2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 978-1-6654-6483-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PIMRC56721.2023.10293911

Enhancing Edge Computing with Unikernels in 6G Networks

Syed Yazdani
Faculty of Science and Engineering
The University of Manchester
Manchester, UK
syedsamar.yazdani@manchester.ac.uk

Naeem Ramzan
School of Computing
University of the West of Scotland
Scotland, UK
Naeem.Ramzan@uws.ac.uk

Pierre Olivier
Faculty of Science and Engineering
The University of Manchester
Manchester, UK
pierre.olivier@manchester.ac.uk

***Abstract*—The next generation of mobile networks, 6G, when compared to 5G is expected to offer faster speeds, lower latency, and increased capacity. At the same time, edge computing is becoming increasingly important for providing low-latency services and reducing the load on centralized data centers. Unikernels, a lightweight form of virtualization, have the potential to improve the performance and security of edge computing systems. This paper investigates the utility of unikernel-based edge computing in 6G networks. The paper will begin by providing an overview of the current state of unikernel-based edge computing and its advantages over traditional virtualization techniques. It will then investigate the potential of using unikernels in 6G networks, including the ability to improve network performance and security. The research will also look into the potential of using unikernels in edge computing systems, such as the ability to reduce the load on centralized data centers, improve the performance of low-latency services, and improve edge computing systems security. Finally, the paper will point towards the research's implications and potential future work in the field.**

***Index Terms*—Virtualization, Unikernel, 6G**

I. Introduction

The arrival of 6G networks, with faster speeds, lower latency, and increased capacity over previous generations, promises significant advancements in mobile communication. Edge computing is becoming an increasingly important aspect of modern computing because it enables low-latency services and reduces the load on centralized data centers. Unikernels, a lightweight form of virtualization, have the potential to improve the performance and security of edge computing systems. This paper investigates the utility of using unikernel-based edge computing in 6G networks and how it can help improve 6G network capabilities.

Unikernels are good candidates for edge computing because of their small size and minimal attack surface. Unikernels are single-address space images that include only the necessary libraries and drivers for a specific application. This results in a smaller and more secure system, compared to traditional monolithic kernels [1]. Unikernels [1]–[8] have gained popularity in recent years as a lightweight and secure form of virtualization. A unikernel instance is intended to run a single application, hence it includes only a minimal amount of systems software (operating system, libraries, etc.) code required to execute that application. This leads to a significant reduction in the memory/disk footprint, boot time, as well as attack surface, when compared to traditional virtual machines [9]–[11].

Edge computing, on the other hand, is a distributed computing paradigm in which computation and data storage are moved closer to the devices that produce or consume data. As a result, processing speed is increased, latency is reduced, and network load is reduced. Edge computing, which allows data to be processed and stored closer to the devices that generate it, can further enhance 6G networks with faster speeds and lower latencies. Because of their small size and low attack surface, unikernels are good candidates for edge computing [12].

This paper investigates the feasibility of combining unikernels and edge computing in 6G networks. The study's aim is to find out how unikernels can be used to improve the performance and security of edge computing systems in 6G networks. How does the use of unikernels in edge computing systems affect the performance of low-latency services in 6G networks? How does the use of unikernels in edge computing systems affect the security of 6G networks?

To answer the first two questions, we contend that Unikernels can improve the performance and security of edge computing systems through faster boot times, improved performance, enhanced security, lower latency, improved resource utilization, improved network efficiency, and scalability. In response to the last question, the use of unikernels in edge computing systems can improve network security in a variety of ways, including reduced attack surface, improved isolation, enhanced resource management, controlled access to resources, and immutable infrastructure.

II. Unikernel

A. Definition/Presentation

Unikernel [2], [13] is a single-purpose appliance that is specialized at compile time into a stand-alone kernel and is protected from modification after deployment. Furthermore, it improves performance by removing unnecessary components from applications and increases security by reducing the attack surface. It was originally intended for cloud computing, but its small footprint and flexibility make it ideal for edge computing, especially the upcoming internet of things edge computing. As a result, each Unikernel may have a distinct set of vulnerabilities, implying that an attacker who successfully penetrates one may be unable to threaten others.

Unikernels have a smaller code base and only run the components that are required for the specific application, resulting in a smaller overall footprint. Unikernels run a single application, which reduces the attack surface and the possibility of compromise. They are optimized to make better use of system resources, which can result in lower latency and higher performance. Individual applications can be isolated in unikernels, reducing the risk of cross-application interference. They are self-contained and do not require a separate runtime environment, so they can be deployed quickly and easily.

Although unikernels have promise and are useful in a variety of situations, they do have some drawbacks, including complexity, a limited ecosystem, performance limitations, portability, security concerns, and a lack of debugging tools. However, there are certain approaches that could be promising for the debugging of unikernels [14]. Because the unikernel ecosystem is still in its infancy, there are few libraries and tools available. Because of their small size and limited runtime environment, they may also have performance limitations. Unikernels might not work with all hardware or cloud platforms. Because of their limited runtime environment, unikernels are more vulnerable to security threats. Due to the limited runtime environment and lack of standard tools, debugging unikernels can be difficult.

Unikernel is also primarily stateless. As a result, it can easily execute edge intelligent algorithms (such as compression, encryption, and network function virtualization). There are numerous Unikernel research projects, the most notable of which are MirageOS Unikernel [15], Hermitux [5], Unikraft [16], NanOS [17], IncludeOS [18], OSv Unikernel [7], ClickOS [19], and others [20]–[23].

Based on the preceding analysis and discussion, we conclude that Unikernel has a smaller image size and a very low memory consumption. It is appropriate for migration in a mobile edge computing environment, especially in vehicular networks. It is capable of responding quickly to user requests. Because of its small image size, it can run on low-resource edge devices. It can also help to ensure code integrity and ease of updating while maintaining high security isolation by reducing the attack surface. It has negligible OS overhead and is appropriate for running applications that require frequent context switching and processing small amounts of data.

B. Use cases

Unikernels are specialized, single-address-space machine images built with library-based operating systems. This technique has the advantages of higher resource utilization, a smaller footprint, improved security due to a smaller attack surface, and faster startup times. These characteristics make them ideal for edge computing scenarios requiring resource limits, minimal latency, and strong security. Some unikernel edge computing use cases are listed below:

- Real-time application: Unikernels are lightweight and efficient, so they can handle real-time edge-based applications like audio and video processing. This may include delivering real-time statistics for video surveillance or managing many audio streams concurrently in a conference call application.
- Edge-gateways: Unikernels can act as edge gateways, acting as middlemen between devices and the cloud. Unikernels gather data from various sources, process it, and then transfer it to the cloud. They can provide the necessary security in this multi-hop communication to maintain data integrity and confidentiality.
- Cloud-native Edge Computing: Unikernels are lightweight and efficient, so they can handle real-time edge-based applications like audio and video processing. This may include delivering real-time statistics for video surveillance or managing many audio streams concurrently in a conference call application.
- Network Functions: Unikernels can act as edge gateways, acting as middlemen between devices and the cloud. Unikernels gather data from various sources, process it, and then transfer it to the cloud. They can provide the necessary security in this multi-hop communication to maintain data integrity and confidentiality.
- Autonomous vehicles: Unikernels can be used in self-driving cars to provide real-time data processing and decision-making capabilities. Unikernels, due to their modest size and efficiency, may operate in limited contexts such as an on-board computer while maintaining security and low latency.

The choice of unikernels is often driven by a variety of factors:

- Resource efficiency: Unikernels are compact and efficient, making them excellent for edge computing applications where hardware resources are restricted.
- Security: Because of their small size, unikernels have a lower attack surface, resulting in a higher level of protection.
- Isolation: Because unikernels provide good isolation between diverse workloads, they are ideal for instances in which numerous tenants share the same hardware.
- Low latency: Because of their low latency and deterministic behavior, unikernels are suited for real-time applications that require quick response times.
- Simplified management: Unikernels facilitate the management and deployment of applications at the edge by minimizing the overhead of standard operating systems. This can help to simplify the management of a fleet of edge devices, each of which may be running a distinct application.

Edge computing is being transformed by the evolution of unikernels, which are addressing the specific issues it offers. As the technology evolves, we may expect to see even more adoption of unikernels in a variety of industries, particularly those that require efficient, secure, and real-time computing at the edge.

III. 6G Technologies

The next generation of mobile network technology is 6G [24], which comes after 5G. While 5G technology is still being implemented around the world, 6G development is already underway. 6G is expected to outperform 5G in a variety of ways, including: 6G is expected to have peak download speeds of up to 1 terabit per second (Tbps), which is significantly faster than 5G peak download speeds (which are around 20 Gbps). The latency of 6G will be as low as a few microseconds, which is significantly lower than the latency of 5G (which is around 1-2 milliseconds). It will support many more devices than 5G, allowing billions of devices to connect to the internet simultaneously. It is extremely reliable, making it suitable for use in critical applications such as industrial automation and autonomous vehicles. It seems that it is more energy-efficient than 5G, which will be important for reducing the environmental impact of mobile networks. It has enhanced security features to combat cyber attacks. 6G will introduce new technologies such as holographic communications and the use of terahertz frequencies, which are both being researched and tested.

The development of 6G technology is still in its early stages, and it may be several years before it is available to consumers.

IV. In a nut shell

A comprehensive edge computing solution based on unikernels has the potential to revolutionize network computing, particularly in the context of 6G networks. This proposal covers the major components of such a sophisticated system:

- Low-power Edge Devices: To satisfy the expectations of 6G networks, edge devices must be extremely power efficient without losing data processing capabilities. These devices, which are small in size yet high in performance, should be designed with energy efficiency in mind, successfully coupling with the high-speed data processing requirements of the 6G era.
- High Speed Network Interfaces: It is critical to design high-speed network interfaces to support the exponential increase in network traffic and the resulting increase in bandwidth and low latency demands of 6G networks. These interfaces should be able to provide effective data transfer rates while maintaining network speed and stability.
- Minimalistic Unikernels: Operating systems that are highly efficient and secure are required for 6G networks. To that aim, unikernels can offer an efficient solution by providing a minimalistic operating system that dramatically decreases the attack surface and minimizes system overhead, hence improving overall system security and efficiency.
- Custom Drivers: Custom drivers are required to interface unikernels with network interfaces, storage devices, and other hardware components. These drivers should be purpose-built to ensure smooth and secure communication between the unikernel and the rest of the hardware components.
- Centralized Management System: A centralized management system is essential given the complexity of deploying, configuring, and monitoring edge devices and unikernels. This system should be able to do these jobs swiftly and effectively, eliminating the requirement for manual intervention and lowering the chance of errors greatly.
- Analytics System: The large amount of data generated by 6G networks needs the use of a sophisticated data analytics solution. This system should be able to process, analyze, and provide useful insights regarding network performance, security, and usage. This information can be utilized to make informed network management and enhancement decisions.
- Comprehensive Security: As cyber dangers such as hacking and malware become more sophisticated, it is critical to develop a complete security system. This system should be built to defend the edge devices and unikernels from a wide range of attacks while maintaining system performance.

With their intrinsic characteristics, unikernels provide a solution that satisfies the high-performance, security, and flexibility objectives of 6G networks. They are much less sensitive to security threats because to their minimalistic code base, intrinsic isolation qualities, immutable image deployments, lack of legacy code, and system hardening. Unikernels use immutable images that cannot be changed at runtime, preventing attackers from compromising the system. They are also purpose-built for certain purposes, so there is no unnecessary old code that could provide an entry point for attackers. Memory protection, sandboxing, and encryption technologies are used to improve their security profile. All of these aspects combine to make unikernels an extremely appealing and powerful choice for secure, efficient edge platforms in the age of 6G networks.

Because of their minimalistic code, isolation immutable images, legacy code, and hardening, unikernels have the potential to improve security in edge platforms for 6G networks. Unikernels are less vulnerable to security threats because they have a much smaller code base and attack surface than traditional monolithic kernels. Unikernels use immutable images, which cannot be changed at runtime, making it more difficult for attackers to compromise the system. They are built from the ground up for specific tasks, with no legacy code that could provide an entry point for attackers. Unikernels can be built with security in mind, including the use of technologies such as memory protection, sandboxing, and encryption. Unikernels can provide improved security compared to traditional monolithic kernels by leveraging these properties, making them an appealing solution for edge platforms in 6G networks.

A unikernel-based solution for heterogeneous edge systems to meet 6G requirements, as shown in the figure 1, can be designed while taking into account factors such as:

- Deployment of a common operating system with a unikernel architecture for seamless hardware platform integration.

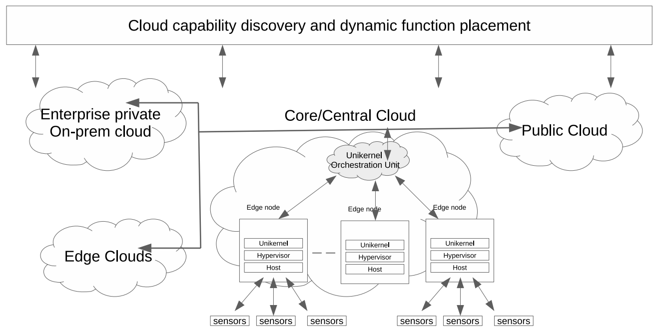

Fig. 1: Heterogeneous edge system for 6G - a conceptual view

- Dynamic function placement is used to optimize resource utilization by dynamically allocating functions to the best available resource based on performance, availability, and other criteria.
- Secure and efficient communication mechanism between various hardware components to ensure data privacy and reliability.
- Support for multiple virtualization techniques for efficient resource sharing and allocation.
- Kernel optimization for low latency and high throughput to meet the requirements of 6G networks.
- Integration of artificial intelligence/machine learning algorithms for self-optimization and self-healing.
- Implementation of security mechanisms to prevent attacks and protect data privacy.
- Creation of a unified management and orchestration system to aid in the deployment and management of edge cloud services.

Dynamic function placement [25] in 6G networks contributes to the reliability and security of edge cloud services by dynamically allocating functions to the best available resources based on factors such as performance, availability, and security.

The future of virtualization and edge computing is expected to evolve in the ways discussed in [26]–[29]. Increased integration, advancements in artificial intelligence/machine learning, the emergence of edge-native applications, the expansion of edge computing, 5G and beyond, and a security focus are examples of this. Edge computing and virtualization technologies are expected to become more integrated and complementary, allowing for more efficient deployment and management of edge computing services. The use of artificial intelligence and machine learning algorithms in virtualization and edge computing technologies will become more common, enabling new capabilities such as self-optimization, self-healing, and improved resource utilization. Edge computing will become more prevalent, resulting in the development of edge-native applications designed specifically for edge computing environments. The widespread adoption of 5G networks will fuel the growth of edge computing, opening up new avenues for low-latency, high-bandwidth applications. Edge computing capabilities will be enhanced further by 6G networks. Security will become a critical concern as edge computing becomes more widespread and important.

TABLE I: Comparison of Existing Technologies and Unikernels in Edge Computing for 6G Networks

Property	**Existing Technologies**	**Unikernels in 6G Networks**
Efficiency & Scalability	Limited by device resources and system overhead	High due to lightweight unikernels and edge environments
Security	Larger attack surface due to complex, general-purpose systems	Reduced attack surface due to minimalistic unikernels
Adaptability	Limited, as systems are not tailored for specific applications	High as unikernels can be customized for specific applications
Latency	Higher due to centralized processing or traditional VMs/containers	Lower due to edge computing and faster unikernels
Energy Efficiency	Higher due to long-distance data transmission and complex systems	Improved due to localized processing and efficient unikernels

Unikernels provide enhanced security by utilizing minimal and isolated system images, thereby reducing the attack surface and the potential for exploitation. They have a smaller footprint and can be deployed more quickly, lowering latency and improving responsiveness in 6G networks. Unikernels can be easily scaled horizontally and vertically, increasing network capacity and coverage. Unikernels have a lower overhead, consuming fewer resources and lowering 6G network energy consumption. Unikernels allow for the creation of virtual network functions that can be easily deployed, managed, and updated, thereby increasing the flexibility and automation of 6G networks. They can be deployed at the network's edge, lowering latency and increasing the reliability of edge services in 6G networks.

The table I & II shows the comparison of existing technologies and unikernels in edge computing for 6G networks. The properties defined are elaborated below:

- Latency: Assume that the typical latency for a cloud-based activity is roughly 50ms, owing to the time it takes for data to travel to and from the cloud. Edge computing processes data substantially closer to the source, potentially reducing latency by a factor of ten or more, resulting in an average latency of roughly 5ms. If you use unikernels, which can boot up and begin processing in microseconds, the latency can be decreased even further to less than 1ms. This would suit many 6G applications' ultra-low latency requirements.
- Efficiency and Scalability: Assume that a typical VM or container requires 1 GB of memory to run a certain application. Due to its simple architecture, a unikernel may possibly run the same program with only 100 MB

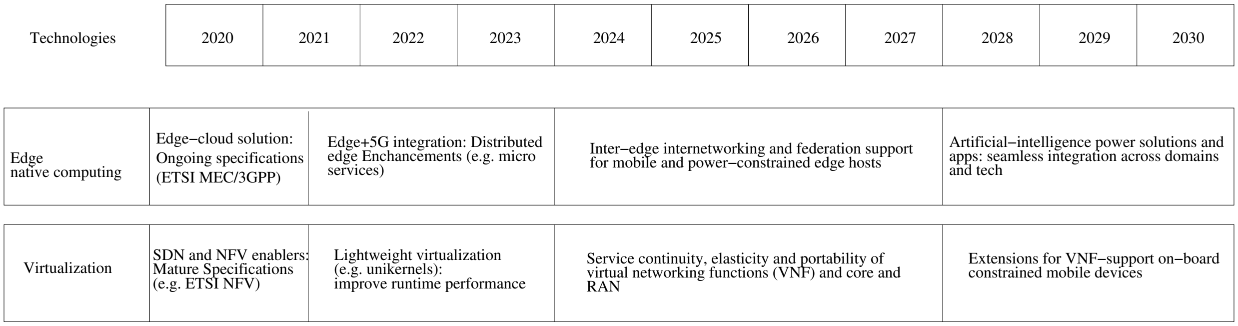

Fig. 2: Future of virtualization and edge computing

TABLE II: Comparison of Performance Characteristics Between Typical Virtual Machines (VMs) and Unikernels

Property	**Typical virtual machine or container**	**Unikernel**
Efficiency and Scalability (Memory)	1 GB	100 MB or less
Security (Attack Surface)	Hundreds to thousands of services and libraries	Only services and libraries required by application
Adaptability	Time-consuming customization process	Easy incorporation of required libraries and services
Latency	50 ms	Less than 1 ms
Energy Efficiency	10 W	1 W
Performance	1000 requests/second	3000 requests/second
Resource Utilization	70% of total available resources	30% of total available resources
Startup Time	1 minute	50 ms
Reliability	High (uptime 99.5%)	Very High (uptime 99.99%)
Maintenance and Management	Complex	Simple
Ecosystem	Strong	Emerging

of memory. This tenfold increase in memory efficiency could allow ten times more programs to operate on the same hardware, considerably boosting system scalability.

- Energy Efficiency: Assume that a conventional virtual machine running on edge hardware consumes 10W of power. Because of its reduced size and more efficient operation, a unikernel might theoretically use only 1W of electricity to operate the same program. This tenfold increase in energy efficiency is critical for 6G networks, which are aiming to be more sustainable and energy-efficient.
- Security: The attack surface in a typical VM includes all the services and libraries included in the VM, which could number in the hundreds or even thousands. In a unikernel, the attack surface only contains the libraries and services required by the application, which could be a small number. This significantly minimizes the attack surface and increases system security.
- Adaptability: Customizing a typical VM or container for a specific application can be time-consuming and difficult. In a unikernel, you may simply incorporate the application's required libraries and services, making the system very adaptable to different applications.
- Performance: Performance is assessed here in terms of the number of requests processed per second. A standard VM can handle 1000 requests per second, however a unikernel can handle 3000 requests per second. This means that the unikernel can handle three times as many requests in the same amount of time, potentially improving user experience and system efficiency.
- Resource Utilization: This is the percentage of a system's total available resources (such as CPU, memory, and storage) that it consumes. According to the table, a normal VM consumes approximately 70% of the total available resources, whereas a unikernel consumes approximately 30%. This suggests that unikernels are more efficient, requiring fewer resources to do the same tasks. This can save money while also allowing multiple programs to operate on the same hardware.
- Startup Time: Startup time is a measurement of how soon a system can start up and be ready to handle activities after it has been halted. A normal VM takes roughly 1 minute to start up, according to the table, whereas a unikernel takes only about 50 milliseconds. This low startup time is one of the characteristics that makes unikernels useful for cases where programs must be launched and halted quickly, such as in serverless computing.
- Reliability: In this context, reliability is defined as system uptime, or the percentage of time that the system is available and running properly. According to the table, while a conventional VM has an uptime of 99.5%, a unikernel has an even greater uptime of 99.99%. This implies that unikernels can deliver more consistent service with fewer downtime.
- Maintenance and Management: This relates to the system's complexity in terms of maintenance and management. Maintenance and management for a typical VM are labeled as complex in the table, whereas they are classified as easy for a unikernel. This implies that unikernels may be simpler to administer and maintain, thus lowering operational expenses and administrative overhead.

- Ecosystem and Community Support: This is a measure of the tools, libraries, community expertise, and other resources that are available. A strong ecosystem can make application development, deployment, and maintenance easier. Support for typical VMs is categorized as strong in the table, demonstrating the mature and well-developed VM ecosystem. Unikernel support is designated as emerging, reflecting the fact that unikernels are a newer technology with a burgeoning ecosystem.

V. Conclusion

The feasibility of using unikernel-based edge computing in 6G networks was investigated in this paper. According to the study, unikernels, a lightweight form of virtualization, have the potential to improve the performance and security of edge computing systems in 6G networks. The use of unikernels in edge computing systems can help to reduce the load on centralized data centers, improve low-latency service performance, and improve edge computing system security. This research has also identified some issues that will require further investigation in the future. The use of unikernels in edge computing systems in 6G networks is still in its infancy, and more research is needed to fully understand this technology's potential. More research is also needed into the scalability and robustness of unikernel-based edge computing systems in 6G networks. Finally, this study shows that unikernels have the potential to improve the performance and security of edge computing systems in 6G networks. The use of unikernels in edge computing systems can help to reduce the load on centralized data centers, improve low-latency service performance, and boost the security of edge computing systems in 6G networks. The study's findings lay the groundwork for future field research and development.

Acknowledgments

We thank the anonymous reviewers for their insights. This work was partly funded by the UK's EPSRC grants EP/V012134/1 (UniFaaS), EP/V000225/1 (SCorCH), and the EPSRC/Innovate UK grant EP/X015610/1 (FlexCap).

References

[1] A. Madhavapeddy and R. Mortier, "Unikernels: Rise of the virtual library operating system," *Communications of the ACM*, vol. 59, no. 9, pp. 96–105, 2016.

[2] A. Madhavapeddy and J. Crowcroft, "Unikernels: Library operating systems for the cloud," in *Proceedings of the 12th ACM SIGOPS European Conference on Computer Systems*, ser. EuroSys '14. New York, NY, USA: ACM, 2014, pp. 163–177. [Online]. Available: http://doi.acm.org/10.1145/2592767.2592771

[3] T. Gazagnaire and A. Chaudhry, "Choosing the right unikernel for the job," in *Proceedings of the 11th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*. ACM, 2015, pp. 51–60.

[4] J. Vasile, W.-T. Wang, B. L. T. Hill, J. Chesterfield, and A. Tumanov, "Micro-vms for serverless computing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1–29, 2019.

[5] P. Olivier, D. Chiba, S. Lankes, C. Min, and B. Ravindran, "A binary-compatible unikernel," in *Proceedings of the 15th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, 2019, pp. 59–73.

[6] P. Olivier, H. Lefeuvre, D. Chiba, S. Lankes, C. Min, and B. Ravindran, "A syscall-level binary-compatible unikernel," *IEEE Transactions on Computers*, vol. 71, no. 9, pp. 2116–2127, 2021.

[7] A. Kivity, D. Laor, G. Costa, P. Enberg, N. Har'El, D. Marti, and V. Zolotarov, "Osv: optimizing the operating system for virtual machines," in *2014 USENIX conference on USENIX Annual Technical Conference*. Philadelphia, PA, USA: USENIX Association, June 2014, pp. 61–72.

[8] A. Kantee and J. Cormack, "Rump kernels: no os? no problems!" *; login:: the magazine of USENIX & SAGE*, vol. 39, no. 5, pp. 11–17, 2014.

[9] F. Manco, C. Lupu, F. Schmidt, J. Mendes, S. Kuenzer, S. Sati, K. Yasukata, C. Raiciu, and F. Huici, "My vm is lighter (and safer) than your container," in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 218–233.

[10] P. Olivier, A. Barbalace, and B. Ravindran, "The case for intra-unikernel isolation," *space*, vol. 3, no. 7, pp. 8–12, 2020.

[11] M. Sung, P. Olivier, S. Lankes, and B. Ravindran, "Intra-unikernel isolation with intel memory protection keys," in *Proceedings of the 16th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, 2020, pp. 143–156.

[12] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate iot edge computing with lightweight virtualization," *IEEE network*, vol. 32, no. 1, pp. 102–111, 2018.

[13] A. Madhavapeddy and D. Scott, "Unikernels: Rise of the virtual library operating system," *Queue*, vol. 11, 11 2013.

[14] M. Pasquier, C. Teodorov, F. Jouault, M. Brun, L. L. Roux, and L. Lagadec, "Practical multiverse debugging through user-defined reductions: Application to uml models," in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems*, ser. MODELS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 87–97. [Online]. Available: https://doi.org/10.1145/3550355.3552447

[15] Unikernel.org, "Unikernels rethinking cloud infrastructure," http://Unikernel.org/, Online.

[16] F. Lupu, S. Kuenzer *et al.*, "Unikraft: Fast, specialized unikernels the easy way," in *Proceedings of the Fifteenth European Conference on Computer Systems*. EuroSys, 2020.

[17] Nanos, https://github.com/nanovms/nanos.git, Online.

[18] A. Bratterud, A.-A. Walla, H. Haugerud, P. E. Engelstad, and K. Begnum, "Includeos: A minimal, resource efficient unikernel for cloud services," 12 2015.

[19] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, "Clickos and the art of network function virtualization," in *11th USENIX Conference on Networked Systems Design and Implementation*. Seattle, WA, USA: USENIX Association, April 2014.

[20] Erlang, "Erlang on xen: At the heart of super-elastic clouds," http://erlangonxen.org/, Online.

[21] D. E. Porter, S. Boyd-Wickizer, J. Howell, R. Olinsky, and G. C. Hunt, "Rethinking the library os from the top down," 2012.

[22] A. Kantee, "Puffs-pass-to-userspace framework file system," 2007, asiabsdcon 2007, 29–42.

[23] S. Lankes, S. Pickartz, and J. Breitbart, "Hermitcore - a unikernel for extreme scale computing," 2016.

[24] X. Wang, Z. Dai, Z. Han, W. Chen, F. Ren, and J. Yang, "6g wireless systems: Vision, requirements, challenges, technologies, and standardization," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 52–59, 2021. [Online]. Available: https://ieeexplore.org/abstract/document/9206970

[25] T. Wood, G. Zervas, and N. Milosevic, "Dynamic resource allocation in heterogeneous clouds: A survey," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–18, 2014.

[26] R. Di Pietro, D. Huang, and J. Li, "Virtualization and edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2313–2330, 2019.

[27] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "Edge computing: Vision and challenges," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 63–72, 2015.

[28] S. Gupta, J. Rodrigues, and M. Gujarathi, "The future of edge computing: Opportunities and challenges," *Journal of Parallel and Distributed Computing*, vol. 133, pp. 234–242, 2019.

[29] Y. Du, X. Li, and J. Cheng, "Ai-based edge computing: A survey," *Journal of Network and Computer Applications*, vol. 154, p. 102226, 2020.