419/ijet.v7i2.33.15474.

[4] M. Guerriero, M. Garriga, D. A. Tamburri, and F. Palomba, "Adoption, support, and challenges of infrastructure-as-code: Insights from industry," in 2019 IEEE International conference on software maintenance and evolution (ICSME), IEEE, 2019, pp. 580–589.

[5] K. K. Jyothi and B. I. Reddy, "Study on virtual private network (VPN), VPN's protocols and security," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 3, no. 5, pp. 919–932, 2018.

[6] M. S. Al-Asaly, M. A. Bencherif, A. Alsanad, and M. M. Hassan, "A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment," Neural Comput. Appl., pp. 1–18, 2021.

[7] C. Zhang, M. Dong, and K. Ota, "Fine-grained management in 5G: DQL based intelligent resource allocation for network function Virtualization in C-RAN," IEEE Trans. Cogn. Commun. Netw., vol. 6, no. 2, pp. 428–435, 2020.

[8] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (CAVs)," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 7, pp. 6240–6259, 2021.

[9] R. T. Rodoshi, T. Kim, and W. Choi, "Resource management in cloud radio access network: Conventional and new approaches," Sensors, vol. 20, no. 9, p. 2708, 2020.

[10] Q. Huang, F. Gao, R. Wang, and Z. Qi, "Power Consumption of Virtual Machine Live Migration in Clouds," in 2011 Third International Conference on Communications and Mobile Computing, Apr. 2011, pp. 122–125. doi: 10.1109/CMC.2011.62.

[11] A. Belbekkouche, Md. M. Hasan, and A. Karmouch, "Resource Discovery and Allocation in Network Virtualization," IEEE Commun. Surv. Tutor., vol. 14, no. 4, pp. 1114–1128, 2012, doi: 10.1109/SURV.2011.122811.00060.

[12] A. Gupta, O. Sarood, L. V. Kale, and D. Milojicic, "Improving HPC Application Performance in Cloud through Dynamic Load Balancing," in 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, May 2013, pp. 402–409. doi: 10.1109/CCGrid.2013.65.

[13] G. Wang, C. Liu, and J. Lin. "Transparency and semantics coexist: When malware analysis meets the hardware assisted Virtualization" in Trustworthy Computing and Services: International Conference, ISCTCS 2013, Beijing, China, November 2013, Revised Selected Papers, pp. 29-37. Springer Berlin Heidelberg.

[14] X. Song, Y. Ma, and D. Teng, "A Load Balancing Scheme Using Federate Migration Based on Virtual Machines for Cloud Simulations," Math. Probl. Eng., vol. 2015, p. e506432, Mar. 2015, doi: 10.1155/2015/506432.

[15] A. Zhou et al., "Cloud Service Reliability Enhancement via Virtual Machine Placement Optimization," IEEE Trans. Serv. Comput., vol. 10, no. 6, pp. 902–913, Nov. 2017, doi: 10.1109/TSC.2016.2519898.

[16] J. Du, L. Zhao, J. Feng and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems With Min-Max Fairness Guarantee," in IEEE Transactions on Communications, vol. 66, no. 4, pp. 1594-1608, April 2018, doi: 10.1109/TCOMM.2017.2787700.

[17] C. Pahl, A. Brogi, J. Soldani and P. Jamshidi, "Cloud Container Technologies: A State-of-the-Art Review," in IEEE Transactions on Cloud Computing, vol. 7, no. 3, pp. 677-692, 1 July-Sept. 2019, doi: 10.1109/TCC.2017.2702586.

[18] M. Alam, J. Rufino, J. Ferreira, S. H. Ahmed, N. Shah and Y. Chen, "Orchestration of Microservices for IoT Using Docker and Edge Computing," in IEEE Communications Magazine, vol. 56, no. 9, pp. 118-123, Sept. 2018, doi: 10.1109/MCOM.2018.1701233.

[19] G. M. Suranegara, D. A. Marenda, R. Hakimi, A. C. Risdianto and E. Mulyana, "Design and Implementation of VM Migration Application on SDN-Based Network," 2018 4th International Conference on Wireless and Telematics (ICWT), Nusa Dua, Bali, Indonesia, 2018, pp. 1-6, doi: 10.1109/ICWT.2018.8527782.

[20] Y. Zeng, M. Chao and R. Stoleru, "EmuEdge: A Hybrid Emulator for Reproducible and Realistic Edge Computing Experiments," 2019 IEEE International Conference on Fog Computing (ICFC), Prague, Czech Republic, 2019, pp. 153-164, doi: 10.1109/ICFC.2019.00027.

[21] Z. Shen et al., "X-Containers: Breaking Down Barriers to Improve Performance and Isolation of Cloud-Native Containers," in Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, in ASPLOS '19. New York, NY, USA: Association for Computing Machinery, Apr. 2019, pp. 121–135. doi: 10.1145/3297858.3304016.

[22] S. Lin, W. Liang and J. Li, "Reliability-Aware Service Function Chain Provisioning in Mobile Edge-Cloud Networks," 2020 29th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 2020, pp. 1-9, doi: 10.1109/ICCCN49398.2020.9209732.

[23] C. R. Storck, E. E. de O. Lousada, G. G. de O. Silva, R. A. F. Mini, and F. Duarte-Figueiredo, "FiVH: A solution of inter-V-Cell handover decision for connected vehicles in ultra-dense 5G networks," Veh. Commun., vol. 28, p. 100307, Apr. 2021, doi: 10.1016/j.vehcom.2020.100307.

[24] Z. Kuai, T. Wang, and S. Wang, "Fair Virtual Network Function Mapping and Scheduling Using Proximal Policy Optimisation," IEEE Trans. Commun., vol. 70, no. 11, pp. 7434–7445, Nov. 2022, doi: 10.1109/TCOMM.2022.3211071.

[25] W. Attaoui, E. Sabir, H. Elbiaze, and M. Guizani, “VNF and CNF Placement in 5G: Recent Advances and Future Trends,” IEEE Trans. Netw. Serv. Manag., pp. 1–1, 2023, doi: 10.1109/TNSM.2023.3264005.

[26] C. Yang, F. Liao, S. Lan, L. Wang, W. Shen, and G. Q. Huang, "Flexible resource scheduling for software-defined cloud manufacturing with edge computing," Engineering, vol. 22, pp. 60–70, 2023.

[27] J. Dogani and F. Khunjush, "Proactive Auto-Scaling Technique for Web Applications in Container-Based Edge Computing Using Federated Learning Model," J. Parallel Distrib. Comput., p. 104837, 2024.

[28] C. Guo and A. Rezaeipanah, "Dynamic service function chains placement based on parallelized requests in edge computing environment," Trans. Emerg. Telecommun. Technol., vol. 35, no. 1, p. e4905, Jan. 2024, doi: 10.1002/ett.4905.

[29] Z. Benomar, F. Longo, G. Merlino, and A. Puliafito, "Enabling container-based fog computing with openstack," in 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2019, pp. 1049–1056.

[30] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, "Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review," J. Syst. Softw., vol. 136, pp. 19–38, 2018.

[31] S. Wang, Y. Hu, and J. Wu, "Kubeedge. ai: Ai platform for edge devices," ArXiv Prepr. ArXiv200709227, 2020.

[32] T. Rausch, A. Rashed, and S. Dustdar, "Optimized container scheduling for data-intensive serverless edge computing," Future Gener. Comput. Syst., vol. 114, pp. 259–271, 2021.

[33] A. Ullah, H. Dagdeviren, R. Ariyattu, J. DesLauriers, T. Kiss, and J. Bowden, "MiCADO-Edge: Towards an Application-level Orchestrator for the Cloud-to-Edge Computing Continuum," J. Grid Comput., vol. 19, Dec. 2021, doi: 10.1007/s10723-021-09589-5.

[34] Z. Cai et al., "RBaaS: A robust blockchain as a service paradigm in cloud-edge collaborative environment," IEEE Access, vol. 10, pp. 35437–35444, 2022.

[35] S. Shahzadi, M. Iqbal, T. Dagiuklas, and Z. U. Qayyum, "Multi-access edge computing: open issues, challenges and future perspectives," J. Cloud Comput., vol. 6, pp. 1–13, 2017.

[36] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.

[37] J. Straub, "Automating software design and configuration for a small spacecraft," in Unmanned Systems Technology XVI, SPIE, 2014, pp. 37–45.

[38] T. Sánchez López, D. C. Ranasinghe, M. Harrison, and D. McFarlane, "Adding sense to the Internet of Things: An architecture framework for Smart Object systems," Pers. Ubiquitous Comput., vol. 16, pp. 291–308, 2012.

[39] B. I. Ismail et al., "Evaluation of docker as edge computing platform," in 2015 IEEE conference on open systems (ICOS), IEEE, 2015, pp. 130–135.

[40] J. M. Kizza, "Virtualization Technologies and Security," in Guide to Computer Network Security, in Texts in Computer Science. , Cham: Springer International

Publishing, 2024, pp. 503–518. doi: 10.1007/978-3-031-47549-8_22.

[41] M. S. Aslanpour, A. N. Toosi, M. A. Cheema, M. B. Chhetri, and M. A. Salehi, "Load balancing for heterogeneous serverless edge computing: A performance-driven and empirical approach," Future Gener. Comput. Syst., 2024.

[42] R. Kumar and S. Charu, "An importance of using Virtualization technology in cloud computing," Glob. J. Comput. Technol., vol. 1, no. 2, 2015.

[43] Y. Xing and Y. Zhan, "Virtualization and cloud computing," in Future Wireless Networks and Information Systems: Volume 1, Springer, 2012, pp. 305–312.

[44] K. Joshi and T. Benson, "Network function Virtualization," IEEE Internet Comput., vol. 20, no. 6, pp. 7–9, 2016.