Cabus, Clement Harold Miguel

IV-ACSAD

Narrative Report: Virtualization vs. Containerization in Edge Computing

The rapid digital transformation across various sectors has created an urgent need to optimize infrastructure for software application deployment, particularly in edge computing environments where data processing occurs closer to the source rather than in centralized cloud servers. The article "Virtualization vs. Containerization, a Comparative Approach for Application Deployment in the Computing Continuum Focused on the Edge" by Sturley et al. (2024) addresses this critical challenge by conducting a comprehensive comparative analysis of two prevalent deployment technologies. Virtual machines (VMs) have long been the established method for creating complete, autonomous systems by abstracting the hardware layer, allowing multiple operating systems to run concurrently on a single physical machine. In contrast, containerization emerged approximately ten years ago as a more lightweight alternative, isolating applications within environments that share the host operating system's kernel, leading to faster deployment times and more efficient resource utilization. As organizations increasingly adopt technologies such as e-Government, Industry 4.0, Agriculture 5.0, and Smart Cities, the choice between virtualization and containerization becomes increasingly critical, particularly in resource-constrained edge environments where Quality of Service requirements, power consumption, and performance must be carefully balanced. This study aims to determine the optimal deployment strategy by evaluating various virtualization and containerization stacks across key factors including performance, scalability, energy efficiency, and ease of management, with a particular focus on edge computing applications.

The research methodology employed by Sturley et al. demonstrates remarkable thoroughness in testing multiple hardware platforms and software configurations to ensure comprehensive and reliable results. The study utilized two primary testbeds: an ARM-based Raspberry Pi 4B+ platform selected to represent the growing prevalence of ARM processors in edge computing infrastructure, and an x86-based platform implemented using the Proxmox Virtual Environment hypervisor. The researchers evaluated multiple containerization stacks including Docker Engine with Docker Compose orchestration, Podman container runtime, and Kubernetes with both Docker Engine and Podman backends, alongside various virtual machine configurations using QEMU virtualization for both x86 and ARM64 architectures. To measure performance, the team employed three primary tools: Stress-ng for applying controlled workloads to system components, the Top system monitor for capturing real-time performance data, and the UM24C power meter data logger for precise power consumption measurements. The testing approach included both short-duration tests of one minute and extended tests of ten to twenty minutes, allowing the researchers to observe both immediate performance characteristics and long-term efficiency trends across different deployment scenarios.

The performance testing results revealed significant differences between deployment strategies, with particularly notable findings in CPU performance, I/O operations, and power efficiency. In short-duration CPU tests lasting one minute, Docker Compose with Podman demonstrated the best performance, achieving approximately 5% better results than Kubernetes implementations with either runtime backend. However, as test duration extended to ten minutes, the performance gap between Kubernetes and other solutions narrowed considerably, suggesting that Kubernetes introduces higher initial overhead but demonstrates improving relative efficiency over time, making it more suitable for long-term service deployments. The disk I/O tests produced even more striking results, with Docker Compose consistently outperforming Kubernetes by approximately 40% in short-duration tests, an advantage attributed to Podman's daemon-less architecture which eliminates overhead from complex network services, while Kubernetes' architectural complexity requiring components like Kubelet, API server, and controller manager introduced significant overhead impacting I/O operations. Memory utilization showed less dramatic differences between solutions, though Kubernetes still exhibited slightly higher resource consumption due to its additional management components. Perhaps most critically for edge computing applications, the power efficiency measurements revealed that native Raspberry Pi OS, Docker containers, and ARM64 VMs with KVM extensions showed remarkably similar power efficiency, demonstrating the effectiveness of containerization while providing deployment flexibility, whereas Kubernetes initially showed lower power efficiency but significantly improved in extended twenty-minute tests, eventually exceeding Docker's efficiency for sustained workloads. The study also uncovered a crucial finding regarding cross-architecture compatibility: x86 emulation on ARM platforms proved extremely power-inefficient with dramatically reduced performance, highlighting the importance of instruction set architecture compatibility in edge computing deployments and revealing that without full CPU parity, no virtual acceleration was occurring—a potential area for future optimization.

To demonstrate real-world applicability of their findings, the researchers implemented a practical use case involving dynamic pointcloud generation using OpenDroneMap (ODM), a computationally intensive task that showcases how containerization and orchestration technologies handle distributed processing. The system architecture utilized a three-tier approach with drone image capture and transmission to a Raspberry Pi for preprocessing, an edge server providing GPU-accelerated transformation matrix computation, and a cloud environment where Kubernetes orchestrated multiple ODM instances in Docker containers to distribute the computational load efficiently. The experiment processed forty-eight drone images using four ODM nodes, each equipped with four virtual CPUs, achieving an average preprocessing time of 3.67 seconds per image and transformation matrix creation time of 24.72 seconds, while parallel pointcloud generation for different spectral bands (RGB, Red Edge, Red, and Green) completed in approximately 850 to 920 seconds depending on the band. This parallel processing approach significantly accelerated the workflow compared to sequential processing, demonstrating the practical effectiveness of Kubernetes orchestration for distributing computational tasks across multiple nodes in production environments. The use case validates the study's findings by showing that even the theoretically less efficient Kubernetes and Docker Engine stack—which served as the baseline in percentage comparison tests—can deliver meaningful performance for real-world applications requiring intensive processing, thereby confirming that the choice between deployment technologies should be guided by specific use case requirements rather than absolute performance metrics alone.

Based on their comprehensive testing results, the researchers provide clear, evidence-based recommendations for different deployment scenarios that help organizations make informed technology choices. Docker Compose with Podman emerges as the optimal solution for resource-constrained edge environments, one-time or short-duration computational tasks, applications requiring maximum resource efficiency, and simple architectures without complex scaling needs, offering distinct advantages including 5% better CPU performance in typical scenarios, 40% better I/O disk performance, lower memory overhead, and minimal configuration requirements due to its daemon-less architecture. Conversely, Kubernetes demonstrates clear superiority for long-running services and applications, complex architectures requiring redundancy, applications needing automatic scaling, production environments with multiple nodes, and systems requiring sophisticated error management and fault tolerance, with particular strengths in better long-term efficiency for sustained workloads, an extensive ecosystem of compatible tools and containers, active community support, advanced load balancing and orchestration features, and superior power efficiency in extended operations lasting twenty minutes or longer. Virtual machines retain their essential role for applications requiring complete hardware isolation, systems needing custom low-level configurations, scenarios requiring graphical user interfaces for monitoring, use cases involving USB device passthrough or specialized hardware, and environments where complete operating system autonomy is necessary, though they come with tradeoffs including higher resource overhead and slower deployment times compared to containers. The researchers emphasize that the similarity in power consumption between properly configured VMs and containers provides organizations with deployment flexibility, allowing technology choices to be based on functional requirements rather than efficiency concerns alone, while noting that the ongoing transition in edge computing infrastructure toward ARM-based processors highlights the growing importance of power efficiency alongside computational capability.

The comprehensive comparative study by Sturley et al. provides valuable, evidence-based insights for organizations navigating the complex landscape of edge computing deployment strategies, ultimately concluding that containerization offers the most ecologically advantageous option in terms of energy consumption and resource efficiency for typical edge computing scenarios. The research demonstrates that Docker Compose with Podman excels in short to moderate-duration tasks requiring maximum resource efficiency and simplicity, while Kubernetes demonstrates superior capabilities for complex orchestration and long-term services despite higher initial overhead, and virtual machines remain the appropriate choice for scenarios demanding complete isolation and hardware control despite higher resource costs. The study's practical demonstration using pointcloud generation effectively illustrates how these technologies can be combined in real-world applications, with edge preprocessing, cloud orchestration, and parallel processing working together to achieve optimal results in computationally intensive tasks. As edge computing continues to evolve and ARM architectures gain prominence in distributed computing environments, the ability to make informed decisions about deployment strategies based on specific use case requirements—rather than pursuing a one-size-fits-all approach—will be crucial for organizations seeking to leverage these technologies effectively while meeting sustainability goals and operational efficiency targets. The researchers acknowledge limitations including the use of non-standardized Bogo Operations metrics and focus on computational workloads, while identifying important areas for future investigation including extended use case analysis across broader application types, scalability studies in larger hardware architectures, comprehensive security evaluation in edge environments, and optimization research for cross-architecture virtualization acceleration. Ultimately, the findings confirm that while containerization represents a significant advancement in deployment technology offering compelling advantages for many scenarios, both containers and virtual machines have essential, complementary roles to play in modern computing infrastructure, and the key to successful deployment lies in understanding the strengths, limitations, and appropriate use cases for each approach to optimize performance, efficiency, and sustainability across the computing continuum from cloud to edge environments.