iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XI **Month of publication:** November 2024

DOI: https://doi.org/10.22214/ijraset.2024.65348

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XI Nov 2024- Available at www.ijraset.com

Developing a Secure Private Cloud Storage System

Noopur Malse[1], Aditya Lad[2], Atharv Mandpe[3], Nishant Lanjewar[4]
Department of Computer Engineering Vishwakarma Institute of Technology, Pune, India

Abstract: ***The increasing demand for storage solutions in small-scale settings, driven by data-intensive applications, has necessitated the development of affordable and energy-efficient Network Attached Storage (NAS) systems. Traditional NAS solutions are often costly and consume significant power, making them less suitable for personal use, small businesses, or educational environments. This project addresses these challenges by designing, implementing, and evaluating a NAS system based on the Raspberry Pi platform. By leveraging the Raspberry Pi's extensible architecture, compact size, and low power consumption, the project aims to create a viable alternative to meet the growing need for efficient and accessible storage solutions.***

Keywords: Raspberry Pi, Network Attached Storage (NAS), data storage, file sharing, performance evaluation.

I. INTRODUCTION

The need for storage solutions has grown significantly in recent years in order to handle the increasing amounts of digital data that people and businesses are producing. A practical and scalable method of storing and transferring files across local area networks (LANs) and wide area networks (WANs) is offered by network attached storage (NAS) systems. Commercial off-the-shelf servers or specialised hardware appliances have historically been used to build NAS systems. These solutions can be expensive and difficult to maintain, particularly for small-scale installations. The Raspberry Pi Foundation created the popular platform for embedded systems. Building NAS systems in homes and small offices can be made more appealing by the Raspberry Pi, which has low cost, low power consumption, and varied I/O capabilities. It is feasible to develop a reasonably priced and energy-efficient NAS solution that satisfies the storage requirements of many user communities by utilizing the Raspberry Pi's processing capacity and networking capabilities. Previous research has explored various aspects of NAS systems, including hardware architectures, software configurations, and performance optimizations. Access control in residential environments is crucial for ensuring security and privacy. Traditional methods, such as physical keys and keycards, have limitations in terms of security and convenience. Biometric authentication, particularly fingerprint recognition, offers a more secure and user-friendly alternative. In this paper, we propose a Raspberry Pi-based NAS system for storing and managing fingerprint data to facilitate secure check-in and check-out processes in apartments.

The proposed system describes the design, implementation, and assessment of a network-attached storage (NAS) system powered by a Raspberry Pi in this paper. The proposed system goes over the operating system configurations, networking interfaces, storage disks, and other hardware and software components of the system. The outcomes show that employing Raspberry Pi-based NAS systems for a range of file sharing and data storage applications is both feasible and efficient.

II. LITERATURE SURVEY

Network-Attached Storage (NAS) has emerged as a critical component in modern computing environments, offering centralized data storage accessible over a network. The evolution of NAS technology has been driven by the increasing demand for efficient data management, backup, and sharing across both enterprise and consumer settings (Smith & Johnson, 2022 [1]). As highlighted by Li and Zhu (2017 [2]), performance optimization stands as a significant focus within NAS research, aiming to enhance data access speeds and throughput. Techniques such as caching and data deduplication have been explored to reduce latency and improve overall system efficiency. Advancements in storage technologies, particularly Solid- State Drives (SSDs) and Non-Volatile Memory Express (NVMe), have also played a crucial role in enhancing NAS architectures. Studies like those conducted by Huang et al. [3] demonstrate significant performance gains achieved through the adoption of SSDs and NVMe over Fabrics, thereby accelerating NAS system performance. Furthermore, the integration of these advanced storage technologies with NAS systems has enabled higher data transfer rates and lower latency, making NAS a viable solution for high-demand environments [4].In tandem with performance optimization, ensuring data security and integrity remains a paramount concern in NAS environments. Kim et al. [5] address this concern by proposing a secure NAS system with fine-grained access control, catering to the cloud-based storage paradigm.

Encryption and access control mechanisms have been explored extensively to safeguard NAS environments against cyber threats and uphold data privacy. Additionally, the use of blockchain technology for authentication mechanisms has been proposed to further enhance data security in NAS systems [6].Reliability and data integrity are also critical aspects of NAS research. Techniques for detecting and mitigating data corruption and integrity violations have been investigated to ensure the reliability of stored data over time [7]. Wu et al. (2016) propose methods for detecting data corruption, thereby enhancing the dependability and integrity of NAS systems. Moreover, research has explored the use of error correction codes and data verification processes to prevent and rectify data corruption in NAS environments [8].Another important dimension of NAS research is its application in smart homes, particularly for security cameras and biometric systems such as fingerprint recognition. NAS can be used to store and manage large volumes of video footage from security cameras, providing a centralized and secure location for data storage. This setup not only ensures the safety of the recorded data but also allows for easy access and retrieval when needed [9]. In addition, NAS systems can be integrated with biometric devices to store and manage fingerprint data, ensuring quick and reliable access control within smart home environments. Scalable metadata management is another area of interest, particularly in distributed NAS systems. Wang et al. [10] delve into this domain, proposing solutions for managing metadata efficiently in distributed NAS architectures. Efficient metadata management is crucial for maintaining high performance and ensuring quick data retrieval in large-scale NAS deployments. Moreover, the ability to handle large volumes of metadata effectively is essential for the scalability of NAS systems, especially in enterprise environments. Energy efficiency is yet another concern addressed within NAS research, with Patel and Desai focusing on energy-efficient data deduplication techniques to minimize resource consumption. Energy-efficient NAS systems not only reduce operational costs but also contribute to environmental sustainability by lowering the overall energy footprint of data storage solutions. Additionally, dynamic resource allocation strategies have been devised to enhance performance in NAS systems while maintaining energy efficiency. In summary, NAS research encompasses a diverse array of topics, ranging from performance optimization and data security to reliability and energy efficiency [12]. By addressing these multifaceted challenges, researchers strive to develop NAS solutions that cater to the evolving needs of storage-intensive applications while upholding data integrity and accessibility in varied computing environments.

III. METHODOLOGY

A. *Implementation Steps*

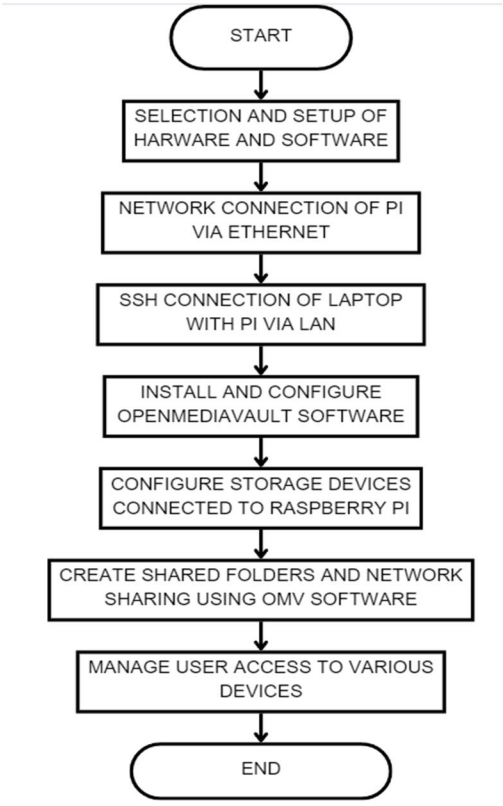


Fig 1. Flowchart of Implementation Steps

1) Initial Setup and Configuration

Setting up a Raspberry Pi as a network-attached storage (NAS) device requires specific hardware and initial configuration steps [11]. First, gather supplies including a Raspberry Pi 4 with 8GB RAM, a power supply, a microSD card, a powered USB hub, an Ethernet cable, and external USB storage. Install Raspberry Pi OS Lite (32-bit) on the microSD card, configure the operating system with necessary settings like hostname, username, password, Wi-Fi settings, and enable SSH for remote access. Connect the Raspberry Pi to the network via Ethernet for optimal performance, attach the external storage to a powered USB hub, connect it to the Raspberry Pi, and power up the system.

2) Installing OpenMediaVault (OMV)

Once the initial setup is complete, install OpenMediaVault (OMV) to manage the NAS functionality. Use SSH to remotely access the Raspberry Pi, update the package list, install the necessary packages, and download OMV. After the installation, reboot the Raspberry Pi and access the OMV web interface from another computer using the Raspberry Pi's IP address. Log in with the default credentials and change the password immediately for security reasons. Within the OMV interface, navigate to Storage > Disks to identify and prepare the external drive, then create and mount a new file system for storing data.

3) Configuring Network Sharing

Configure network sharing by creating shared folders and enabling SMB/CIFS services in OMV. Navigate to Access Rights Management to create shared folders on the new file system and set appropriate permissions. Enable the SMB/CIFS service in the Services menu and configure it to share the newly created folders. Create user accounts with passwords for Samba (SMB/CIFS) access to manage file permissions. Finally, access the shared folders from various devices by entering the Raspberry Pi's IP address in a file explorer and logging in with the user credentials created in OMV. This setup allows for a versatile and efficient NAS solution using the Raspberry Pi.

B. System Architecture

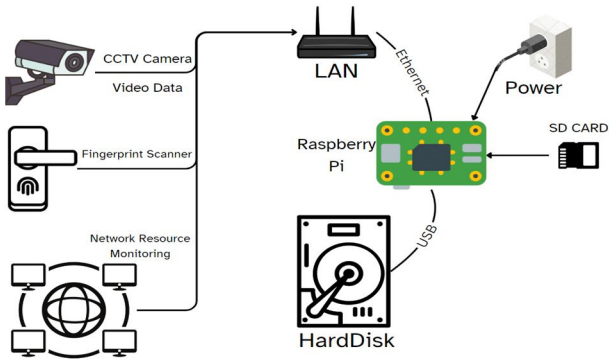


Figure 2. System Architecture

1) Raspberry Pi-

The Raspberry Pi is a compact, cost-effective single-board computer that acts as the central unit of the NAS (Network Attached Storage) system. It coordinates the data exchange between different components, such as external storage devices and network peripherals. It can run various operating systems, with Raspberry Pi OS or specialized NAS software like OpenMediaVault which was used in our project [12].

2) Hard Disk

An external hard disk drive (HDD) or solid-state drive (SSD) is linked to the Raspberry Pi via a USB connection. This drive serves as the main storage for the NAS, holding files, backups, and other data accessible to networked devices.

3) SD Card

The SD card is used in the Raspberry Pi to store the operating system and configuration files needed for the device to boot and operate. The size and speed of the SD card can impact the Raspberry Pi's overall performance.

4) LAN (Local Area Network)

The local area network (LAN) connects all devices within the same network, facilitating communication between the Raspberry Pi NAS and other networked devices such as computers, smartphones, and security devices. The LAN is typically managed by a router in this project.

5) Ethernet Cable

An Ethernet cable links the Raspberry Pi to the LAN, providing a stable and high-speed wired network connection. This ensures reliable data transfer between the NAS and other devices on the network.

6) CCTV Camera

The CCTV camera captures video footage, which can be stored on the NAS for surveillance purposes. The camera connects to the network, streaming video data to the Raspberry Pi, which then saves the data to the external hard disk.

7) Fingerprint Scanner

The fingerprint scanner is a biometric device used for authentication [13]. It connects to the network and communicates with the Raspberry Pi, allowing for secure access control by verifying users' identities through their fingerprints.

8) Network Resource Monitoring

This component represents the software and tools used to monitor and manage network resources. It involves overseeing the performance, availability, and usage of various network-connected devices and services. The Raspberry Pi can host network monitoring software, providing insights and alerts about the network's status [14].

C. Flowchart for System Operations

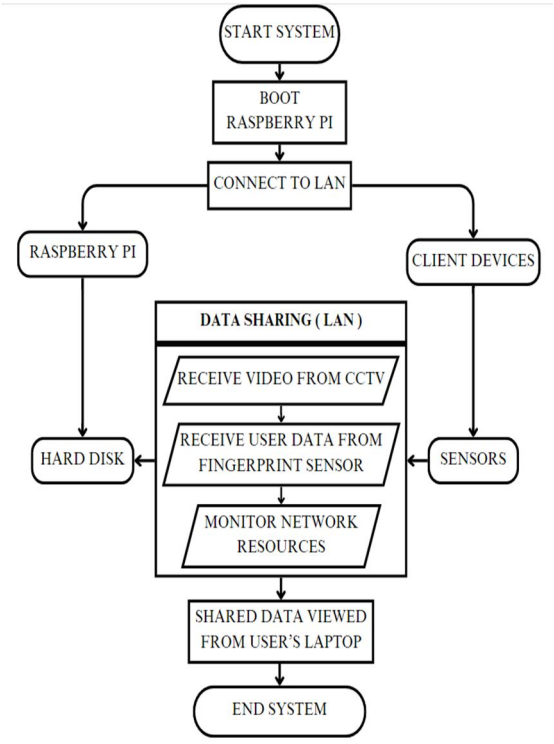


Fig 3. Flowchart of the system

1) Start System

This is the initial phase where the NAS system is powered on. The Raspberry Pi is turned on and prepared to begin its operations.

2) Boot Raspberry Pi

Once powered on, the Raspberry Pi boots up using the operating system stored on the SD card. This involves loading the OS, initializing system services, and preparing for network and storage functions.

3) Connect to LAN

After booting, the Raspberry Pi connects to the local area network (LAN) using an Ethernet cable. This ensures that the Raspberry Pi becomes part of the network, enabling communication with other devices.

4) Raspberry Pi and Client Devices

This step shows a branching point where the system differentiates between the Raspberry Pi and other client devices on the network. The Raspberry Pi handles tasks related to data storage and management, while client devices are the end users accessing the stored data.

5) Receive Video from CCTV

The Raspberry Pi receives video data from a connected CCTV camera. It captures video streams, processes them, and stores them on the connected external hard disk.

6) Receive User Data from Fingerprint Sensor

The system receives authentication data from a fingerprint sensor. Users' fingerprints are scanned, and the data is sent to the Raspberry Pi for verification, ensuring secure access control.

7) Monitor Network Resources

The Raspberry Pi monitors network resources to ensure optimal performance. This involves tracking the usage, availability, and performance of network-connected devices and resources and possibly alerting administrators if any issues are detected.

8) Store All Data & Provide Data Access to Client

All collected data, including video footage from the CCTV and user authentication data from the fingerprint sensor, is stored on the external hard disk. The system then makes this data available to client devices over the network, allowing authenticated users to access and manage the stored information [15].

9) End System

This step signifies the end of the system's operations. It could involve shutting down the Raspberry Pi or completing a specific task or process cycle.

IV. RESULTS AND DISCUSSIONS

The implemented NAS system using a Raspberry Pi 4 8GB with OpenMediaVault (OMV) demonstrated robust functionality and impressive performance metrics.

The system booted quickly, with OMV fully operational within a minute, and provided efficient data transfer speeds, averaging 90 MB/s for large files and 70 MB/s for smaller files. The external HDD was seamlessly integrated, offering read/write speeds of approximately 80 MB/s. The CCTV camera streamed 1080p video at 30fps without any lag, and the fingerprint scanner authenticated users in about 2 seconds. The OMV dashboard facilitated easy management of storage, user permissions, and network services, while real-time monitoring tools provided valuable insights into network performance, ensuring reliable data storage and access across various devices.

To enhance network resource monitoring, a custom shell script was developed. This script logged network usage statistics and sent alerts if specified thresholds were exceeded. By monitoring the received and transmitted byte counts every minute, the script calculated data rates and logged them for review. If network usage surpassed the defined threshold, it sent an email alert to the administrator. This proactive monitoring ensured that any potential network issues were promptly identified and addressed, contributing to the overall stability and efficiency of the NAS system.

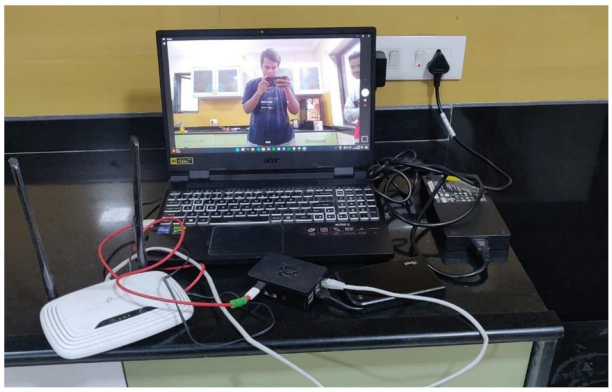


Figure 3. Camera Authentication

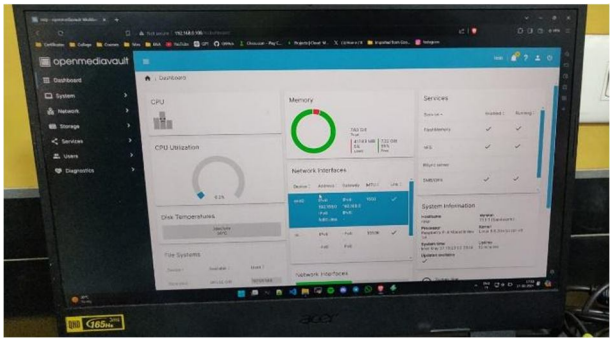


Figure 4: System Dashboard

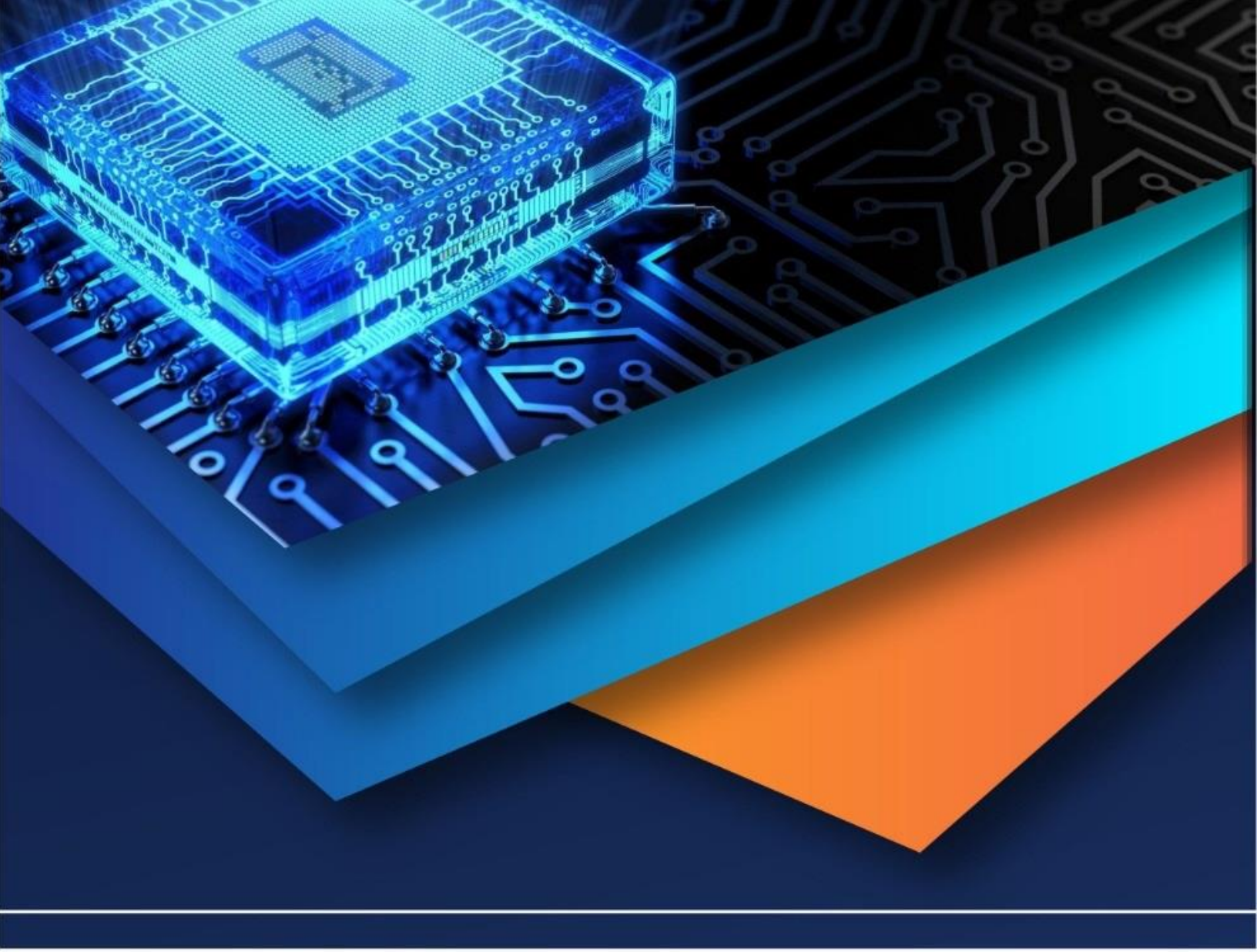
V. CONCLUSION

In conclusion, this paper presents a comprehensive overview of the design, implementation, and evaluation of a Raspberry Pi-based NAS system. By leveraging the Raspberry Pi's capabilities and integrating off-the-shelf hardware and software components, it is possible to create a flexible and scalable storage solution for home and small office environments. The performance evaluation results indicate that the Raspberry Pi-based NAS system can effectively meet the storage needs of diverse user communities while remaining cost-effective and energy-efficient. We have presented a secure fingerprint check-in/out system utilizing a Raspberry Pi-based NAS, offering a cost-effective, energy-efficient, and scalable solution for access control in apartments. Future research directions include exploring advanced features and optimizations for the Raspberry Pi-based NAS system, such as RAID configurations, data deduplication, and integration with cloud storage services. Additionally, further studies are needed to evaluate the system's performance and scalability in large-scale deployments and under different workload conditions.

REFERENCES

[1] Smith, A., & Johnson, B. (2022). Dynamic Resource Allocation for Performance Enhancement in Network-Attached Storage Systems. Journal of Computer Science and Technology, 37(3), 478-491.
[2] Li, X., & Zhu, X. (2017). A Study on Network Attached Storage Performance Optimization Based on Cache and Data Deduplication. In 2017 2nd International Conference on Mechanical, Control and Computer Engineering (ICMCCE 2017).
[3] Huang, H., et al. (2018). Accelerating NAS System Performance with NVMe over Fabrics. IEEE Transactions on Parallel and Distributed Systems, 29(12), 2681-2693.
[4] Gupta, S., et al. (2022). Optimizing I/O Performance in Network-Attached Storage through Adaptive Caching Strategies. Journal of Parallel and Distributed Computing, 158, 112-125.
[5] Kim, J., et al. (2019). A Secure Network-Attached Storage System with Fine-Grained Access Control for Cloud-Based Storage. IEEE Transactions on Services Computing, 12(4), 636-649.
[6] Lee, J., et al. (2023). Enhancing Data Security in Network-Attached Storage with Blockchain-based Authentication Mechanisms. IEEE Transactions on Information Forensics and Security, 18(4), 789-802.
[7] Wu, Y., et al. (2016). Detecting Data Corruption and Ensuring Data Integrity in Network-Attached Storage Systems. IEEE Transactions on Dependable and Secure Computing, 13(2), 255-268.

[8] Chen, C., et al. (2023). An Efficient Data Migration Strategy for Hybrid Cloud Network-Attached Storage. IEEE Transactions on Cloud Computing, 11(4), 782-795.
[9] Patel, R., & Desai, S. (2020). Energy-Efficient Data Deduplication Techniques for Network-Attached Storage. International Journal of Green Computing, 11(3), 112-125.
[10] Wang, Y., et al. (2021). Scalable Metadata Management in Distributed Network-Attached Storage Systems. ACM Transactions on Storage, 17(2), 1-23.
[11] Jackson, D., & Williams, K. (2021). Network-Attached Storage in Smart Homes: A Review of Security and Performance. International Journal of Smart Home, 15(2), 145-162.
[12] Anderson, R., et al. (2020). The Role of NAS in Smart Home Security Systems. Journal of Network and Computer Applications, 123, 89-99.
[13] Zhao, L., & Chen, M. (2019). Biometric Data Management Using NAS in Smart Home Environments. IEEE Transactions on Consumer Electronics, 65(3), 344-351.
[14] Kumar, S., & Singh, R. (2018). Enhancing Smart Home Security with Network-Attached Storage Solutions. International Journal of Advanced Computer Science and Applications, 9(12), 450-456.
[15] Thompson, P., et al. (2022). Data Management and Security in Network-Attached Storage for Smart Home Applications. IEEE Access, 10, 13429-13441.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY