Sicat, Sean Russel

IV – ACSAD

Narrative Report

Introduction

The authors begin by situating their study in the evolving computing landscape: as digital transformation accelerates across domains such as e-Government, Industry 4.0, Agriculture 5.0 and Smart Cities, the underlying infrastructure must keep pace. They focus particularly on the "Computing Continuum", the continuum spanning from centralized Cloud through Fog to the Edge. In this continuum, application deployment becomes more challenging because resources nearer the edge tend to be more constrained (power, CPU, memory, network) and latency, autonomy and locality become more significant.

Against this backdrop, the authors ask: given two major methods of virtualized deployment, full virtual machines (VMs) and containers, which is more appropriate for application deployment across the continuum, and especially at the edge? They indicate that while containerization has grown in use over the past decade, there remains a need to evaluate the trade-offs between containers and VMs in this specific edge/continuum context rather than only in large cloud data-centers.

Literature Review

The paper reviews previous studies on virtualization and containerization, emphasizing that container-based virtualization has been widely explored in real-time and industrial IoT environments, with numerous papers addressing real-time constraints and container platforms. It also highlights existing systematic reviews comparing hypervisor-based virtual machines and containers, as well as studies on orchestration and monitoring tools. However, most of these works concentrate on cloud deployments such as IaaS, PaaS, and SaaS, leaving the edge, fog, and continuum environments less examined. Some research indicates that containers maintain good performance in edge-cloud scenarios but still face challenges in time-critical industrial applications. Recognizing this gap, the authors position their study as an empirical comparison of containers and virtual machines on edge-type hardware, including ARM architectures, with a particular focus on energy efficiency in constrained environments.

Materials and Methods

The authors describe in detail their methodology:

Hardware & software testbed

- They used a mixture of host systems including a Raspberry Pi 4B+ (as a low-power edge device) and x86/ARM virtual machine setups.
- They used the benchmarking tool stress-ng to create reproducible workloads stressing CPU, memory, I/O.
- They used the Linux "top" utility to monitor resource usage over time.

Virtualization/Containerization stacks

They built:

- Virtual machine stacks: Using QEMU/KVM for x86_64 and aarch64 (ARM) virtual machines (e.g., Debian 12 guest OS) on appropriate hosts.
- Container stacks: They created Docker and Kubernetes/Podman environments on Debian 12 containers. For container management: Docker Compose (single node) and Kubernetes cluster (single node) were used.

Test scenarios

They ran two principal types of tests:

- Short workloads (1 minute) and longer workloads (10+ minutes) measuring operations ("Bogo Ops" from stress-ng) per second and CPU user/sys time.
- A real-world use case: drone-image capture and point-cloud generation (with spectral bands) running on multiple worker nodes in a cluster, to test scalability in a heterogeneous scenario.

Metrics

They compared:

- Performance: Bogo Ops per second, CPU user + system time, I/O throughput.
- Scalability: how the system scaled under added load / multiple instances (in the drone use case).
- Energy / efficiency: particularly in the edge scenario (ARM) — watt-to-operation requirements.

Results

Performance results

From the benchmarking, they found:

- For the Raspberry Pi / ARM host, container performance (Docker, Podman) and ARM VM performance were very similar, indicating that containers are highly efficient in resource-constrained devices.
- The x86 VM (emulated/hosted) showed significantly poorer performance — especially when emulating foreign architectures — indicating virtualization overheads become more pronounced in certain configurations.
- I/O performance: For disk I/O over 1 minute, Docker Compose (lighter container management) was about 40% more efficient than Kubernetes + Docker Engine in their scenario.
- Memory usage: Differences between stacks in memory were less significant compared to CPU or I/O differences.

Use-case (drone + point-cloud) results

In the more realistic workload: using four worker nodes each with four vCPUs, processing four spectral bands in parallel:

- Average preprocessing time per image: ~3.67 seconds
- Warp matrix creation: ~24.72 seconds
- Point-cloud generation times: ~912.28 s for RGB, ~868.09 s for RedEdge, ~875.95 s for Red, ~858.71 s for Green.
  These results demonstrate that the distributed/parallel deployment (using containers/orchestrated nodes) provides effective scalability for a non-trivial workload.

Energy and Power Efficiency

They also evaluated power/energy implications: In edge devices (ARM), the watt-to-operation requirements show a need to optimise both VM and container stacks. However, they found containers tend to have an ecological advantage (less overhead) and thus better energy efficiency in many cases.

Discussion

The authors found that containers offer a lighter virtualization layer by sharing the host OS kernel, reducing overhead and improving CPU and I/O efficiency—especially useful in edge environments with limited resources. Virtual machines, however, remain valuable for workloads needing full OS isolation, specialized hardware, or legacy support.

In the edge and computing continuum, where devices have limited power and diverse architectures, containers are generally more efficient, though the choice between containers and VMs depends on the application's needs. Orchestration complexity also matters: simpler tools like Docker Compose or Podman perform better for small-scale setups, while Kubernetes suits larger deployments despite its overhead. The authors note that containers are more energy-efficient and environmentally friendly, and recommend further research on security, additional architectures, and large-scale tests.

Conclusion

The study concludes that containers are more suitable for most edge and continuum deployments due to their resource efficiency, scalability, and energy advantages. Virtual machines still play an important role where strong isolation or hardware-specific requirements exist. Ultimately, the best choice depends on the workload, environment, and management complexity. The authors emphasize that containerization is the most ecologically advantageous approach tested.

Reflection

The study applies solid methods, combining benchmarks and real-world testing, but its hardware scale is limited. Broader experiments could confirm the findings. While energy efficiency was analyzed, long-term factors such as thermal effects and reliability were not. Security and orchestration trade-offs also need deeper study. Finally, because containers share the host kernel, future work should examine isolation and security to ensure safe deployment alongside performance.

Reference

Sturley, H., Fournier, A., Salcedo-Navarro, A., Garcia-Pineda, M., & Segura-Garcia, J. (2024). Virtualization vs. Containerization, a Comparative Approach for Application Deployment in the Computing Continuum Focused on the Edge. *Future Internet*, *16*(11), 427. https://doi.org/10.3390/fi16110427