2022 Australian & New Zealand Control Conference (ANZCC)
24-25 November 2022. Gold Coast, Australia

An Edge Computing Architecture Based on Unikernel

Shichao Chen
The State Key Laboratory for Management and Control of Complex System, Institute of Automation, Chinese Academy of Sciences
The Center of National Railway Intelligent Transportation System Engineering and Technology,
China Academy of Railway Sciences Corporation Limited
Beijing, China
shichao.chen@ia.ac.cn

Ruijie Xu
The State Key Laboratory for Management and Control of Complex System, Institute of Automation, Chinese Academy of Sciences
College of Information Science & Technology,
Beijing University of Chemical Technology
Beijing, China
2019040108@mail.buct.edu.cn

Wenqiao Sun
Transportation and Economics Research Institute, and the Center of National Railway Intelligent Transportation System Engineering and Technology, China Academy of Railway Sciences Corporation Limited
Beijing, China
sunwenqiao@rails.cn

2022 Australian & New Zealand Control Conference (ANZCC) | 978-1-6654-9887-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ANZCC56036.2022.9966982

***Abstract*—In the era of big data, more and more applications of smart devices are computing-intensive, thus raising the strong demand for task offloading to cloud data centers. However, it gives rise to network delay and privacy data leak issues. Edge computing can effectively solve latency, bandwidth occupation and data privacy problems, but the deployment of applications are also limited by hardware architectures and resources, i.e., computing and storage resources. Therefore, the combination of virtualization technology and edge computing become important in order to realize the rapid deployment of intelligent application in an edge server or an edge node by virtualization technology. The traditional virtual machine (VM) is no longer suitable for resource-constrained devices. Container technique including Docker can effectively solve these problems, but it also depends on an operating system. Unikernel is the state-of-art virtualization technology. In this paper, we combine Unikernel with edge computing to explore its application in an edge computing system. An application architecture of edge computing based on Unikernel is proposed. It is suitable for application in edge computing.**

Keywords—unikernel, docker, virtualization technology, edge computing

I. INTRODUCTION

With the fast development of Internet of things (IoT), more and more smart devices are connected to the Internet. They inevitably produce a large amount of data. All the data can be transmitted to cloud data centers for processing. Yet this causes some problems, e.g., network delay [1] and privacy leak [2]. Although 5G technology can increase the network bandwidth and reduce the latency of data transmission, we face the storage and computing pressure for big data and the geographical location limitation of a cloud center [3]. In addition, many latency-sensitive applications need millisecond or even microsecond response time, e.g., automatic drive, telemedicine, virtual reality and industrial Internet. However, these problems are difficult to solve with a cloud computing paradigm. In order to solve them and promote the further applications of IoT [4], a new computing paradigm, called edge computing, is proposed [5], which is a good complement for cloud computing.

Edge computing can use edge nodes near the data sources to process and analyze data in real time. They do not need to offload all computing tasks to the cloud center [6]. Edge computing can effectively solve the problems of network delay, network congestion, and data security. But hardware resources of edge nodes are limited in comparison with a cloud center. In order to realize the rapid deployment of applications on heterogeneous edge nodes, virtualization techniques are needed to match with resource-constrained edge nodes.

Virtualization technology has been mature with the development of cloud computing. The traditional VM has been widely used in cloud computing. However, they need to load a complete operating system and initialize virtual resources when they start, thus making their start-up time too long [7]. Due to the large data (e.g., images) mage size and slow start-up speed, VM is not appropriate for edge computing, and it cannot respond to the request and processing of data quickly for edge nodes, especially for resource-constrained ones. Container including Docker can effectively solve some problems of VM, e.g., slow start-up speed and large resource consumption. Yet Docker's security is a concern in comparison with VM [8]. Although Containers are now used for the application deployment in many edge computing systems [9,10, 11], they meet some limitations in terms of lightweight and security. Considering the convenience of application deployment and the lightweight of image files, we have to find a more secure, efficient and lightweight virtualization technique. In recent years, Unikernel with the advantages of high security, high start-up speed and simplicity [12] has attracted researchers' interest in the field of virtualization. It is expected to be the core of the next generation of cloud computing and edge computing. Its concept was proposed in 2013 [13]. It can perfectly solve the problems of security and isolation in Container, and can be applied directly to hardware without concerning an operating system. We have observed research and projects about Unikernel, e.g., Mirage [14], HalVM [15], Osv[16], NetBSD rump[17], ClickOS[18], Drawbridge[19], IncludeOS [20], Clive [21], LING Unikernel [22], Runtime. Js [23], Unik [24], NanoVMs [25].

Based on the study of Unikernel, this work aims to combine Unikernel with edge computing to study its application in an edge computing system. In Section II, we introduce Unikernel's concept, characteristics and comparison with VM and Container. In Section III, based on the survey of virtualization techniques in an edge computing system, we propose the application architecture of an edge computing system based on Unikernel. Finally, the issues and prospective development of Unikernel in an edge computing system is summarized in Section V.

II. UNIKERNEL

A. Basic definition

At present, there is no universal definition of Unikernel. It is a specialized, single address space image constructed by library operating systems (Library OS). In order to support the execution of programs, developers can select the smallest set of class libraries from a module stack, and build the corresponding OS. The class library, together with application code and configuration files, are built into a dedicated image file. It can run directly on such hypervisors as Xen, Kernel-based Virtual Machine (KVM), or hardware without an operation system, e.g., Linux. From its existing definition, it can be known that its essence is a kind of image file with special purpose. Like Docker, it is a microservice solution and widely used in developing applications in a microservice architecture.

B. Architecture of Unikernel

Unikernel is divided into three parts over hardware platform or hypervisor: application, LibOS and platform adaptation layer (PAL), as shown in Fig. 1.

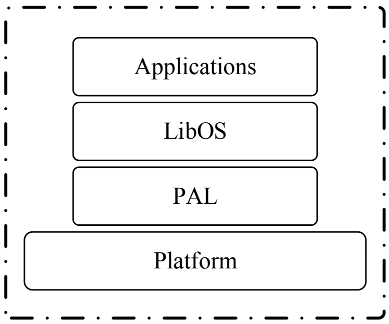

Fig. 1. The architecture of Unikernel

User applications are running at its top layer. These application programs are generally single function, small size and high performance, which are convenient for fast deployment. Their characteristics conform to microservice application [26]. LibOS is at its middle layer, which is specifically divided into a kernel base and functional system. The former is to complete the basic functions of OS, namely task scheduling and thread management. The latter is the minimum core components to support applications execution. PAL includes an adaptation layer with host environment running on a bare machine and virtual environment running on a hypervisor. PAL is mainly for LibOS application migration, and needs not change LibOS or application code when hardware is updated. Fig. 2 is about the comparison among Unikernel, Docker and VM architecture.

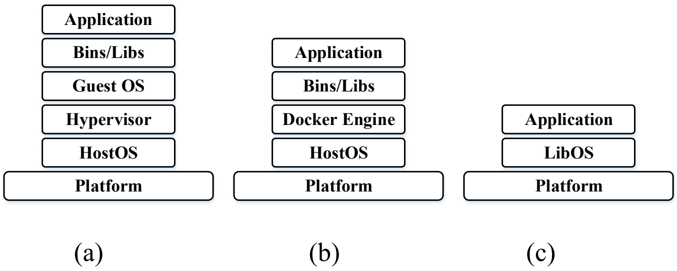

Fig. 2. The architecture comparison among (a) VM, (b) Docker and (c) Unikernel

Compared to VM and Container, Unikernel does not rely on an OS. Its applications can run directly on the bare hardware or hypervisor via LibOS. Applications are more lightweight and specialized. Because it is not similar to Container built on an OS, its security is assured. We can obtain its following characteristics in its process of practical utilization.

- High security. It does not involve an existing OS, and the attack probability of a hypervisor is less. Compared to the larger attack surface of Docker daemon, it has a great advantage over Docker in terms of security.
- Small size of image files. It only contains the system library needed by running applications, so it takes up fewer space resources than its peers.
- Fast start-up speed. It can start up in milliseconds and be deployed quickly and massively. Meanwhile, it can be started again when a user makes a request, which greatly improves the system efficiency.
- Specificity. There is only one process in its image, and no multi-process switching, which also guarantees its extremely lightweight.

III. ARCHITECTURE OF EDGE COMPUTING PLATFORM BASED ON UNIKERNEL

A. Virtualization techniques in edge computing

An edge computing system can manage heterogeneous computing, storage and network resources. It deals with a lot of heterogeneous data and application loads. It is responsible for the deployment, scheduling and migration of complex computing tasks on edge computing nodes, so as to ensure the reliability of computing tasks and the maximum utilization of hardware resources. In a word, it is more inclined to the management framework of data, computing tasks and computing resources [27].

With the increasing development of edge computing in research and industry, a number of edge computing systems with different functions have emerged for different application scenarios, e.g., ParaDrop, EdgeX Foundation, Apache Edgent, FocuStack and AirBox for IoT scenarios, CloudLet and PCloud for latency-sensitive mobile applications, and CORD, AKranino and Edge Stack for network operators.

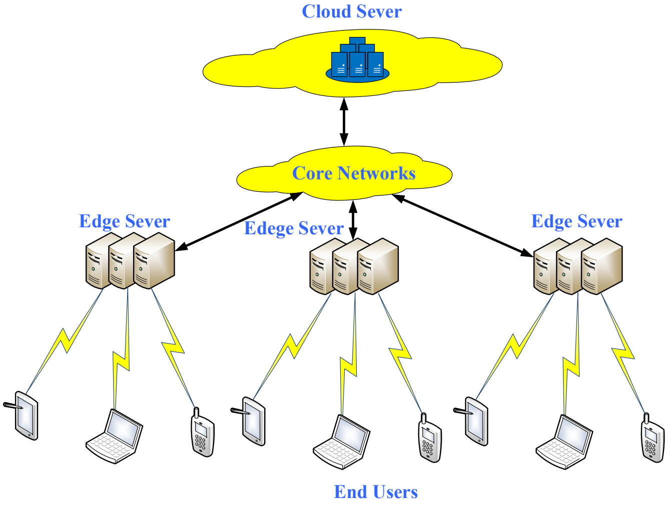

Fig. 3. The basic framework of edge computing

While comprehensively analyzing the architectures, functions and key techniques of the aforementioned systems is beyond the scope of this paper, we only focus on the combination of edge computing systems and virtualization techniques. Chen et al, [28] analyze the virtualization techniques used for edge computing. It can find that virtualization techniques are widely used due to their rapid service deployment and management capability. Especially, container is widely used in edge computing systems, because of its flexible deployment and management, small size and high start-up speed. Yet some edge computing systems adopt the virtualization mode of combining VMs and Container to manage the hardware resources and application services. However, it has almost not been aware the usage of the Unikernel in an edge computing system.

Due to Container, especially Docker, it adopts a client and server (C/S) architecture mode as shown in Fig. 4. A client runs a Docker tool and uses remote API to manage and create a Container. The host downloads image files from the repository to deploy various Containers, and communicates with the client. The repository is used to store application images. This architecture enables the services to be loaded on demand for the edge nodes. Therefore, it can realize the batch distributed deployment of applications. Edge nodes can deploy the environment required for Docker and directly run locally the executable Docker image files from a Docker hub.

Docker adopts process level isolation, which greatly improves the reuse rate of resources, speeds up the response time and makes the start-up speed of applications fast. The number of Docker that can run at the same time on an edge server is more than that in VMs. In terms of advantages, e.g., high resource reuse rate, low response time and high concurrency, it is appropriate for edge computing. However, application services based on Docker need the support of OS and many hardware resources, which will limit the large-scale and on-demand deployment of applications in IoT scenarios. Thus, new virtualization technologies are needed to enable rapid deployment of image files in IoT.

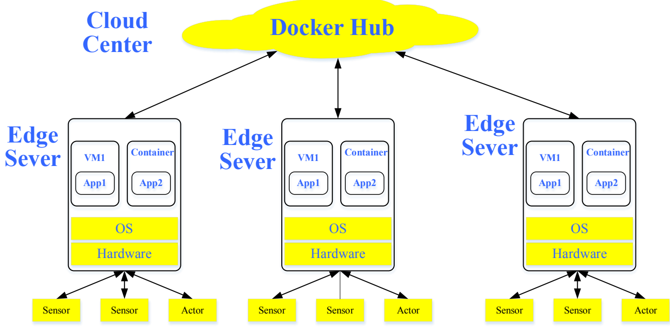

Fig. 4. General architecture of edge computing platform based on virtualization technology

B. Edge computing system based on Unikernel

According to the Docker used in an edge computing system, we propose a new architecture of an edge computing system based on Unikernel. It is shown in Fig. 5.

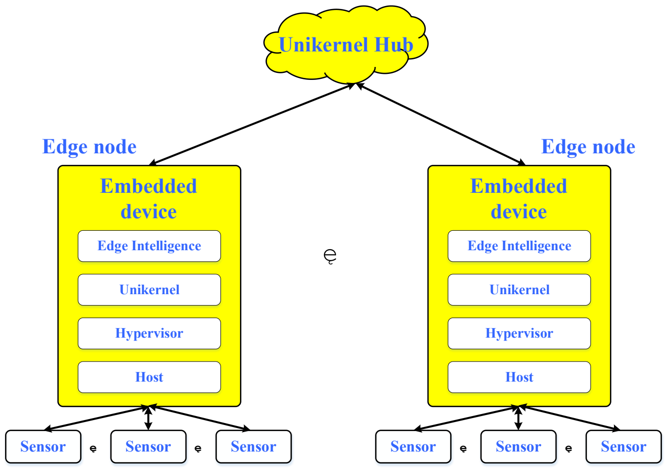

Fig. 5. Edge computing system based on Unikernel

Its architecture is similar to the computing system based on Container, including Unikernel hub, client, host and Unikernel image. A Unikernel hub is responsible for storing, managing and arranging image files. Client is responsible for the management and creation of image files. Host is responsible for running the image flies of applications in edge nodes. Because Unikernel image can run on the hypervisor or bare hardware, and dose not depends on OS to provide services, namely resources dispatching. Unikernel used in an edge computing system can break away from the dependency of OS. It can pull the image files of applications from the Unikernel hub by using a simple stand-alone program only, which makes the application easy to expand, especially for the IoT devices with limited resources.

In an edge computing system, a series of services such as collection, integration and analysis for data can be realized through the Unikernel applications. These processed results can be fed back to a cloud center in real time, so as to realize the edge's intelligent computing based on Unikernel. This lightweight solution is faster, less resource-consumed and more efficient than VMs and

Container.

IV. CONCLUSIONS

In this paper, we introduce the characteristics and architecture of Unikernel. It has some advantages with Container and VM. It can find the virtualization techniques are widely used in edge computing, especially Docker technique. A novel architecture of edge computing system based on Unikernel is proposed. It is more suitable for application in edge computing, especially for resource-constrained edge nodes.

ACKNOWLEDGEMENT

This work was supported by CAS STS Dongguan Project (20201600200132) and the Center of National Railway Intelligent Transportation System Engineering and Technology (Contract No. RITS2021KF03,2021YJ004), China Academy of Railway Sciences.

REFERENCES

[1] I. Sadooghi, et al. “Understanding the Performance and Potential of Cloud Computing for Scientific Applications,” IEEE Transactions on Cloud Computing, vol. 5, no. 2, pp. 358-371, 1 April-June 2017.

[2] M. Au, et al. “Privacy-preserving personal data operation on mobile cloud-Chances and challenges over advanced persistent threat.” Future Generation Computer Systems, vol. 79, pp. 337-349, 2018.

[3] F. Cicirelli, et al. “Edge enabled development of Smart Cyber-Physical Environments.” 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, pp. 003463-003468, 2016.

[4] Y. Xing, et al., "Advances in Vision-Based Lane Detection: Algorithms, Integration, Assessment, and Perspectives on ACP-Based Parallel Vision," in IEEE/CAA Journal of Automatica Sinica, vol. 5, no. 3, pp. 645-661, May 2018.

[5] W. Shi, et al. "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct. 2016.

[6] P. Lang, et al. "Cooperative Computation Offloading in Blockchain-Based Vehicular Edge Computing Networks," in IEEE Transactions on Intelligent Vehicles, 2022, doi: 10.1109/TIV.2022.3190308.

[7] J. Xu. “The comparison of virtual machine, container and Docker Technology.” Science and wealth, vol. 008, no. 004, pp. 364-364, 2016.

[8] T. Lu, et al. “Research on Docker security.” Computer technology and development, vol. 028, no. 006, pp. 115-120, 2018.

[9] M. Alam, et al. “Orchestration of Microservices for IoT Using Docker and Edge Computing.” IEEE Communications Magazine, vol. 56, no. 9, pp. 118-123, Sept. 2018.

[10] R. Ezhilarasie, et al. “Grefenstette Bias based genetic algorithm for multi-site offloading using docker container in edge computing.” Journal of Intelligent & Fuzzy Systems, vol. 36, no. 3, pp. 2419-2429, 2019.

[11] M. Hasan, S. Mohan, T. Shimizu and H. Lu. "Securing Vehicle-to-Everything (V2X) Communication Platforms," in IEEE Transactions on Intelligent Vehicles, vol. 5, no. 4, pp. 693-713, Dec. 2020.

[12] D. R. Engler, et al. “Exokernel: An operating system architecture for application-level resource management.” ACM SIGOPS Operating Sys- tems Review, vol. 29 no. 5, pp.251-266, 1995.

[13] A. Madhavapeddy, et al. “Unikernels: Library operating systems for the cloud.” ACM SIGARCH Computer Architecture News, vol. 41, no. 1, pp. 461-472, 2013.

[14] A. Madhavapeddy, et al. “Unikernels: library operating systems for the cloud.” In Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) pp. 461-472, 2013.

[15] Galois Inc. The Haskell Lightweight Virtual Machine (HaLVM) source archive.https://github.com/ Galois- Inc/HaLVM.

[16] Cloudius Systems. OSv.https://github.com/cloudius-systems/osv.

[17] A. Kantee. “Flexible operating system internals: the design and implementation of the any kernel and rump kernels.”, 2012.

[18] Y. Minsky. “OCaml for the masses”. Communications of the ACM, vol. 54, no. 11, pp. 53-58, 2011.

[19] D. E. Porter, et al. “Rethinking the library OS from the top down.” Proceedings of the sixteenth international conference on Architectural sup- port for programming languages and operating systems. pp. 291-304.

[20] Bratterud A, Walla A A, Haugerud H, et al. “IncludeOS: A minimal, resource efficient unikernel for cloud ser- vices.” 2015 IEEE 7th international conference on cloud computing technology and science (cloudcom). IEEE, pp. 250-257, 2015.

[21] F. J. Ballesteros. “The clive operating system.” Technical report, 2014.

[22] Erlangonxen. org. “Erlang on Xen: at the heart of super-elastic clouds”. (2017-12-25) [2018-01-20]. http://erlangonxen.org/.

[23] Runtime. js community. Runtime. Js-JavaScript library operating system for the cloud [EB/OL]. (2017-12- 25) [2018-01-20]. http://runtimejs.org/.

[24] Unik. [Online]. Available: https://github.com/solo-io/unik.

[25] NanoVMs. [Online]. Available: https://nanovms.com/.

[26] Y. Xin, et al. Overview of microservice architecture implementation framework. Computer Engineering and Applications, vol. 54, no. 19, pp. 10-17, 2018.

[27] W. Shi, et al. “Edge Computing: State-of-the-Art and Future Directions.” Journal of Computer Research and Development, vol. 56, pp. 69, 2019.

[28] S. Chen and M. Zhou, “Evolving Container to Unikernel for Edge Computing and Applications in Process Industry,” Processes, vol. 9, no. 2, p. 351, Feb. 2021, doi: 10.3390/pr9020351.