JEREMIAH B. DELA VEGA
IV-BCSAD
Assignment# 4

NARRATIVE REPORT

Executive Summary

This report summarizes the findings of a research paper titled "Virtualization vs. Containerization, a Comparative Approach for Application Deployment in the Computing Continuum Focused on the Edge." The study conducts a comprehensive performance and efficiency analysis to determine the optimal deployment strategy (Virtual Machines vs. Containers) for modern applications, particularly in resource-constrained Edge Computing environments. The key finding is that while both technologies have valid use cases, containerization is the "most ecologically advantageous option" due to its superior energy efficiency.

1. Research Objective

The central goal of the paper was to compare two primary application deployment methods:

- **Virtualization (VMs):** A traditional approach that emulates an entire computer, including a full guest operating system (OS).
- **Containerization (e.g., Docker, Kubernetes):** A lightweight approach that packages an application and its dependencies, sharing the host OS kernel.

The researchers aimed to move beyond theoretical differences and quantify the practical impact of each method on **performance, resource utilization, and energy consumption**, with a specific focus on their suitability for Edge Computing.

2. Methodology

To conduct a fair comparison, the researchers established a controlled testing environment:

- **Hardware Testbeds:** The primary testbed was a **Raspberry Pi 4B+**, chosen to represent a typical ARM-based Edge device. Comparative tests were also run on a standard x86 (Intel/AMD) host.

- **Stress Testing:** The **stress-ng** tool was used to apply consistent and measurable workloads, targeting the CPU, system memory, and I/O (disk) operations.
- **Power Measurement:** Crucially, the team used a **physical power meter data logger (UM24C)** to measure the actual wattage consumed by the hardware during each test, providing a clear metric for energy efficiency.
- **Configurations Tested:** The study was exhaustive, testing multiple combinations, including:
    - Docker containers
    - Kubernetes (minikube) with a Podman runtime
    - Native ARM Virtual Machines (using KVM)
    - Emulated x86 Virtual Machines (using QEMU)

3. Key Findings

The experimental results revealed several significant trends:

- **Performance:**
    - **Emulation is Inefficient:** Emulating an x86 VM on the ARM-based Raspberry Pi resulted in "extremely poor performance," confirming it as an impractical solution for the edge.
    - **Containers vs. Native VMs:** The performance difference between a lightweight Docker container and a properly configured (native) ARM virtual machine was **surprisingly small**. This indicates that modern virtualization with the correct extensions is highly efficient.
- **Orchestration Overhead:**
    - The choice of "manager" (orchestrator) had a significant performance impact. For **short-term, simple tasks**, lightweight solutions like Docker Compose were the most efficient.
    - **Kubernetes**, a complex orchestrator, demonstrated significant initial "overhead," consuming more resources just to run itself, which made it less efficient on short tests.
    - However, on **longer-duration tests (20 minutes)**, Kubernetes's efficiency improved, suggesting its sophisticated management capabilities become more beneficial for sustained, long-running services.
- **Energy Efficiency:**
    - This was the study's most definitive finding. Across the tests, container-based solutions consistently consumed less power to perform the same amount of work. The paper concludes that **containerization offers the most ecologically advantageous** approach, a critical factor for battery-powered or low-power edge devices.

4. Practical Use Case Validation

To validate these findings, the researchers applied their setup to a real-world scenario: **processing drone imagery to generate 3D point clouds** using OpenDroneMap (ODM). They demonstrated that even one of the "worst-performing" test configurations (Kubernetes + Docker Engine) was still fully capable of managing and completing this complex, data-intensive workload, proving the viability of these technologies for demanding edge tasks.

5. Conclusion and Recommendations

The paper concludes that there is **no single "winner"** for all scenarios; the choice depends on the specific requirements of the project.

- **Virtual Machines (VMs)** remain the necessary choice when **strict hardware-level isolation** is a security requirement or when an application needs direct access to physical hardware (like a USB device).
- **Containers** are the recommended choice for most modern applications, especially on the edge. They are **lightweight, flexible, faster to deploy, and demonstrably more energy-efficient**.