





Article

Implementing Virtualization on Single-Board Computers: A Case Study on Edge Computing

Georgios Lambropoulos ^{1,*}, Sarandis Mitropoulos ^{2,†}, Christos Douligeris ^{1,†} and Leandros Maglaras ^{3,*}

¹ Department of Informatics, University of Piraeus, 18534 Piraeus, Greece; cdoulig@unipi.gr

² Regional Development Department, Ionian University, 31100 Lefkada, Greece; smitropoulos@ionio.gr

³ School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK

* Correspondence: glampropoulos@unipi.gr (G.L.); l.maglaras@napier.ac.uk (L.M.)

† These authors contributed equally to this work.

Abstract: The widespread adoption of cloud computing has resulted in centralized datacenter structures; however, there is a requirement for smaller-scale distributed infrastructures to meet the demands for speed, responsiveness, and security for critical applications. Single-Board Computers (SBCs) present numerous advantages such as low power consumption, low cost, minimal heat emission, and high processing power, making them suitable for applications such as the Internet of Things (IoT), experimentation, and other advanced projects. This paper investigates the possibility of adopting virtualization technology on Single-Board Computers (SBCs) for the implementation of reliable and cost-efficient edge-computing environments. The results of this study are based on experimental implementations and testing conducted in the course of a case study performed on the edge infrastructure of a financial organization, where workload migration was achieved from a traditional to an SBC-based edge infrastructure. The performance of the two infrastructures was studied and compared during this process, providing important insights into the power efficiency gains, resource utilization, and overall suitability for the organization's operational needs.

Keywords: SBC; virtualization; edge computing; Raspberry Pi



Citation: Lambropoulos, G.; Mitropoulos, S.; Douligeris, C.; Maglaras, L. Implementing Virtualization on Single-Board Computers: A Case Study on Edge Computing. *Computers* **2024**, *13*, 54. <https://doi.org/10.3390/computers13020054>

Academic Editors: Paolo Bellavista, Sheng-Joue Young, Shouou-Jinn Chang, Liang-Wen Ji and Stephen D. Prior

Received: 28 December 2023

Revised: 10 February 2024

Accepted: 15 February 2024

Published: 18 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid adoption of cloud computing by most modern corporations leads to centralized and consolidated datacenter structures. Nevertheless, in both public and private implementations, cloud computing may not always meet the necessary requirements in terms of speed, responsiveness, and security to cover the needs of several critical applications. To address the above shortcomings, the implementation of smaller-scale distributed infrastructures at the edges of corporate networks and specifically near endpoints that feature intense data transactions is recommended. This practice is often referred to as edge computing. The edge-computing model features distributed micro datacenter infrastructures closer to the data generation sites to allow faster networking response, local data storage, and enhanced security. Specifically, by creating decentralized datacenters near the data creation source, edge computing reduces exposure concerns since data processing takes place on-premise by utilizing local resources, thus minimizing the potential attack risks that arise by the continuous data transmission to remote infrastructures. Furthermore, edge computing facilitates the adoption of traditional security policies and tools that cannot otherwise be implemented in complex cloud-oriented environments [1].

Despite the advantages of edge computing, there are a few concerns that are mostly due to the servicing needs, power consumption and remote administration of the infrastructures that need to be implemented. Especially in cases of small office branches or shop-in-a-shop scenarios, a dedicated and controlled environment for hosting sensitive hardware equipment is very difficult to allocate. Power consumption and air conditioning

needs are also limiting factors. A possible solution that addresses these concerns is the usage of Single-Board Computers (SBCs).

Over the last decade, SBCs have become increasingly relevant due to their low power consumption, low purchasing cost and minimal heat generation. Additionally, the rapid development of power-efficient processors, mostly based on the Aarch64 (ARM64) architecture, makes SBCs ideal for numerous applications such as Internet of Things (IoT), experimentation, prototyping and robotics. The increased demand for more powerful and scalable SBC platforms drives hardware manufacturing companies to produce several different boards either for general-purpose development or optimized for specific tasks (i.e., sensor control, image processing and data analytics) [2]. In the same context, modern SBCs also feature powerful specifications, such as more physical memory (RAM), and are equipped with faster embedded hardware, such as USB3 ports, gigabit Ethernet controllers, Bluetooth radios and Wi-Fi adapters. Indicative examples of such SBCs are Raspberry Pi (by Raspberry Foundation), NVIDIA Jetson (by NVIDIA Corporation), Layerscape Design Board (by NXP Semiconductors) and Quartz64 (by Pine64).

Even though SBCs seem to be a viable and appealing option for edge computing, it is essential to take into account a number of important factors in order to implement reliable, expandable and efficient infrastructures. Specifically, one of the most important prerequisites is that these edge infrastructures shall feature enterprise-level functionalities, such as flexible administration, failover clustering capabilities, and disaster recovery tools. Additionally, all hosted services should be hardware-independent and easily migratable among different types of hosts. Based on the above facts, the underlying technology on which these infrastructures should be based on is virtualization.

This paper aims to investigate the possibility of adopting virtualization technology on Single-Board Computers (SBCs) for the implementation of reliable and cost-efficient edge-computing environments. The scope of this investigation is to study the current technological advances and capabilities in both hardware and software bases in order to examine the viability of such implementations.

The structure of the paper is as follows. Section 2 provides an overview of works in the field of Single-Board Computer, edge computing and virtualization technology implementations along with relevant studies that combine these technologies. Section 3 presents the overall research structure and roadmap, featuring a detailed analysis of the phases and the involved steps followed. Section 4 provides a description of the experimentation process concerning the installation of type-1 hypervisors on the selected SBC along with the hardware environment analysis and limitations. Section 5 features the testing environment preparation and the hypervisor platforms testing, and identifies the possible technical limitations involved. Section 6 presents a case study for replacing a traditional edge-computing infrastructure of a financial organization with an SBC base. The performance testing and results analysis are provided. Section 7 presents the conclusions and proposes future work.

2. Related Work

This section provides an overview of works in the field of Single-Board Computer, edge computing, serverless computing and virtualization technology implementations along with relevant studies that combine these technologies. This analysis is crucial in order to define the enabling technologies that could be used in edge SBC computing virtualized scenarios.

The idea of employing a small, reasonably priced, linked computer in various scientific and educational setups was made more popular by the founding of the Raspberry Pi foundation, a nonprofit organization promoting the educational value of its devices. Single-Board Computer research is mainly focused on studying their employment in sectors such as science, engineering and education [3,4], the implementation of Software-Defined Radio (SDR) systems [5], as well as their usage for creating clustered computing environments that leverage their cost efficiency compared to traditional computer systems [6]. Other works study their energy efficiency on edge-computing implementations [7] and their

ability to integrate sensor technologies for specific IoT applications [8]. It should be noted that Single-Board Computers have both benefits and drawbacks. On the one hand, vendors can speed up time to market by needing less development time, and a wide range of sizes, functions, and prices are offered by several providers. However, they are not always economically viable for high volumes of computation or data.

As far as edge computing is concerned, the relevant research is mainly focused on the enhancement of cloud provided services due to the incremental growth of utilization and connected devices mostly in the field of IoT [9]. Researchers have identified key areas such as network performance, availability, power consumption and security, where edge computing may considerably contribute [10]. International Data Corporation (IDC) in cooperation with VMware, identifies edge computing as the next step for the transformation and evolution of the cloud industry [11]. Investments on edge computing are expected to increase mainly in the fields of customer service, transportation, tourism and logistics [12]. This is further validated by a forecast by IDC Corporation that predicts an average of USD 176 billion on edge-computing investments by the end of 2022. The same forecast predicts that total investments on edge computing are expected to reach USD 274 billion by the end of 2025. These investments include hardware, software and service procurement costs [13].

Virtualization technology has been employed for more than one decade in most enterprise datacenter implementations. Specifically, virtualization features a variety of benefits, such as significant cost reduction, higher performance, and availability as well as easier maintenance and administrative flexibility [14]. Additionally, virtualization facilitates the deployment and migration of applications while ensuring high availability for operational and application areas. Particularly in terms of energy efficiency and the lowering of an organization's CO₂ footprint, virtualization is an excellent technique for minimizing the environmental effect of datacenters. Additionally, it aids in enhancing flexibility and decreasing maintenance expenses [15]. As compared to traditional virtualization solutions (VMware, KVM), Docker is a high-level container engine technology that is based on LXC (Linux Container), the widely used method for virtualization processes. Lightweight virtualization for resource and process separation is provided by the kernel virtualization technology LXC. Docker containers are the mainstream solution in the current virtualization field [16].

With the massive use of edge computing, new possibilities have arisen for IoT and IIoT. These come along with new problems related to storage and computing power. Efficient resource utilization became an urgent need, and virtualization technology came to partially solve this issue. It can solve these issues but at the cost of duplicate resource configuration and provision delays in some instances [17]. To overcome these problems, a new model called serverless computing has recently been introduced [18,19]. Serverless computing can autoscale the service offered following the customers' demand and also charge the customers fairly only for the service offered, independently of the underlying infrastructure [20]. Moreover, other scholars focused on solving resource allocation problems through the use of optimization methods [21]. Finally, distributed intelligence sharing is handled efficiently in [22]. The latter method can be the solution to the overfitting of learning algorithms that work in edge environments where the data samples can be limited.

Based on the analysis of the related work, it is evident that the technology has progressed to such a state where the transition to Single-Board Computers could be feasible for some applications and processes. This study looks at the idea of using Single-Board Computers (SBCs) with virtualization technologies to develop secure and economical edge-computing environments. The goal of this analysis is to investigate the plausibility of such implementations both now and in the near future by studying current hardware and software technology advancements and capabilities.

3. Research Structure and Roadmap

This section provides an overview of the research structure and roadmap followed in this paper. Specifically, the scope of work for this research is to provide a comprehensive evaluation of virtualization on Single-Board Computers (SBCs) for edge-computing purposes. It is important to note that all tests, both hardware- and software-related, may utilize trial, unsupported, and testing versions of various tools, operating systems and applications. The primary objective is to demonstrate and assess the current technological advancements in the integration of virtualization technology on SBCs, without targeting solutions with long-term support or final software versions at this stage. The focus is on exploring the capabilities and potential of SBCs as virtualization hosts for edge-computing scenarios, understanding the limitations and challenges that may arise, and identifying possible remedies. By using a variety of trial and testing versions, this research aims to gain insights into the feasibility and performance of SBCs in virtualized environments, providing an overview for future optimizations and enhancements. The research flow, which is structured in three main phases, is presented in Figure 1:

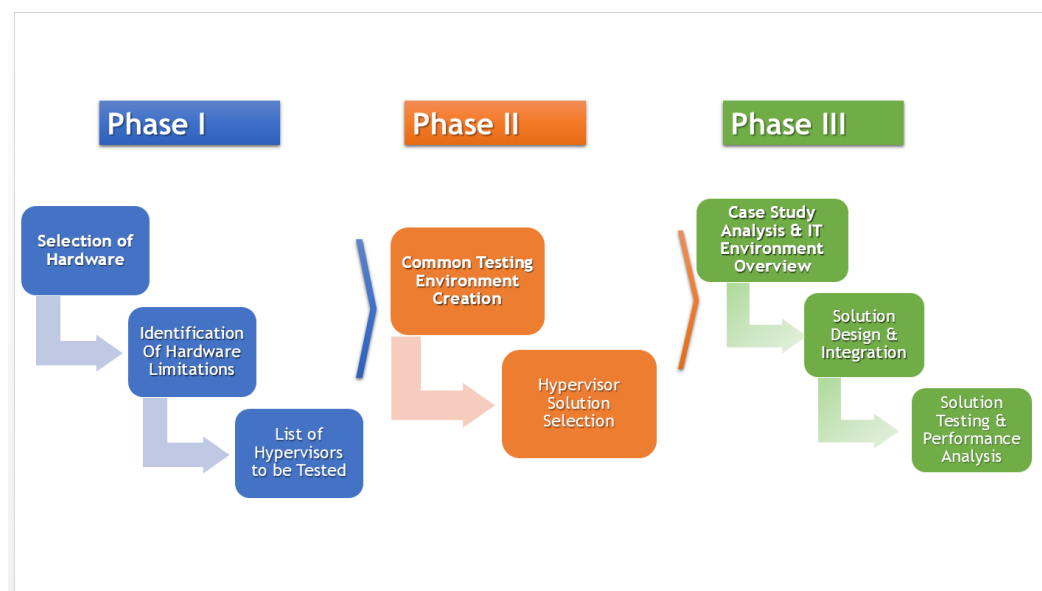


Figure 1. Research phases and flow.

In the first phase, we present the process of hardware selection and analysis and the identification of hardware limitations, and conclude by defining the hypervisors to be tested.

In the second phase, we describe the establishment of a common hardware testing environment to ensure a consistent evaluation of the selected hypervisors. The testing environment creation should be reproducible and should remain constant throughout the entire testing process. The final step of the second phase is concerned with the results of the hypervisors' distributions testing and the final hypervisor selection, which shall be used afterwards during the third phase.

In the third and final phase, we present a case study that took place in a real-world production environment of a financial organization. Specifically, the first step of this phase is the presentation an analysis of the existing IT environment in order to define the provided IT services and integration requirements to lead to the definition of the study. In the second step, we analyze the design, implementation and integration of the examined solution, and the third and final step describes the solution-testing process and performance analysis, and delivers the conclusions of the case study.

4. Phase 1: Hardware Selection and Testing Process

This section presents and analyzes all the necessary steps involved during the phase of the SBC selection, the identification of its hardware features, and the limitations that derive from its architectural design. Specifically, this section highlights features such as memory capacity and processing power, taking into account factors like network connectivity, compatibility with different operating systems, and the potential for deploying virtualization technology. Moreover, this section presents the systematic steps for analyzing the hardware environment by considering aspects such as storage, firmware installation and power supply considerations. Additionally, noteworthy limitations such as performance constraints, reliability issues and the absence of essential hardware components are identified and analyzed.

4.1. Hardware Selection and Technical Specifications

For the purposes of this research, the Raspberry Pi 4B was chosen as the subject of investigation due to its hardware attributes and features. The first and most important consideration for selecting the Raspberry Pi 4B is that it features an edition equipped with 8GB LPDDR4-3200 SDRAM. At the time this research was conducted, other popular SBCs, such as Nvidia Jetson, Quartz64, and Layerscape Design Board, were equipped with lower RAM configurations and specifically with a maximum of 4 GB or less. By having more RAM available, the Raspberry Pi 4B is able to handle more demanding application workloads, providing greater scalability. More importantly, since this research focuses on virtualization technology, RAM is a determining factor, mostly regarding the amount and load of virtual machines (VMs) that it may facilitate.

Additionally, Raspberry Pi 4B embeds a Gigabit Network adapter, enabling it to integrate with existing networking infrastructures and a dual USB3.0 controller, allowing the interconnection of hi-speed peripherals, such as external storage devices. Raspberry Pi 4B supports a variety of operating systems, such as Raspberry Pi OS, Ubuntu, FreeBSD, Microsoft Windows and other Linux distributions [23], providing the ability to select among multiple software environments for testing and experimentation. The technical specifications of Raspberry PI 4B are summarized in Table 1:

Table 1. Raspberry Pi 4B specifications.

Category	Specifications
Processor	Broadcom BCM2711 Quad-core SoC ARM Cortex-A72 (ARM v8), 1.5GHz [24]
Network Connectivity	Gigabit Ethernet (Microchip LAN7515) [25] Dual-band Wi-Fi (Cypress CYW43455) [26] Bluetooth 5.0 with BLE [27] Four USB ports (2 USB2, 2 USB3.0) [28]
Form Factor	Compact (85.6mm × 56.5mm)
Input Power	5V DC-3A

4.2. Hardware Environment Analysis and Limitations

Before proceeding with the installation process and the testing of the hypervisor platforms, it is essential to further explain a number of preparatory steps concerning the Raspberry Pi 4B hardware environment. These steps include the interfacing of the main storage unit, the firmware and booting configuration, the power supply selection and thermal protection. During the analysis of these steps, hardware limitations that may impact the overall reliability and performance of the system are identified and addressed.

4.2.1. Storage Devices

The Raspberry Pi 4B is equipped with a micro SD card slot, which serves as a default primary storage device. However, micro SD cards are not reliable or fast enough to be used

as the primary storage for a server operating system and therefore, not recommended for hosting virtual machines. Another significant concern is that the Raspberry Pi 4B does not embed an onboard storage controller, such as SAS or SATA, nor does it feature an expansion bus such as a PCI-e interface for installing an external one. Because of these constraints, the best available option for installing an external hard disk that performs adequately and reliably is to utilize the provided 2-port USB3.0 controller. To connect a SATA3 hard drive, a third-party SATA3 to USB3 adapter is required. Nevertheless, the USB3.0 bus supports a maximum theoretical transfer speed of up to 5 Gbit/s that practically limits the overall throughput of the drive. Additionally, another limitation is the absence of hardware RAID options, which would allow for redundancy and improved performance by creating and utilizing drive arrays.

4.2.2. Firmware and Booting

Raspberry Pi 4B does not include a built-in Unified Extensible Firmware Interface (UEFI). For booting the majority of modern operating systems such as Microsoft Windows or linux-based distributions, UEFI firmware is required. For the purposes of this research, a custom community UEFI firmware [29] based on the QEMU Tianocore EDK2 image [30] was employed. The utilization of the custom UEFI is achieved by storing it in a FAT16 or FAT32 formatted micro SD card inserted to the Raspberry Pi at all times. During the initial boot process, the custom ROM code is executed and loads the UEFI firmware while initializing the hardware components and establishing the operating system booting environment. Once the hardware and boot environment are prepared, the UEFI firmware loads the operating system from the predefined storage volume, which in this case is a USB3-attached hard drive. Finally, the loaded operating system takes control of the hardware, initializes further components and launches the user interface.

4.2.3. Power Supply and Thermal Protection

According to the manufacturer's specifications, the Raspberry Pi 4B requires a power supply of 5V DC capable of delivering a minimum current of 3A. During the course of this research, the official Raspberry 15.3W AC adapter (model SC0217) was used in order to provide sufficient power for the operational needs of the Raspberry Pi and to also cover the needs of the USB3 external drive [31]. Additionally, to ensure the safety of the board components during extensive operation and to mitigate the risk of high temperatures that could potentially damage it, a passive aluminum heatsink was installed that covers the CPU, chipset and memory modules.

5. Phase 2: Testing Environment Creation and Hypervisor Selection

This section focuses on the establishment of the testing environment utilizing Raspberry Pi 4B, involving both hardware specifications and the evaluation of hypervisors. Specifically, it provides an analysis of the necessary steps employed to configure the testing environment, detailing the hardware components and their specific configurations. Furthermore, this section analyzes the selection process of hypervisors for evaluation, presenting the outcomes of installation tests. Concluding, the main goal in this phase is to strategically choose a hypervisor for the upcoming research stages. This decision was reached by analyzing and evaluating factors such as compatibility and overall suitability for virtualization tasks on the Raspberry Pi 4B platform.

5.1. Hardware Environment Overview

Based on the observations and limitations described in Section 4, the selected hardware configuration to be used for the hypervisors' installation testing is summarized in Table 2:

Table 2. Hardware specifications of the testing system.

Hardware Attribute	Value
Board Model	Raspberry Pi model 4B
Processor Count	1 Physical Processor - 4 Cores
Processor Speed	1.5 GHz
Physical Memory	8 GB LPDDR4-3200
NIC Type	Single-Port 1 Gbps Ethernet Adapter
Primary Hard Disk	1 TB SATA3 SSD Drive
Hard Disk Controller	USB3.0 to SATA3 Adapter
Controller Speed	Max. 5 Gbit/sec

A graphical representation of the hardware environment is presented in Figure 2.

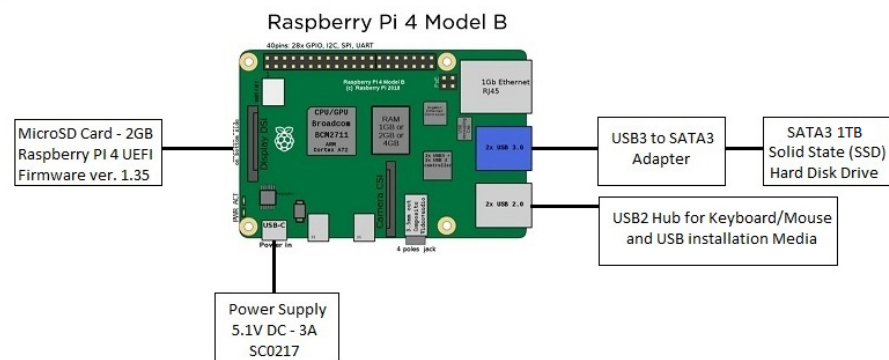


Figure 2. Primary hardware environment overview, including connected peripheral devices (storage, power supply and USB devices).

5.2. Hypervisor Installation Testing

The scope of this research is focused on Type 1 Hypervisors, also known as a “Bare-Metal” Hypervisors. Specifically, Type 1 Hypervisors are designed to directly operate on the host’s hardware, leveraging direct access and control to physical resources and allocate them to virtual machines without the need for an underlying operating system layer [32]. Type 1 Hypervisors commonly feature mechanisms for hardware isolation between hosted virtual machines and the ability to dynamically assign physical resources such as processors, memory and storage, as well as tools for live migration that enable administrators to move virtual machines between different hosts without service disruption.

Gartner, a prominent research and advisory company, regularly publishes its “Magic Quadrant for Server Virtualization Hypervisors”, which provides a ranking of the top Type 1 Hypervisors [33]. As per the latest Magic Quadrant, the top two Type 1 Hypervisors are as follows:

- **VMware vSphere:** VMware vSphere is a market-leading hypervisor that provides a wide range of virtualization and cloud management tools, making it the most commonly used hypervisor in enterprise environments.
- **Microsoft Hyper-V:** Microsoft Hyper-V is a mature and highly dependable hypervisor that integrates smoothly with the Microsoft products ecosystem, offering substantial security and automation features.

The aforementioned hypervisors are considered the best due to their performance, security, overall reliability, scalability and compatibility. They also provide a wide variety of tools and features that enable efficient management and monitoring for large-scale virtualized environments. Additionally, they are currently holding the largest market share in enterprise environments.

In order to evaluate the installation of Microsoft Hyper-V and VMware ESXi 7 on ARM, tests were conducted with the following outcomes:

5.2.1. Microsoft Hyper-V

In order to utilize Microsoft Hyper-V, it is necessary to install either a version of Microsoft Windows Server (2012 R2 or later), Microsoft Windows 10, or Windows 11 (Pro or Enterprise editions). ARM64 processor architecture is currently supported by Microsoft in the form of Windows 10 and 11 versions, which can be successfully installed on Raspberry Pi 4 hardware. However, Microsoft does not officially provide support for Raspberry Pi hardware, resulting in missing or improperly functioning device drivers. Specifically, the process of installing Windows 10 or Windows 11 designed for the ARM architecture on Raspberry Pi 4B presents several technical challenges and important considerations due to the use of unsupported hardware. One major challenge is the absence of official ARM64 versions of Windows publicly available for download. Consequently, users need to resort to alternative sources like “Unified Update Platform (UUP) Dump” to obtain these versions [34]. However, such unofficial sources carry inherent risks, including potential security vulnerabilities, system instability, and exposure to harmful code, raising concerns about the authenticity and legality of the downloaded files.

Another obstacle is the compatibility of hardware drivers. Since Microsoft does not officially support Windows 10/11 on ARM for Raspberry Pi 4B, users must rely on community-driven projects such as the “WoR (Windows on Raspberry) Project” [35,36] that offer alternative drivers. These drivers might not have official digital signatures, creating uncertainty about their origin and reliability. Activating “Test Mode” in Windows allows the installation of unsigned drivers, including community-developed ones. However, this compromises system security by permitting the installation of potentially unsafe or unstable drivers [37,38].

Running x86 applications on the ARM64 architecture of Raspberry Pi 4B requires emulation [39], leading to performance overhead that can affect the speed and responsiveness of certain applications, resulting in a degraded user experience [40]. Additionally, the limited availability of ARM64-compatible applications compared to their x86 counterparts may cause compatibility issues with certain software. Apart from the technical challenges, there are also licensing implications. Although the legal activation of Windows 10/11 on Raspberry Pi 4B can be achieved by using genuine purchased Windows keys, users must carefully review Microsoft’s licensing agreements to avoid violating their policies and facing potential legal consequences.

In conclusion, while installing Windows 10 or Windows 11 on Raspberry Pi 4B is technically feasible, the process involves several challenges and concerns such as using unofficial sources, unsupported hardware, and driver compatibility that may impact system security and stability.

Despite the availability of Hyper-V in the Microsoft Windows 10 ARM64 build 19559 and above, attempting to enable and install the Microsoft Hyper-V role under Windows 10 or 11 resulted in an operating system boot failure that persisted until the role was manually uninstalled through recovery mode. As for Microsoft Windows Server on ARM64, no public version has been released by Microsoft, although they are currently in the process of being developed for internal use and testing with Microsoft Azure. As a result of these findings, Microsoft Hyper-V was unable to be installed or tested on Raspberry Pi at the time of this research.

5.2.2. VMware ESXi 7 on ARM

In August of 2020, VMware Corporation announced and released a testing version of the ESXi 7.0 hypervisor specifically designed for ARM64 processors.

This version supports a number of Single-Board Computers, including Raspberry Pi 4. Although this version of ESXi is still considered a testing version and not an official product of VMware, it has reached a sufficient level of maturity to be considered for future enterprise-level implementations. The ESXi on ARM supports a variety of ARM64 Linux and Microsoft Windows operating systems, as well as most of the features provided in

commercial x86 versions, such as vMotion, vSAN, and vSphere central management. This makes it a suitable candidate for integration with large enterprise environments.

The process of obtaining the suitable installation image for VMware ESXi 7 on ARM64 is very straightforward. The installation image can be acquired directly from VMware's official website, which has a dedicated section solely devoted to the ESXi on ARM edition project. To access and download the appropriate version designed for Raspberry Pi, it is necessary to complete a free sign-up process in order to create a VMware account. Additionally, on the same website, there is the option of accessing several documentation and instructional resources related to installation on other supported ARM64 devices. Furthermore, a community forum is available for engaging in discussions about bugs and various topics of interest [41].

Once the appropriate image is downloaded in ISO form, it is necessary to prepare a UEFI bootable USB installation media based on it. This media will be used to boot the Raspberry Pi in the ESXi installation environment, which guides the user on installing the ESXi platform by selecting the destination drive and by providing the administrator's (root user) password. The entire installation process is very well described and documented by the ESXi on the ARM development team, and the installation documentation is also available for downloading along with the installation image.

Following the successful completion of the installation process, the ESXi management interface is accessible via web interface by navigating to the assigned IP address of the Raspberry Pi by a web browser on any computer connected to the same network.

As it can be observed from Figure 3, the ESXi on the ARM management web interface is identical to any other traditional x86 version, making it easy to navigate for administrators that are familiar with the ESXi products family. Additionally, by further navigating to the management and monitoring functions, all common ESXi options are present, and the processes for creating and modifying networking, data stores and virtual machines are also identical. This is a rather important factor since administrators that are familiar with VMware's ESXi environments may manage and maintain this version without needing any additional training.

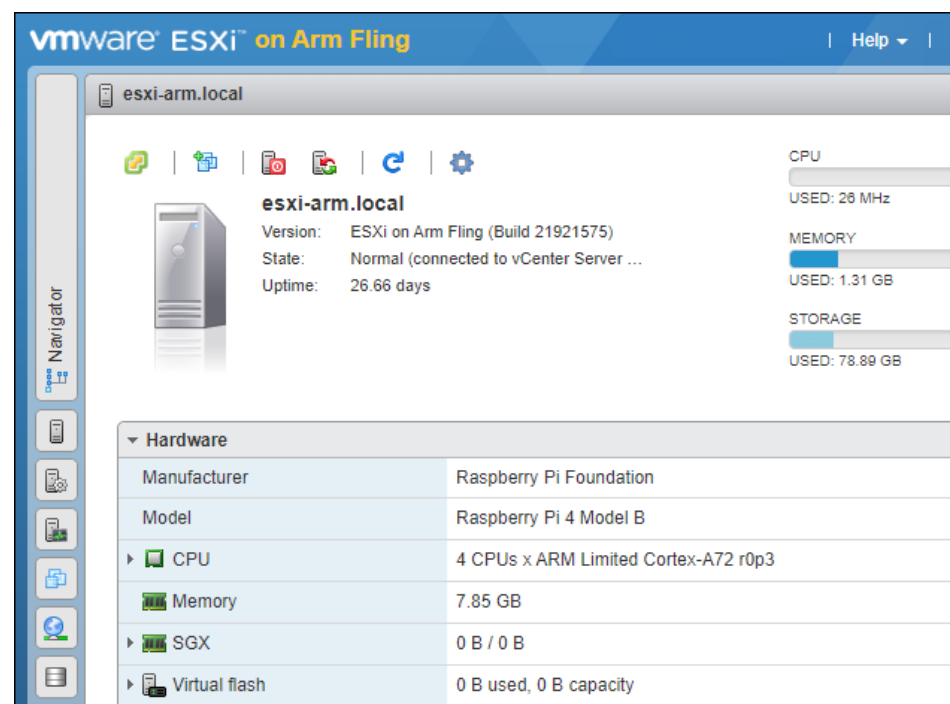


Figure 3. The ESXi on the ARM management web interface.

5.2.3. Centralized Management and Policies

The centralized administration of VMware ESXi hosts can be achieved via the vSphere vCenter Management platform, which enables the standardization and optimization of tasks such as VM deployment, resource allocation, and virtual hardware configuration. Additionally, vCenter provides advanced features for performance monitoring and task automation, enhancing the overall reliability and stability of virtualized environments [42]. Enrolling a Raspberry Pi 4B with pre-installed ESXi 7 into vSphere vCenter involves a straightforward process, specifically, when a Raspberry Pi is added as a new host to vCenter, becoming part of its inventory. This process is identical to the onboarding of traditional x86 systems, simplifying the integration with pre-existing environments. Once the new Raspberry host is enrolled, it can be assigned to an existing datacenter or cluster, allowing administrators to manage virtual machines, resource allocation, network interfaces and storage through vCenter's unified web interface. Moreover, administrators may easily monitor hardware health, performance and resource utilization of the edge infrastructure through the same web interface. This approach simplifies overall management, allowing administrators to focus on critical tasks rather than spend time on individual host management. Additionally, the integration with vCenter enables the creation of clusters with multiple Raspberry Pi hosts, providing advanced features like live migration (vMotion) that enable administrators to move virtual machines between hosts, facilitating easier hardware maintenance and optimizing overall resource utilization. The management dashboard for a Raspberry Pi 4B enrolled to the vCenter vSphere environment is presented in Figure 4.

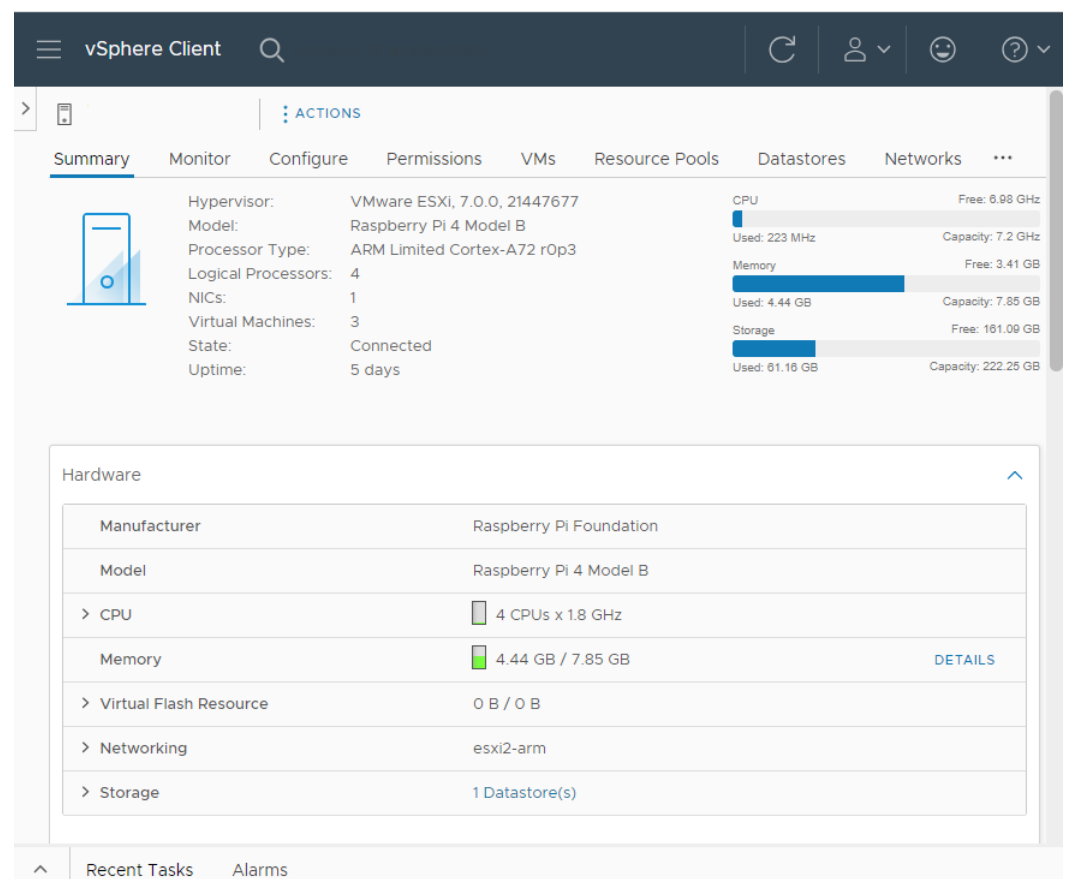


Figure 4. Raspberry Pi enrolled to the vCenter vSphere environment.

6. Phase 3: Case Study—The Edge Infrastructure of a Financial Organization

This case study focuses on evaluating the actual integration of a Single-Board Computer (SBC) server virtualization host as an edge-computing solution within an actual operating organization environment. The selected organization, apart from its traditional

branches, has a presence in various remote premises such as shop-in-a-shop locations, temporary remote sites, and B2B micro branches that are especially designed to serve the transactional needs of selected corporate large clients. In these described situations, a workstation that acts as a hypervisor is currently used to provide a networking infrastructure for the banking employees and connectivity with the main banking infrastructure. The purpose of this study is to evaluate the possibility of replacing these hosts with SBCs and migrate their workloads to the SBC environment.

The primary aim of this case study is to assess the integration of SBCs as server virtualization hosts in the organization's edge-computing environment and to evaluate their suitability and effectiveness. By utilizing SBCs, the organization aims to achieve a more cost-effective, scalable, and energy-efficient solution while ensuring uninterrupted connectivity and secure access to corporate resources.

During the implementation of this case study, valuable insights into the challenges, benefits, and practical implications of SBC integration in real-world edge-computing scenarios will be gained. The findings will provide guidance for the organization in making informed decisions about their infrastructure and exploring the potential of SBC-based edge-computing solutions.

Throughout the study, a comprehensive methodology will be employed, encompassing the evaluation of SBC performance, resource utilization, network connectivity, and security considerations. The specific edge-computing scenarios encountered within the banking organization, including shop-in-a-shop locations, temporary remote sites, and B2B micro branches, will be simulated and analyzed.

The outcomes of this case study will contribute to a better understanding of the capabilities and limitations of SBCs as server virtualization hosts in edge-computing environments within the banking industry. Practical insights will be provided for the organization to assess the viability of migrating workloads from traditional workstation hosts to SBCs, potentially leading to improved operational efficiency, cost savings and enhanced agility.

In the following sections, the methodology, experimental setup, and analysis of the results obtained from this case study will be outlined, shedding light on the potential benefits and challenges associated with the adoption of SBCs for edge computing within the banking industry.

6.1. Infrastructure Overview

The current IT infrastructure of the organization offers to the corporate users a variety of services. Specifically, the services that are relevant and directly affect the parameters of the specific case study are (a) active directory, (b) management and monitoring, (c) routing and firewall. The analysis for each of these services is presented below.

Active Directory Services: The active directory environment has been designed and implemented by utilizing Windows Server 2022 Datacenter edition as the domain controller operating system. The functional and forest level of the active directory has been raised to the highest available level (Windows Server 2016), ensuring access to the latest features and capabilities. The entire infrastructure is fully domain-joined, integrating all devices into the active directory domain to assure centralized administration, efficient user authentication and the simplified deployment of software updates and patches. Additional features and configurations include multiple password policies per different user groups, Sysvol DFS replication for efficient information exchange between domain controllers and the AD Recycle Bin for quick restoration of deleted data [43]. Through these policies and mechanisms, the active directory environment ensures enhanced security, flexible administration, and efficient user management, while the domain-joined infrastructure facilitates seamless authentication and centralized permission control across the enterprise network [44].

Management and Monitoring Services: The infrastructure has been equipped with advanced monitoring and management services, leveraging the full capabilities of SCCM (System Center Configuration Manager) and SCOM (System Center Operations Manager)

platforms. These platforms have been integrated into the environment, providing extensive configuration management, software deployment, and real-time monitoring capabilities. Specifically, SCCM serves as a centralized management tool, enabling administrators to perform configuration tasks across the infrastructure such as software deployment and operating systems updating [45]. SCOM, and acts as a monitoring platform that provides real-time visibility into the health, performance and availability of the infrastructure by collecting data from various sources such as servers, applications and network devices in order to detect issues and generate proactive alerts [46]. Another significant feature of the infrastructure is the existence of Microsoft Windows Deployment Services (WDS) 2022. WDS is a Windows Server server role that automates the deployment of Windows-based operating systems over the corporate network [47]. A significant aspect of WDS 2022 is its compatibility with ARM64 images. This feature has already been employed by the organization for facilitating deployments on portable devices that feature ARM64 architecture. Additionally, VMWare vSphere Server 7 is utilized to manage the physical hypervisors hosting the infrastructure, providing centralized control and administration of virtualized environments. vSphere Server is also configured to provide advanced features such as VMotion for live migration and High Availability (HA) for ensuring continuous operation, even in the event of host failures.

Routing and Firewall Services: Routing is achieved through a dedicated proprietary virtualized appliance, which is clustered on multiple VMs for high availability and provides firewall and routing services to the entire infrastructure. The network is segmented into several VLANs, with one VLAN serving as the internal network, and the others serving each dedicated broadband Internet line with firewall protection enabled. The IPsec server for LAN-to-LAN is enabled on each of them, and L2TP with the IPsec VPN server listens to all public IPs for remote client connection.

6.2. Edge Branch Infrastructure Overview

This subsection provides a detailed technical overview of the current edge branch infrastructure, which is implemented based on a single workstation featuring an Intel Core i5 650 processor, 8 GB of RAM, and 1TB SSD storage. Specifically, this workstation serves the role of a hypervisor by hosting a virtualized environment based on VMware ESXi 6.5 U3. The detailed hardware attributes of the edge workstation are presented in Table 3:

Table 3. Current edge workstation hardware attributes.

Hardware Attribute	Value
Processor Type	Intel Core i5 650
Cores/Threads Count	2 Cores/4 Threads
Processor Speed	3.20 GHz
Physical Memory	8 GB DDR3-1600 SDRAM
NIC Type	Single-Port 1 Gbps Ethernet Adapter
Primary Hard Disk	1 TB SATA3 SSD Drive
Controller Speed	Max. 6 Gbit/sec

6.2.1. Current Virtual Machine Configuration

The ESXi hypervisor hosts two virtual machines specifically configured to meet the requirements of the edge branch office. The first virtual machine features a FreeBSD installation configured for functioning as a routing appliance. Specifically, this VM is a proprietary build of a firewall, routing and VPN services software appliance. By establishing and managing the internal network, this VM provides a dedicated Class C network with integrated DHCP and DNS services, facilitating efficient IP address allocation and domain name resolution within the branch network. Moreover, this VM is responsible for establishing a secure VPN connection to the main organization's network, ensuring encrypted tunneling for secure data transfer.

The second virtual machine features a Microsoft Windows 10 Enterprise 22H2 installation functioning as an application server. Attached to the internal network provided by the FreeBSD-Based VM, it enables controlled access to corporate resources. Integration with the main organization's active directory enables centralized user management and domain policies enforcement. This integration enables the application of various policies, regulating user access and ensuring that authorized users may interact with specified corporate resources. Additionally, this Windows VM also acts as a local file server and as a Remote Desktop Protocol (RDP) host for facilitating specified users within the edge network. A graphical representation of the edge branch networking topology is given in Figure 5:

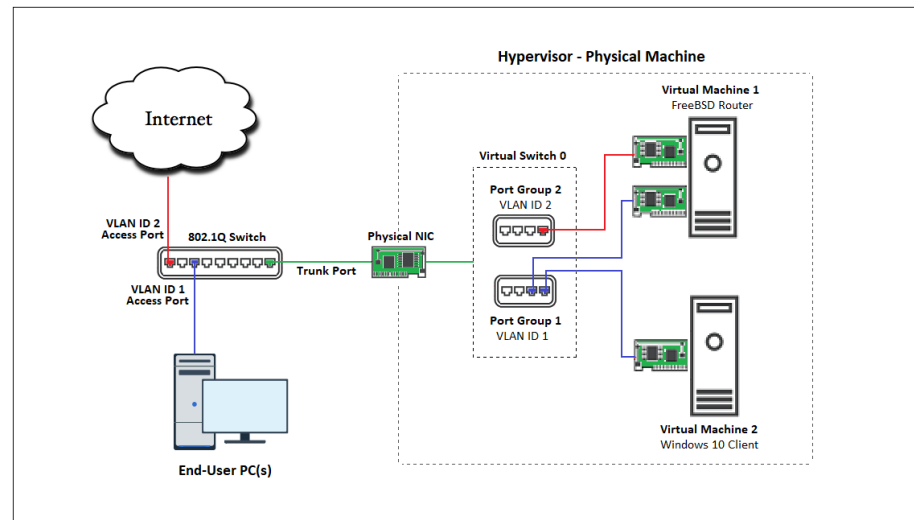


Figure 5. Representation of edge branch topology.

6.2.2. Current Implications and Upgrade Possibilities

The current implementation of the edge branch infrastructure introduces several implications that require careful consideration. Specifically, it is important to note that the currently used workstation represents an older model featuring a CPU that has reached its end of support, a fact that directly impacts the availability of future software updates and patches. Additionally, the specific workstation, being an older model, may exhibit a larger form factor, higher power consumption, higher noise levels and increased thermal overhead. Additionally, VMware ESXi 6.5 U3, although a stable and reliable hypervisor, is also an older version that practically limits the availability of functionalities and enhancements available in more recent versions.

Considering the above implications, it becomes apparent that a substantial upgrade is necessary to ensure that the edge infrastructure achieves similar or even higher efficiency compared to the existing setup. To address the limitations associated with the currently used infrastructure, an upgrade to a more modern and advanced solution was requested by the organization. One such solution is the utilization of a high-performance and low-energy consumption SBC that employs a scalable enterprise-level hypervisor. Towards this direction, the adoption of Raspberry Pi 4B in conjunction with ESXi 7.0 on ARM64 may benefit the edge branch with significant power saving compared to the older workstation-based approach. Additionally, the Raspberry Pi 4B's small form factor and low heat generation make it ideal for edge environments with limited physical space.

6.2.3. Transition Considerations

The transition to a Raspberry Pi 4B-based infrastructure presents an appealing solution to overcome the limitations of the existing workstation-based setup. By carefully managing the migration process and by leveraging the advantages of the Raspberry Pi 4B and VMware

ESXi 7.0, the edge branch can establish a more efficient and future-ready infrastructure for its networking and application needs. Nevertheless, transitioning from the existing infrastructure to a Raspberry Pi 4B-based one requires careful planning. The most important technical key factors that should be taken into account are analyzed below:

- **Operating System Compatibility:** The transition process requires the utilization of ARM64-compatible operating systems. Specifically, for both the FreeBSD and the Windows 10 VMs, an ARM64-based equivalent version should be employed. Since the direct migration of x86 VMs to the ARM64 architecture is not technically feasible, fresh installations and proper configuration of the new ARM64 VMs is required.
- **Resource Allocation and Capacity Planning:** Resource allocation for the Raspberry-based infrastructure should be identical to the existing VMs. Specifically, for the FreeBSD-Based VM, 1 GB of RAM, 64 GB of storage, and 2 CPU cores need to be allocated. For the Windows 10 Enterprise VM, 4 GB of RAM, 128 GB of primary storage, and an additional 512 GB virtual disk for shared storage are required.
- **Performance Testing and Validation:** A thorough testing process needs to be followed in order to ensure that the Raspberry Pi 4B-based infrastructure meets the expected performance requirements. Factors such as stability, responsiveness and capacity to handle the anticipated workloads should be evaluated to ensure a successful transition.

By considering the above-mentioned aspects and by carefully planning the transition process, the edge infrastructure may be successfully migrated to a Raspberry Pi 4B-based one while ensuring improved energy efficiency, reduced footprint, and enhanced service provisioning.

6.2.4. Transition Results

The transition to the Raspberry Pi 4B-based infrastructure yielded promising results, with several key steps and outcomes worth highlighting.

Firstly, the FreeBSD ARM64 variation was conveniently available preinstalled by the firewall service provider. This streamlined the transition process, as the FreeBSD VM could be effortlessly imported into the Raspberry ESXi infrastructure, significantly reducing the setup complexities. To ensure a flawless integration, Open VMware Tools were installed. In terms of networking performance, both the internal IP network and the VPN connection with the main organization's infrastructure performed consistently with the x86 variation previously installed on the workstation-based infrastructure. This consistency ensured that network operations continued without disruption, reassuring the organization's stakeholders about the reliability of the transition.

The deployment of the Windows 10 ARM64 VM followed a different distinctive path. Specifically, a pre-staged installation image that has previously been used for ARM64-based devices was utilized. The deployment of this image was made possible through the existing Windows Deployment Services 2022 (WDS) service of the organization. At this point, it should be noted that ESXi on ARM demonstrated its capability to perform UEFI boot over Ethernet for its hosted VMs, which was a crucial function for this deployment and an important feature concerning the overall testing process. A significant variation in the Windows 10 ARM64 VM pre-staged setup was the inclusion of the VMWare VMXNET 3 driver in the deployed image. This addition was necessary to successfully facilitate booting from the virtual network adapter, enhancing the compatibility of the VM within the infrastructure. However, it should be noted that VMware tools developed for ARM64 Windows were not available at the time of this research. As a result, certain devices such as the virtual SCSI and VMware Display adapter could not be utilized properly. Instead, the generic Microsoft Windows display driver was used, and the virtual disk was connected via the provided Virtual SATA controller. Additionally, integration services commonly accessible for x86 VMs were not available for ARM64, necessitating the employment of alternative configurations.

After the completion of the Windows 10 ARM64 VM installation, all preparatory actions, such as active directory and SCCM enrollment, were successfully achieved in the

same manner as in the x86 deployment. Active directory policies were successfully enforced, ensuring consistent management and security. Furthermore, essential services such as SMB3 file sharing and Remote Desktop Services (RDP) were thoroughly tested and proven to function seamlessly in the ARM64 environment. It is also worth mentioning that Windows 10 on ARM64 delivered an identical user experience to its x86 counterpart, emphasizing the feasibility of the ARM64-based infrastructure in maintaining user familiarity.

The above successful transition results emphasize the adaptability of the ARM64-based infrastructure, whereas, at the same time, they highlight the necessity for further adjustments regarding the ARM64 VMs in comparison with their x86 counterparts. Concluding the transition process, the successful virtual machines installation, the incorporation with existing services and the consistent user experience contribute to the overall success of the transition.

6.3. Performance Comparisons

This subsection presents a comparative performance testing between the newly implemented Raspberry-based infrastructure and the pre-existing workstation-based infrastructure that has been in operation. Specifically, this performance testing is focused on critical metrics that directly affect the performance of virtualized infrastructures. These metrics include average and peak CPU utilization, average and peak power consumption, average and peak datastore latency and the performance of live migration via vMotion. The specific comparative testing is performed so as to cover and explore two distinctive areas:

- The ability and efficiency of the newly created Raspberry-based infrastructure to manage the required workloads;
- To compare the efficiency and performance of the Raspberry-based infrastructure against the workstation-based infrastructure already in place.

6.3.1. CPU Utilization

CPU utilization is a critical factor for hypervisors, as it directly affects how efficiently virtualized environments operate. Since hypervisors manage multiple virtual machines that simultaneously operate on a single physical host, it is essential to manage CPU resources effectively. Specifically, high CPU utilization may lead to conflicts in the virtual machines operation, causing serious performance degradation for the entire virtualized infrastructure. On the contrary, low CPU utilization is an indication that a physical host might be under-utilized.

Proper CPU resource management ensures that VMs respond quickly and smoothly, leading to improved user experience and more stable virtualized environments. It also enables fast and reliable VM migration between hosts, ensuring uninterrupted performance during transitions.

The CPU utilization monitoring process for the Raspberry-based infrastructure and the workstation-based infrastructure running ESXi involved the use of two identically configured Windows 10 virtual machines, one per each host. Both VMs were joined to the organization's active directory, with all necessary group and security policies enforced, both per user and per computer.

To ensure comparison fairness, both infrastructures were connected to the same management VLAN and monitored using vSphere vCenter. The CPU monitoring for each infrastructure began at the moment each Windows 10 VM was started and lasted for 19 minutes with a sampling interval of 20 s between each recorded value.

To minimize any potential variations, both systems were connected to the same internal network, ensuring an equal administrative distance from the organization's domain controllers. This setup aimed to provide a consistent and controlled environment for monitoring CPU utilization on both infrastructures.

By following this monitoring process, it was possible to gather data on the CPU utilization of both infrastructures under similar conditions, enabling a meaningful comparison

between the Raspberry-based and workstation-based systems running ESXi. The consolidated collected data for both infrastructures are illustrated in Figure 6.

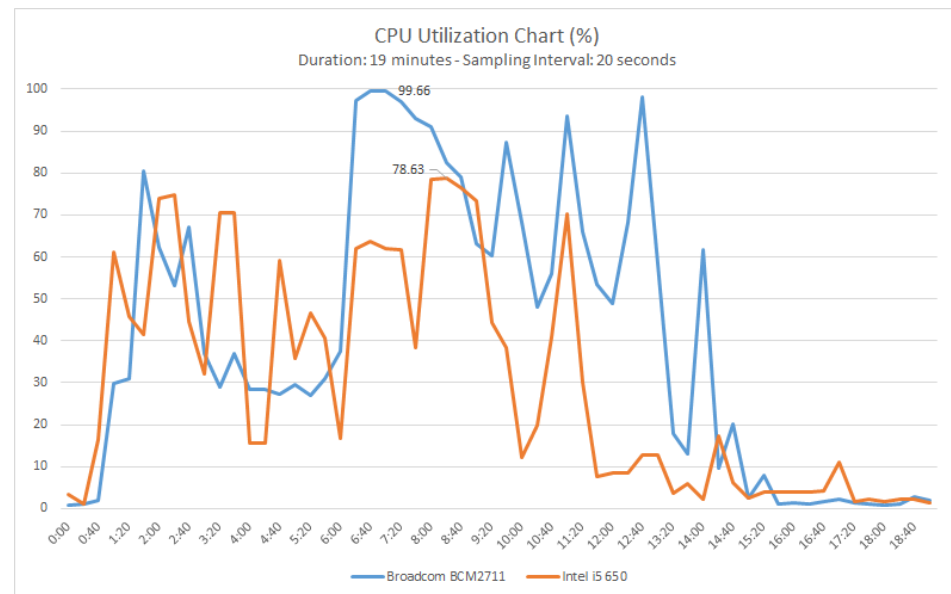


Figure 6. Consolidated CPU Utilization for BCM2711 and i5 650 processors.

As it can be observed, during the monitoring process, both systems exhibited considerable CPU utilization. In comparison, Raspberry Pi 4B infrastructure exhibited higher utilization with an average of 41.34% and a peak value of 99.6%, whereas the workstation-based infrastructure exhibited an average of 30.11% and a peak value of 78.63%. The overall CPU utilization results are presented in Table 4:

Table 4. CPU utilization comparison.

CPU Model	Average Utilization (%)	Peak Utilization (%)
Broadcom BCM2711	41.34	99.66
Intel i5 650	30.11	78.63

6.3.2. Power Consumption

The power consumption of IT equipment is crucial for all organizations since it directly impacts the overall operational cost, efficiency and environmental sustainability. One of the most important advantages of using ARM64-based systems is that their architectural design is based on the Reduced Instruction Set Computing (RISC) principles that mandate reduced power consumption [48].

To examine the impact in power consumption caused by the replacement of the edge infrastructure, the first step is to examine the electrical specifications of both the workstation and Raspberry Pi power supply units (PSUs). Specifically, the host of the workstation-based infrastructure is currently equipped with a DPS-300AB-10 PSU by Delta Electronics INC. As described in the manufacturer's datasheet, this particular PSU is capable of delivering a maximum power of 300 watts (W) with a maximum input current less than 4 amperes (A) at 230 volts (V) [49]. As previously mentioned in Section 4.2.3, the Raspberry Pi is equipped with the official Raspberry SC0217 USB-C power supply. According to the manufacturer's datasheet, this power supply is capable of delivering a maximum power of 15.3 W with a maximum input current of 0.5 A at 230 V [31].

Even though by directly comparing the electrical specifications of the two power supplies it is obvious that the workstation may exhibit much higher power consumption than the Raspberry Pi, this fact needs to be further validated by comparing their actual consumption during real-time operation. Power consumption (P) in watts can be determined

by multiplying the voltage (V) with the current (I). Assuming that the mains voltage is constantly 230 V, it is necessary to measure the actual value of the current. This can be achieved by connecting an ammeter in series between the main power and the power supply unit. For the purposes of this research, an UNI-T UT161D digital multimeter was employed and configured to measure the alternating current (AC) in amperes with a resolution precision of 0.001 A per reading [50]. The connection of the ammeter is represented in Figure 7:

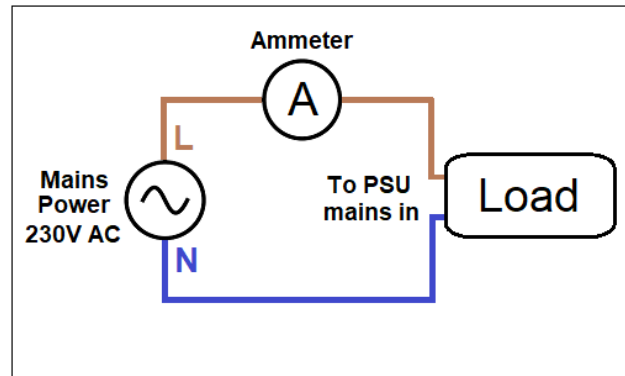


Figure 7. Ammeter connection diagram for PSU current measuring.

Since the power consumption is mainly affected by hardware utilization and particularly by CPU usage, the ammeter readings exhibit constant fluctuations. In order to ensure an accurate consumption calculation and a fair comparison, it is essential to gather multiple current readings in fixed intervals for both devices while executing a predefined utilization scenario.

For the purposes of this comparison, the same utilization scenario previously described in Section 6.3.1 was executed. In this case, the monitoring process involves the recording of the ammeter values for each infrastructure. The recording of current values for each host began at the moment that each Windows 10 VM was started and lasted for 19 min with a sampling interval of 20 s between each recorded value. Following the above process, it was possible to calculate the power consumption of both infrastructures under similar conditions enabling a fair comparison. The consolidated data of the calculated power consumption for both systems are illustrated in Figure 8.

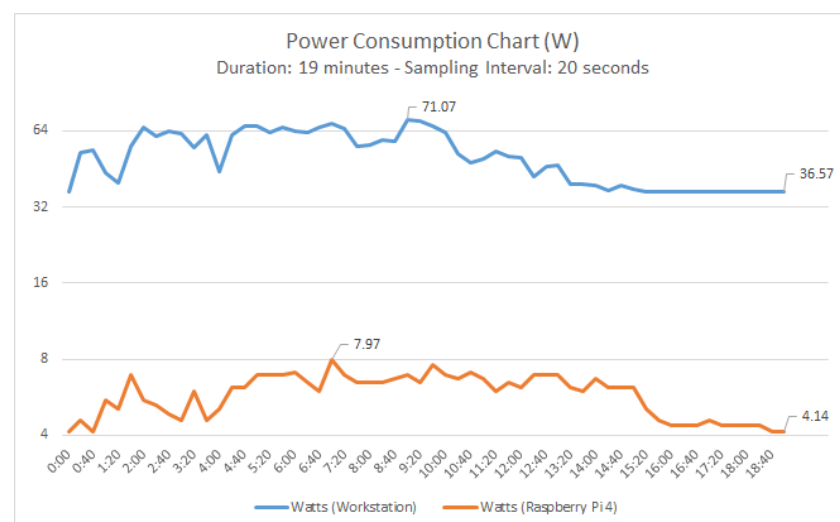


Figure 8. Power consumption for workstation and Raspberry Pi 4.

As it can be observed, there is a vast power consumption difference between the two systems. Specifically, the workstation-based infrastructure exhibited an average power consumption of 50.95 W with a peak value of 71.07 W, whereas the Raspberry-based infrastructure exhibited an average of 5.8 W with a peak value of 7.97 W. The overall power consumption results for both infrastructures are presented in Table 5.

Table 5. Power consumption summary.

Metric	Workstation	Raspberry Pi 4B
Min. Consumption (W)	36.57	4.14
Max. Consumption (W)	71.07	7.97
Average Consumption (W)	50.95	5.80

6.3.3. Datastore Latency

One of the most critical performance metrics for virtualized environments is storage latency. Storage latency refers to the time needed by a storage device to respond to read and write requests originating from virtual machines. Since in virtualized environments multiple VMs are sharing the same physical storage resources, when a single virtual machine requests to read or write data, this request is processed and submitted by the hypervisor to the storage system in order to process it and return the requested data back to the virtual machine.

The storage latency is commonly measured in milliseconds (ms) and defines the amount of time needed for the storage system to respond to read and write requests. Specifically, a low latency value indicates that a storage system responds quickly, and as a result, it may provide rapid I/O operations and fast data access. On the contrary, high latency values indicate slower response times, commonly leading to delayed I/O operations and overall performance degradation [51].

Specifically, storage latency is a critical metric for virtualized environments since it affects a number of important functions and features, such as overall responsiveness, workload performance, I/O intensive operations and live migration. High latency values may indicate either storage errors or performance issues concerning the underlying hardware such as locally attached storage controllers. Considering the main concern of the Raspberry-based infrastructure being the storage performance due to its USB3 connectivity, it was essential to compare the overall read and write latency of the Raspberry-based system with the currently used workstation. To achieve this, the following steps were undertaken:

- Both the Raspberry-based and workstation-based hypervisors were connected to the same management VLAN and concurrently monitored using vSphere Center.
- To assess the write latency, a file copy operation was initiated from a computer within the internal network to the Windows 10 Virtual Machine on each host. The duration of each file copy operation was six minutes.
- Similarly, to monitor the read latency, a file copy operation was initiated from the Windows 10 VM to the same computer used in the previous step. The duration of each read operation was also six minutes.
- The monitoring process was initiated simultaneously for each operation, and latency readings were sampled at 20 s intervals.

This approach allowed for a direct comparison of the read and write latency between the Raspberry-based and workstation-based hypervisors, providing insights into the storage performance of the Raspberry-based infrastructure. The consolidated results for both the read and write operations for each host are illustrated in Figure 9:

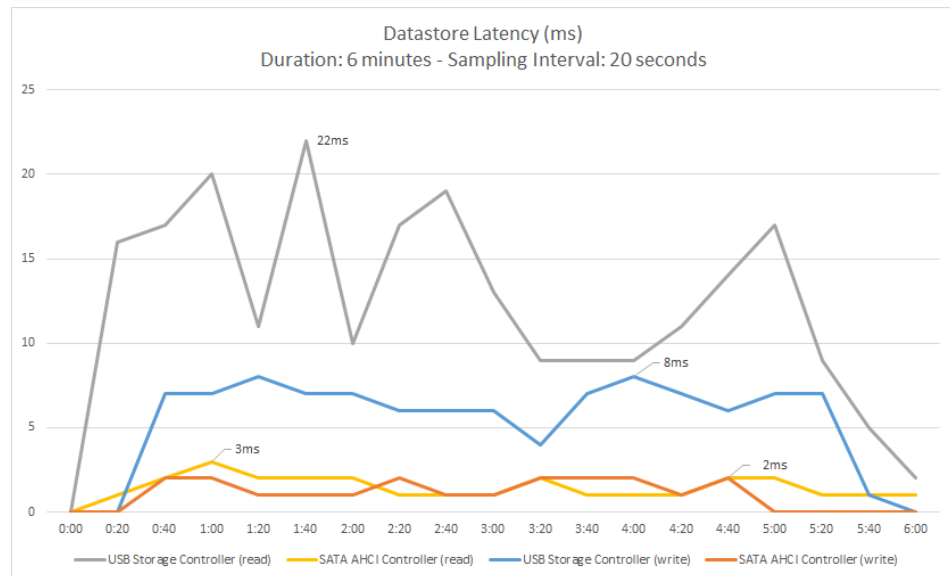


Figure 9. Consolidated datastore latency for USB and SATA AHCI controllers.

As it can be clearly observed, the overall datastore latency for the Raspberry-based infrastructure is considerably higher. Specifically, during the test, the Raspberry-based infrastructure exhibited an average latency of 12.1 ms for read operations and 5.31 ms for write operations, with peak latency reaching 22 ms and 8 ms, respectively, while the workstation-based infrastructure exhibited an average latency of 1.42 ms for read and 1.05 ms for write operations, with peak latency of 3 ms and 2 ms, respectively. These results highlight the considerable disparity in latency performance between the two infrastructures.

The latency testing results are summarized in Table 6.

Table 6. Datastore latency.

Storage Controller	Operation	Average (ms)	Peak (ms)
USB3 Storage Controller	Read	12.1	22
	Write	5.31	8
SATA AHCI Controller	Read	1.42	3
	Write	1.05	2

6.3.4. Live Migration (vMotion) Performance

The live migration of virtual machines (VMs) between hosts, a feature commonly referred to by VMWare systems as vMotion, is a critical component for every enterprise-level virtualized environment. Specifically, vMotion enables the migration of running VMs from a virtualization host to another, in real time, without disrupting their operation. Live migration holds significant importance for enterprise-level environments, offering a crucial solution to address the ever-evolving needs of modern IT operations. In critical occurrences, where maintaining uninterrupted services, optimizing resources, and retaining flexibility are essential, live migration plays an indispensable role. For example, within datacenters, live migration enables administrators to effectively distribute workloads, ensuring efficient resource utilization and real-time adaptation to varying demands, without causing service disruptions. Additionally, in the financial sector, live migration serves as a cornerstone of business continuity during hardware maintenance, safeguarding critical financial operations.

Since live migration is available only between hosts with the same processor architecture, for the purposes of this evaluation, an additional Raspberry Pi host was utilized,

featuring identical hardware and installation specifications with the one tested. This host was added to the same management network with vMotion enabled and enrolled to the vCenter Server 7 environment. The same process was followed for the workstation-based host. The testing process aims to evaluate the ability of the Raspberry hosts to perform live migration of their workloads and to compare their performance against the workstation-based hosts currently in place. Specifically, the testing process aims to assess vMotion's efficiency by migrating a Windows 10 ARM64 VM between two Raspberry Pi 4B hosts and a Windows 10 x64 VM between two hosts featuring the Intel i5-650 processor.

To ensure the integrity and fairness of the test, a consistent environment was established. All hosts, both ARM64-based Raspberry Pi and x86-based Intel i5-650, were enrolled in the same infrastructure, managed by vCenter Server 7. This configuration ensured that all hosts had an equivalent administrative distance, eliminating potential bias. The VMs on both hardware platforms shared identical configurations, including a 50 GB virtual disk, 4 GB of allocated RAM, and 4 virtual processors. Moreover, uniform domain policies and configurations were enforced on both VMs, enabling a comprehensive and equitable assessment of vMotion performance.

The vMotion process was meticulously scrutinized throughout the test to capture performance accurately. The procedure consisted of the following steps:

- **Infrastructure Setup:** All hosts were integrated into the same infrastructure, managed by vCenter Server 7, guaranteeing consistent administrative distances.
- **VM Configuration:** VMs on both hardware platforms were identically configured with a 50 GB virtual disk, 4 GB of RAM, and 4 virtual processors. Domain policies and configurations were synchronized between the VMs.
- **vMotion Initiation:** The vMotion process was initiated for each hardware configuration. This involved live-migrating the VM from one host to another within the same network.
- **Performance Monitoring:** CPU utilization and read/write latency were continuously monitored through vCenter Server during the vMotion process for both Raspberry Pi and Intel i5-650 environments. Specifically, the interval between values collected for both CPU utilization and datastore read/write latency was 20 s for both environments.
- **VM Availability Assessment:** To assess VM availability during migration, each VM was subjected to continuous pingging by another machine on the same network.

The results of CPU utilization and storage latency for the Raspberry-based infrastructure are illustrated in Figures 10 and 11, respectively, and summarized in Table 7:

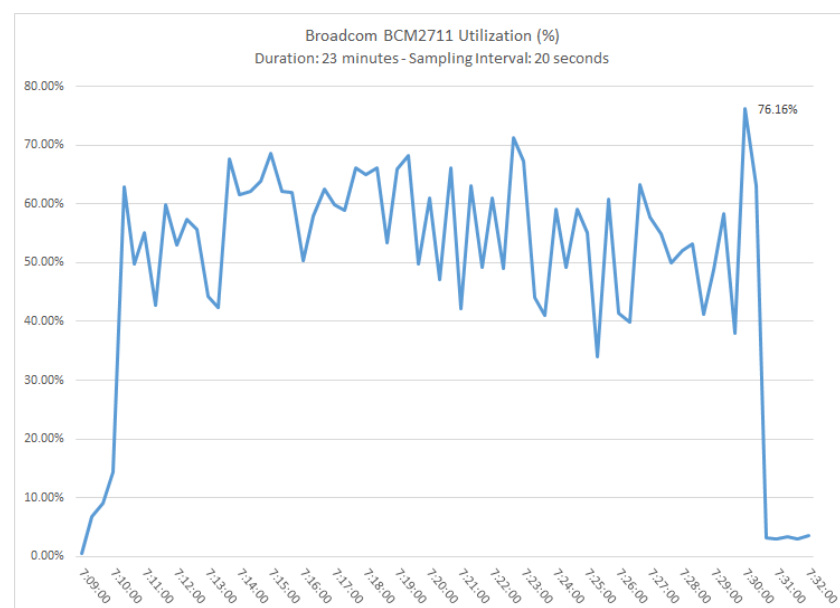


Figure 10. Broadcom BCM2711 utilization.

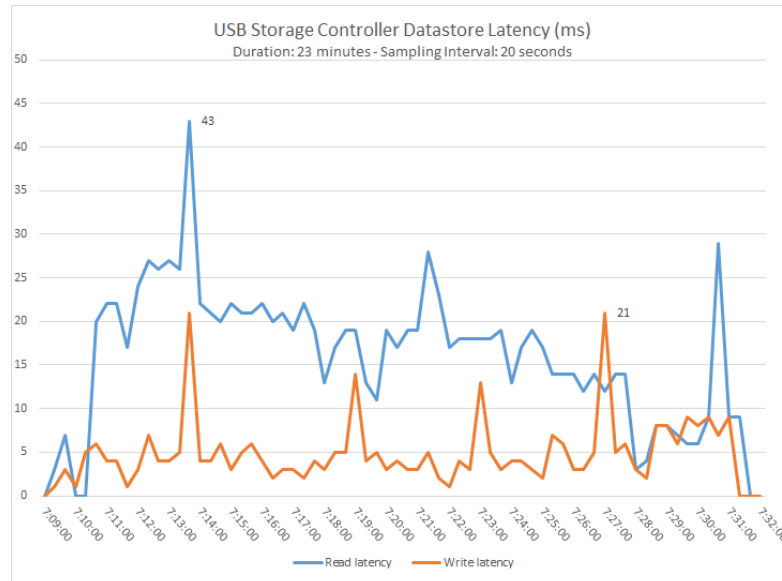


Figure 11. USB3 storage controller latency.

Table 7. Test results—Raspberry Pi 4B host.

Metric	Value
Migration Duration	23 min
Average CPU Usage	49.44%
Peak CPU Usage	76.16%
Average Read Latency	15.8 ms
Average Write Latency	4.8 ms
Peak Read Latency	43 ms
Peak Write Latency	21 ms
VM Availability	Packet Loss: 1%, Minimum: 0 ms, Maximum: 437 ms, Average: 3 ms

The results of CPU utilization and storage latency for the workstation-based infrastructure are illustrated in Figures 12 and 13, respectively, and summarized in Table 8:

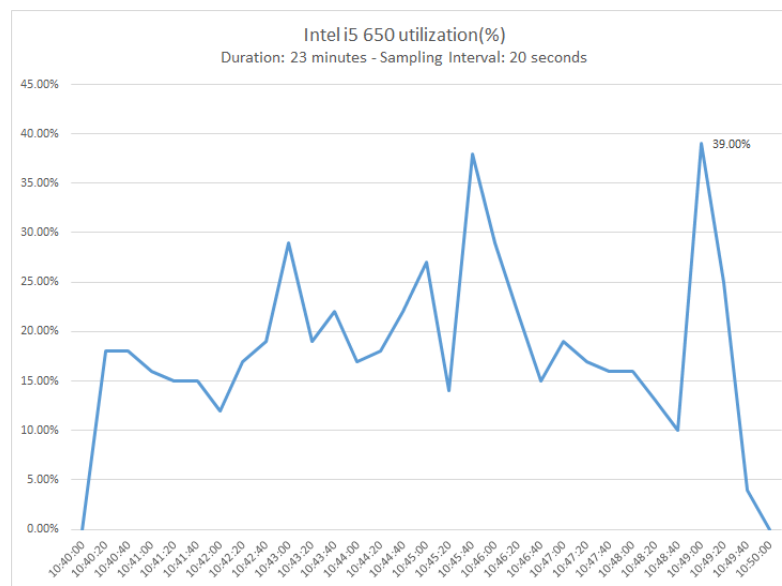


Figure 12. Intel i5 650 utilization.

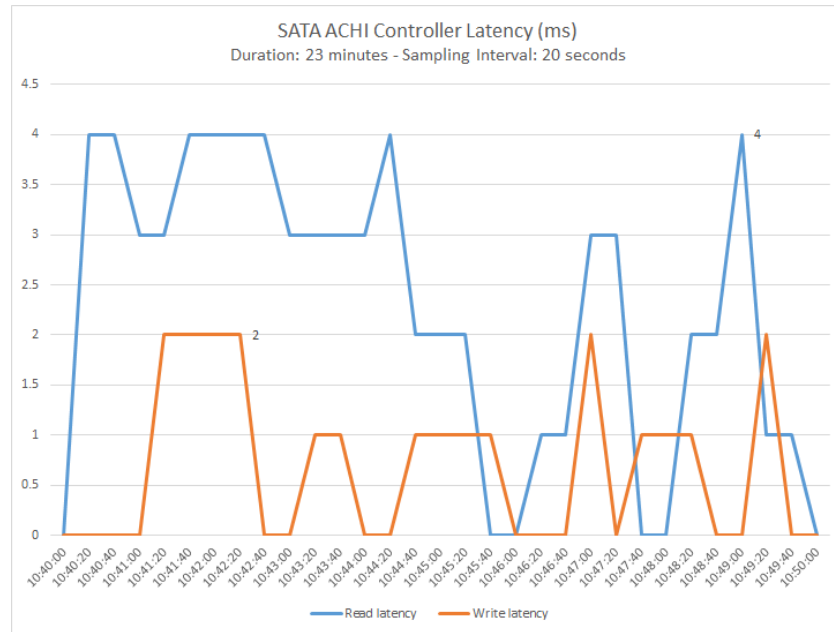


Figure 13. SATA ACHI controller latency.

Table 8. Test results—Intel i5 650 host.

Metric	Value
Migration Duration	10 min
Average CPU Usage	18.10%
Peak CPU Usage	39%
Average Read Latency	2.3 ms
Average Write Latency	0.7 ms
Peak Read Latency	4 ms
Peak Write Latency	2 ms
VM Availability	Packet Loss: 0%, Minimum: 0 ms, Maximum: 129 ms, Average: 1 ms

Based on the above results, the success of vMotion in an ARM64-based environment, exemplified by the Raspberry Pi 4B, showcases its adaptability to diverse hardware configurations. While the migration duration was significantly longer, and CPU utilization was higher on the Raspberry Pi, this test clearly demonstrates the feasibility of vMotion in ARM64-based systems, potentially opening doors for specific use cases. However, it is crucial to contextualize the elevated datastore latency observed in the Raspberry Pi environment. This latency can be once again attributed to the Raspberry Pi's storage, connected via a USB3 to the SATA adapter, introducing additional latency compared to direct SATA controller connections.

6.4. Discussion on Results

This subsection focuses on the examination of performance comparisons between the Raspberry Pi 4B and the workstation-based ESXi host, with a specific focus on CPU utilization, power consumption, datastore latency, and vMotion performance.

Specifically, CPU utilization, a critical consideration for hypervisors, was directly influenced by both systems, with significant utilization observed in both instances. As presented on Table 4, the Raspberry Pi-based infrastructure featured increased utilization, with an average of 41.34% and a significant peak value of 99.66%, whereas the workstation-based infrastructure demonstrated an average of 30.11% with a lower peak value of 78.63%. The higher CPU utilization in the Raspberry Pi environment suggests potential differences in processing efficiency between ARM64 and x86 architectures, which is crucial for resource

management and performance optimization. Nevertheless, this disparity can be partially justified since the Intel i5 650 processor, though older, is considered more powerful than the Broadcom BCM2711.

Power consumption is a crucial factor concerning computing infrastructures due to its major impact on operational costs, sustainability and overall efficiency of the systems involved. As presented in Table 5, the Raspberry Pi-based infrastructure exhibited an average consumption of 5.8 watts, ranging from 4.14 to 7.97 watts. On the contrary, the workstation-based infrastructure exhibited considerably higher values, with an average power consumption of 50.95 watts, ranging from 36.57 to 71.07 watts. This significant disparity in power consumption further validates the arguments regarding the power efficiency of ARM-based SBCs.

Datastore latency, another critical metric for virtualization infrastructures, also features significant disparities between the Raspberry Pi- and workstation-based infrastructures. Specifically, the Raspberry Pi infrastructure exhibited notably higher latency, with average read and write latencies of 12.1 ms and 5.31 ms, respectively, and peak latency reaching 22 ms and 8 ms. On the contrary, the workstation-based infrastructure demonstrated considerably lower latency, with average read and write latencies of 1.42 ms and 1.05 ms, and peak latency of 3 ms and 2 ms. These findings reveal the substantial performance gap in latency between the two infrastructures. The considerable disparity in latency performance between the Raspberry Pi- and workstation-based infrastructures reveals the major challenges and concerns in storage efficiency on the Raspberry Pi platform, which are directly attributed to the lack of a dedicated storage controller and the implementation of a USB3 to SATA adapter setup.

Live migration (vMotion) is an essential feature for enterprise-level virtualized environments, allowing the direct transfer of running virtual machines between hosts without disruption. The evaluation process was based on the migration of a Windows 10 ARM64 VM between two Raspberry Pi 4B hosts and a Windows 10 x64 VM between two workstation-based hosts. Despite the longer migration duration and higher CPU utilization observed in the Raspberry Pi environment, the successful execution of vMotion in the ARM64-based environment highlights its adaptability to diverse hardware configurations. The feasibility of vMotion in the ARM64-based Raspberry Pi environment, despite the significantly higher resource utilization and longer migration duration, suggests the potential versatility in ARM64 systems for specific use cases, while emphasizing the importance of optimizing for hardware variations, particularly storage latency. In conclusion, these results emphasize the potential utility of vMotion in ARM64-based systems, such as the Raspberry Pi 4B, for specific use cases. However, the significant disparities in CPU utilization and storage latency underscore the need for strategic hardware selection and optimization to maximize performance and ensure seamless VM migration in virtualized environments.

7. Conclusions and Future Work

The primary focus of this research was to examine the possibility of adopting SBCs on edge-computing scenarios by employing virtualization technology. During the course of hardware investigation, the Raspberry Pi 4B SBC was used as a reference to create a testing environment used to conduct an evaluation of Microsoft Hyper V and VMware ESXi 7 on ARM64 Type 1 Hypervisors. This evaluation revealed that while Microsoft Hyper V was unable to operate on Raspberry Pi 4B, the VMWare ESXi on ARM is fully operational and exhibited adequate performance and features compatibility to be considered the base platform for the rest of the research. During the course of both hardware and hypervisor investigations, a number of limitations were revealed that could potentially lead to performance degradation or disruption of the provided services. These limitations were mostly concerning hardware and software compatibility, storage performance and other reliability issues.

The transition from traditional x86-based edge infrastructure to an ARM64-based Single-Board Computer (SBC) setup brings both opportunities and challenges, notably in

the context of virtualization technology. ARM64-based SBCs, such as the Raspberry Pi 4B, excel in power efficiency, making them ideal for energy-conscious edge deployments, crucial in resource-constrained environments. Their compact form factor and minimal heat generation address space limitations in edge deployments. However, it is important to acknowledge that the ARM64 architecture, while gaining traction, is still less mature than x86, leading to issues mostly concerning software compatibility, driver support and system stability, especially regarding virtualization solutions.

Despite all the challenges mentioned above, during the course of this research, it has been proven that Raspberry Pi 4B may successfully operate as a dependable virtualization host in conjunction with VMware ESXi 7 on ARM. Additionally, even though ESXi on ARM is still under development and not an enterprise-ready product, the majority of its features are operational and can be adequately utilized. Furthermore, the host can be successfully managed by an existing vCenter Server infrastructure, featuring advanced remote management capabilities such as advanced health monitoring and live migration for virtual machines, enabling it to be fully integrated in an existing x86 enterprise-level infrastructure.

Specifically, during the course of the case study conducted in an actual environment of a financial organization, an edge-computing infrastructure based on a traditional x86 workstation, featuring VMware ESXi hosting two virtual machines was successfully replaced by a Raspberry-based host, supporting the same workloads. Even though comparative performance testing between the two edge infrastructures revealed higher CPU utilization and increased storage latency for the Raspberry-based host, we could potentially disqualify the solution due to poor performance, as there was no service downtime reported either for the host or for the hosted VMs. Additionally, the most severe performance issue revealed by the conducted tests is datastore latency, which was approximately ten times higher in comparison with the average latency exhibited by the workstation-based infrastructure. This situation can be directly attributed to the absence of a dedicated storage controller and not due to the lack of processing power. Nevertheless, it is essential to mention that during the execution of identical computing scenarios, Raspberry Pi exhibited a substantially lower power consumption of approximately nine times lower on an average scale compared to the workstation-based infrastructure, revealing the enormous potential for sustainability, energy efficiency and cost effectiveness of the solution.

In conclusion, transitioning from the x86-based edge infrastructure to ARM64-based SBCs, like the Raspberry Pi 4B, has been proven to be feasible and beneficial in terms of power expenditures but is still in its early stage due to the serious performance issues that need to be addressed. These issues derive from the lack of standardization concerning ARM64 SBCs, limited software support and hardware expandability that could potentially assist system engineers in solving critical issues such as storage performance by customizing the hardware according to the needs of each implementation. Nevertheless, the compelling difference in power consumption is a strong motivating factor for working on overcoming the above issues towards the creation of modern, cost-effective and environmentally friendly computing solutions. Even though this research revealed that for the specific edge environment, the poor storage performance could potentially disqualify the solution, it is important to clarify that all integration tests with the organization's infrastructure were successful and that all necessary services were provided properly and as intended. Because of that, the edge infrastructure developed for the purposes of this research can be used in the context of another case study featuring less disk-intensive VMs than Microsoft Windows. Additionally, as a future research, the possibility of replacing the USB3-based storage with a faster dedicated storage controller on a hardware level should be considered. Such research could involve performing hardware modifications on an SBC that is currently available or by employing another more efficient model that will be produced in the near future.

Author Contributions: Conceptualization, G.L.; data curation, G.L.; formal analysis, S.M., C.D. and L.M.; investigation, G.L.; methodology, G.L. and L.M.; project administration, S.M. and C.D.; resources, G.L. and L.M.; supervision, S.M. and C.D.; validation, S.M. and C.D.; writing—original draft, G.L.; writing—review and editing, S.M., C.D. and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: The work in this paper has been co-financed by Greece and the European Union (European Regional Development Fund-ERDF) through the Regional Operational Program “Attiki” 2014-2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: Georgios Lambropoulos acknowledges the use of ChatGTP 3.5 for correcting spelling, punctuation and grammar errors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Husain, B.H.; Askar, S. Survey on edge computing security. *Int. J. Sci. Bus.* **2021**, *5*, 52–60.
2. Lambropoulos, G.; Mitropoulos, S.; Douligeris, C. ARM64 Architecture: A Review in Virtualization Technology and Cloud Computing Maturity, in the context of Environmental Sustainability. In Proceedings of the SEEDA-CECNSM 2023: 8th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, Piraeus, Greece, 10 November 2023.
3. Ariza, J.A.; Baez, H. Understanding the role of single-board computers in engineering and computer science education: A systematic literature review. *Comput. Appl. Eng. Educ.* **2022**, *30*, 304–329. [[CrossRef](#)]
4. Ambrož, M.; Perna, J.; Haatainen, O.; Aksela, M. Promoting STEM Education of Future Chemistry Teachers with an Engineering Approach Involving Single-Board Computers. *Appl. Sci.* **2023**, *13*, 3278. [[CrossRef](#)]
5. Jovanović, P.; Mileusnic, M.; Pavić, B.; Mišković, B. Applications of the Single Board Computers in the Software Defined Radio Systems. In *Sinteza 2014-Impact of the Internet on Business Activities in Serbia and Worldwide*; Singidunum University: Belgrade, Serbia, 2014; pp. 882–886.
6. Baun, C. Performance and energy-efficiency aspects of clusters of single board computers. *Int. J. Distrib. Parallel Syst. (IJDPSS)* **2016**, *7*, 4. [[CrossRef](#)]
7. Kaup, F.; Hacker, S.; Mentzendorff, E.; Meurisch, C.; Hausheer, D. The Progress of the Energy-efficiency of Single-board Computers. *Tech. Rep. NetSys-TR-2018-01* **2018**.
8. Álvarez, J.L.; Mozo, J.D.; Durán, E. Analysis of single board architectures integrating sensors technologies. *Sensors* **2021**, *21*, 6303. [[CrossRef](#)] [[PubMed](#)]
9. IDC FutureScape. *IDC FutureScape: Worldwide IT Industry 2019 Predictions*; IDC FutureScape: Needham, MA, USA, 2018.
10. Angel, N.A.; Ravindran, D.; Vincent, P.D.R.; Srinivasan, K.; Hu, Y.C. Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies. *Sensors* **2021**, *22*, 196. [[CrossRef](#)] [[PubMed](#)]
11. Sriram Subramanian, B.C. Enabling More Agile and Sustainable Business Through Carbon-Efficient Digital Transformations. In *International Data Corporation, Sponsored by VMware*. 2020. Available online: <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/company/vmware-idc-whitepaper-2020.pdf> (accessed on 5 January 2024).
12. Shi, W.; Pallis, G.; Xu, Z. Edge computing [scanning the issue]. *Proc. IEEE* **2019**, *107*, 1474–1481. [[CrossRef](#)]
13. Rydning, D.R.J.G.J.; Reinsel, J.; Gantz, J. The digitization of the world from edge to core. *Fram. Int. Data Corp.* **2018**, *16*.
14. Lambropoulos, G.; Mitropoulos, S.; Douligeris, C. Improving business performance by employing virtualization technology: A case study in the financial sector. *Computers* **2021**, *10*, 52. [[CrossRef](#)]
15. Jusoh, M.Y.; Haron, H.; Kaur, J. Virtualization Technology to Support Green Computing Among IT Personnel in the Public Sector. In Proceedings of the Advances in Visual Informatics: 7th International Visual Informatics Conference, IVIC 2021, Kajang, Malaysia, 23–25 November 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 676–688.
16. Ju, H.; Wang, J.; Zhu, E.; Zhang, X.; Zheng, F. Design Scheme of a Docker Container File Isolation against Computer Virus Spreading. *Math. Probl. Eng.* **2022**, *2022*, 5348370. [[CrossRef](#)]
17. Lee, H.; Satyam, K.; Fox, G. Evaluation of production serverless computing environments. In Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018; pp. 442–450.
18. Christidis, A.; Moschogiannis, S.; Hsu, C.H.; Davies, R. Enabling serverless deployment of large-scale ai workloads. *IEEE Access* **2020**, *8*, 70150–70161. [[CrossRef](#)]
19. Xie, R.; Gu, D.; Tang, Q.; Huang, T.; Yu, F.R. Workflow Scheduling in Serverless Edge Computing for the Industrial Internet of Things: A Learning Approach. *IEEE Trans. Ind. Inform.* **2022**, *19*, 8242–8252. [[CrossRef](#)]

20. Shafiee, H.; Khonsari, A.; Mousavi, P. Serverless computing: A survey of opportunities, challenges, and applications. *ACM Comput. Surv.* **2022**, *54*, 1–32. [CrossRef]
21. Wang, Z.; Hu, Q.; Xiong, Z.; Liu, Y.; Niyato, D. Resource optimization for blockchain-based federated learning in mobile edge computing. *IEEE Int. Things J.* **2023**. [CrossRef]
22. Tang, Q.; Xie, R.; Yu, F.R.; Chen, T.; Zhang, R.; Huang, T.; Liu, Y. Collective deep reinforcement learning for intelligence sharing in the internet of intelligence-empowered edge computing. *IEEE Trans. Mob. Comput.* **2022**, *22*, 6327–6342. [CrossRef]
23. Raspberry Pi Foundation. Raspberry Pi 4 Model B. Available online: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/> (accessed on 5 January 2024).
24. Raspberry Pi Foundation. BCM2711 Peripherals. Available online: <https://datasheets.raspberrypi.com/bcm2711/bcm2711-peripherals.pdf> (accessed on 10 December 2023).
25. Microchip Technology Inc. LAN7515-USB 2.0 to 10/100/1000 Ethernet Bridge Controller. Available online: <https://www.microchip.com/wwwproducts/en/LAN7515> (accessed on 10 December 2023).
26. Cypress Semiconductor. CYW43455 Single-Chip 5G Wi-Fi IEEE 802.11ac MAC/Baseband/Radio. Available online: <https://www.cypress.com/documentation/datasheets/cyw43455-single-chip-5g-wi-fi-ieee-80211ac-macbasebandradio> (accessed on 10 December 2023).
27. Raspberry Pi Foundation. Raspberry Pi 4 Model B Datasheet. Available online: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf> (accessed on 10 December 2023).
28. Raspberry Pi Foundation. Raspberry Pi 4 Model B Schematic Diagrams. Available online: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> (accessed on 10 December 2023).
29. Pi Firmware Task Force (PFTF). GitHub Repository: Pftf. Available online: <https://github.com/pftf> (accessed on 19 June 2023).
30. Tianocore. GitHub Repository: Tianocore/edk2. Available online: <https://github.com/tianocore/edk2> (accessed on 5 December 2023).
31. Raspberry 15W USB-C Power Supply Product Brief. Available online: <https://datasheets.raspberrypi.com/power-supply/15w-usb-c-power-supply-product-brief.pdf> (accessed on 7 November 2023).
32. Budhprakash; Bhatia, A.; Bhattal, G. A comparative study of Various Hypervisors Performance. *Int. J. Sci. Eng. Res.* **2023**, *7*, 65–71.
33. Gartner Peer Insights. Server Virtualization Reviews 2023. Available online: https://www.gartner.com/reviews/market/server-virtualization?fbclid=IwAR05pJSWPVT9ATCOo_k2QXy3uxAt42eMQQVa0RDnCC2CVmfW6pE8jgCniy4. (accessed on 31 January 2023).
34. uup-dump-dev. uup-dump-dev/website-public—Frequently Asked Questions. Available online: <https://git.uupdump.net/uup-dump/misc/src/branch/master/FAQ.md> (accessed on 5 December 2023).
35. worproject. worproject/RPi-Windows-Drivers. Available online: <https://github.com/worproject/RPi-Windows-Drivers> (accessed on 5 December 2023).
36. Kurniawan, A. *Getting Started with Windows 10 IoT Core for Raspberry Pi 3*; PE Press: Riverside, CA, USA, 2016.
37. Microsoft Corporation. The TESTSIGNING Boot Configuration Option. Available online: <https://ro.scribd.com/document/127320381/The-TESTSIGNING-Boot-Configuration-Option-Windows-Drivers> (accessed on 5 December 2023).
38. Blaauwendraad, B.; Ouddeken, T.; Van Bockhaven, C. Using Mimikatz’ driver, Mimidrv, to disable Windows Defender in Windows. *Comput. Sci* **2020**.
39. Microsoft Corporation. Apps on ARM: X86 Emulation. Available online: <https://learn.microsoft.com/en-us/windows/arm/apps-on-arm-x86-emulation> (accessed on 5 December 2023).
40. Rendón, D. *Building Applications with Azure Resource Manager (ARM): Leverage IaC to Vastly Improve the Life Cycle of Your Applications*; Springer: Berlin/Heidelberg, Germany, 2022.
41. VMware. ESXi-Arm Edition. Available online: <https://flings.vmware.com/esxi-arm-edition> (accessed on 1 December 2023).
42. VMware. VMware vSphere Documentation. 2023. Available online: <https://docs.vmware.com/en/VMware-vSphere/index.html> (accessed on 5 January 2024).
43. Iainfoulds. *Active Directory Domain Services Overview*; Microsoft Learn. Available online: <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview> (accessed on 5 January 2024).
44. Binduf, A.; Alamoudi, H.O.; Balahmar, H.; Alshamrani, S.; Al-Omar, H.; Nagy, N. Active directory and related aspects of security. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 4474–4479.
45. Microsoft Corporation. Introduction to Configuration Manager. 2023. Microsoft Docs. Available online: <https://learn.microsoft.com/en-us/mem/configmgr/core/understand/introduction> (accessed on 2 October 2023).
46. Microsoft Corporation. Getting Started with SCOM—Microsoft Services Hub. 2023. Microsoft Services Hub. Available online: <https://learn.microsoft.com/en-us/services-hub/unified/health/getting-started-scom> (accessed on 2 October 2023).
47. Microsoft Corporation. Windows Deployment Services Overview. 2023. Microsoft Learn. Available online: [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/jj648426\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/jj648426(v=ws.11)) (accessed on 2 October 2023).

48. Dandamudi, S. *Guide to RISC Processors: For Programmers and Engineers*; Springer Science Business Media: New York, NY, USA, 2005.
49. Delta Electronics DPS-300AB-10 Manual. Available online: <https://www.manualslib.com/manual/228133/Delta-Electronics-Dps-300ab-10.html> (accessed on 7 November 2023).
50. UNI-TREND Technology Co., Ltd. UT161 Series Product Information. Available online: <https://meters.uni-trend.com/product/ut161-series/#> (accessed on 7 November 2023).
51. VMware, Inc. *Performance Best Practices for VMware vSphere 6.7*; Technical Report; VMware Inc.: Palo Alto, CA, USA, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.