applied sciences

MDPI

Article

Performance Analysis of Storage Systems in Edge Computing Infrastructures

Antonios Makris [1,*], Ioannis Kontopoulos [1], Evangelos Psomakelis [1,2], Stylianos Nektarios Xyalis [1], Theodoros Theodoropoulos [1] and Konstantinos Tserpes [1]

[1] Department of Informatics and Telematics, Harokopio University of Athens, 9 Omirou Str., 17671 Athens, Greece
[2] Distributed Knowledge and Media Systems Group, National Technical University of Athens, 9 Heroon Polytechniou Str., 15780 Athens, Greece
* Correspondence: amakris@hua.gr

Abstract: Edge computing constitutes a promising paradigm of managing and processing the massive amounts of data generated by Internet of Things (IoT) devices. Data and computation are moved closer to the client, thus enabling latency- and bandwidth-sensitive applications. However, the distributed and heterogeneous nature of the edge as well as its limited resource capabilities pose several challenges in implementing or choosing an efficient edge-enabled storage system. Therefore, it is imperative for the research community to contribute to the clarification of the purposes and highlight the advantages and disadvantages of various edge-enabled storage systems. This work aspires to contribute toward this direction by presenting a performance analysis of three different storage systems, namely MinIO, BigchainDB, and the IPFS. We selected these three systems as they have been proven to be valid candidates for edge computing infrastructures. In addition, as the three evaluated systems belong to different types of storage, we evaluated a wide range of storage systems, increasing the variability of the results. The performance evaluation is performed using a set of resource utilization and Quality of Service (QoS) metrics. Each storage system is deployed and installed on a Raspberry Pi (small single-board computers), which serves as an edge device, able to optimize the overall efficiency with minimum power and minimum cost. The experimental results revealed that MinIO has the best overall performance regarding query response times, RAM consumption, disk IO time, and transaction rate. The results presented in this paper are intended for researchers in the field of edge computing and database systems.

Keywords: performance evaluation; blockchain; BigchainDB; MinIO; IFPS; object storage; file storage; databases; edge; secure storage

Citation: Makris, A.; Kontopoulos, I.; Psomakelis, E.; Xyalis, S.N.; Theodoropoulos, T.; Tserpes, K. Performance Analysis of Storage Systems in Edge Computing Infrastructures. *Appl. Sci.* **2022**, *12*, 8923. https://doi.org/10.3390/app12178923

Academic Editors: Vassilios V. Dimakopoulos and Spiridoula V. Margariti

Received: 1 August 2022
Accepted: 3 September 2022
Published: 5 September 2022

1. Introduction

The amount of data generated by Internet of Things (IoT) devices is expected to grow dramatically in the future. According to Cisco [1], there will be almost 30 billion devices connected to the network by the end of 2023. Therefore, existing infrastructures will not be able to support, manage, and process such massive amounts of data. In fact, the current cloud infrastructure alone cannot support a large number of the current IoT applications as end devices are usually distant from the cloud servers, thus adding processing and network overhead, resulting in high latency, low bandwidth, and overall performance degradation.

A conceptual approach which combines the benefits of the cloud and the decentralized processing of services on edge devices is known as edge computing. Edge computing is a promising paradigm able to avoid network bottlenecks, overcome communication overheads, and reduce the data transfer delay [2–9], as the computational load is moved to the edge of the network, thus leveraging the computational capabilities of the edge nodes. Resource-rich computational resources are placed closer to mobile or IoT devices [10] and therefore edge computing offers higher scalability and availability than traditional cloud platforms [11,12]. Over the years, several edge architectures have been proposed to improve throughput, latency, and network coverage [13,14]. In order to realize the cloud/edge integration, various technologies from different domains should be combined, including computing, network, and application-oriented fragments [15]. Such an example is the use of Artificial Intelligence (AI) algorithms in the Internet of Vehicles (IoV) [16]—where the edge devices are vehicles—to help make dynamic decisions according to real-time requests.

One of the main challenges in the development of applications at the edge is the efficient data sharing between the edge nodes, and it can be accomplished within individual application frameworks or through an external storage service. Despite significant improvements in offering an efficient edge storage solution, there are still some issues to be addressed related to the functional and non-functional requirements of cloud/edge-based applications, including low data retrieval latency, high availability and integrity, dealing with a potential shortage of storage resources at an edge node, supporting rapid application component deployment or automatic restart/replacement of unresponsive components, and dealing with the high heterogeneity presented in edge environments. These requirements can be achieved by optimizing resource usage, allocation, and data management plans on edge devices. Hence, the edge storage needs to provide a reliable, fast, stable, and secure shared storage engine, and because it is designed for edge devices with limited resource capabilities, it needs to be extremely lightweight.

The limited computation, storage, network, or power resources of the edge nodes, along with the diverse application's requirements, pose several inherent challenges that need to be addressed, such as:

- The coordination of unreliable devices and networks.
- Hardware and software incompatibilities.
- The integration of different data storage formats and data types.
- The data locality (enabling low access time).
- Security concerns.
- QoS and QoE insurance.

The plethora of available storage systems and underlying technologies have left researchers and practitioners alike puzzled as to what is the best option to employ in order to manage and process, in the most efficient way, the massive amount of data generated by IoT devices. Therefore, this work focuses on highlighting the advantages and disadvantages of various edge-enabled storage systems. Thus, we present an overview and a performance analysis of three different storage systems in an edge context, namely MinIO, BigchainDB, and the IPFS. The performance evaluation is based on local access, by employing a set of resource utilization and performance metrics (QoS), during intense data transactions and during the normal functionality of the node. Each storage system is deployed and installed on a Raspberry Pi (RPi) device. A Raspberry Pi is a small, low-cost, single-board edge device, able to optimize and improve the overall efficiency with minimum power.

The rest of this paper is organized as follows. Section 2 presents the related work in the field of storage solutions for edge computing infrastructures. Section 3 presents an overview of the three evaluated storage systems, and Section 4 evaluates the performance against a set of resource utilization and performance metrics. Finally, Section 5 outlines the conclusion and future work.

2. Related Work

The Internet of Things and WEB 4.0 are quickly becoming more dominant in more and more domains and daily life or industrial applications. This gives rise to a series of new challenges and problems that researchers are actively trying to tackle, both in the cloud [17] and in the edge [18,19]. One of the major problems that falls in this category is the minimization of data latency and network overload in fog or edge networks [20]. One of the most common solutions for this problem is the development of edge storage methodologies in order to move all or part of the necessary data and their processing to the edge, near the edge devices that use them. Edge storage services are actively focusing on decentralization and resource efficiency due to the nature of the edge networks and the devices that are taking part in them. These two main goals are driving the current research in the field. A plethora of traditional technologies in storage are being adapted in order to fit these two requirements, such as the blockchain and block storage technologies.

Blockchain [21] is the well-known technology that came into existence in order to support Bitcoin, but since then, blockchain has developed a "life" of its own, being used in a great deal of other use cases and domains. Blockchain works by creating a central repository of transactions in the form of chained exchanges. Each of these exchanges must be validated by a number of peers in order to be registered in this central repository and be considered a valid transaction. When being applied in edge storage, blockchain has two major flaws: it needs heavy computational power to perform the transaction validations and it requires a centralized database in order to store the chain of transactions [22]. These two characteristics are causing direct conflict with the decentralization and low resource demand requirements of edge storage services. That is the reason that many researchers are trying to combine it with other technologies, such as peer-to-peer networks, limiting or even completely countering these flaws.

Peer-to-peer (P2P) networks are a form of file storage and file sharing technology that is fully decentralized. These types of networks are using a set of protocols that ensure the safe and secure communication between the interconnected devices, called "peers" [23]. These protocols are usually lightweight, adding only minimal overheads to the actual data that peers are exchanging between themselves [24]. Modern peer-to-peer networks are using distributed hash tables (DHT) in order to enhance their functionality and security, some of them even integrating encryption algorithms in order to protect their data from a wider set of possible attacks [25]. The problem with these networks is in the integrity, immutability, and reliability because they provide no adequate security controls over these factors [26]. This limitation is forcing researchers to combine them with other, more secure technologies, such as blockchain, which provide the missing controls.

The literature is actively trying to find a balance between the available frameworks by comparing their throughput, resource efficiency, and limitations, either on their own or when combined with each other. Blockchain and P2P networks are widely used for this purpose because peer-to-peer networks seem the ideal candidate for edge storage solutions, if the drawbacks already mentioned can be tackled. In relevant experiments, the interaction between these two frameworks seems to provide an efficient solution to the edge storage problem because blockchain can cover almost all of the weaknesses that P2P networks possess without adding much overhead, both in read/write operations, the throughput, and the network traffic [22,27–29]. The only drawback is that blockchain mechanisms require more redundancy than P2P, which requires more available disk space in the edge clusters that host these solutions, placing limitations on the network architecture options for IoT and fog networks.

Depending on the priorities of the researchers, two of the most important fields of interest in the relevant literature regarding edge storage architectures are security and resource efficiency [30]. In most of the cases, these two priorities are in direct conflict, because in order to improve the resource efficiency, some security rules need to be relaxed, and in order to improve the security, more resources need to be committed. For example, in systems that are based on blockchain and cryptographic security controllers, a great deal of middleware and network orchestrators are needed, allowing the framework to perform the necessary encryptions, decryptions, and security checks on each data transaction [31,32]. Some of the work performed in secure edge storage architectures prioritizes a different set of data security goals, such as availability and integrity. These approaches require a high redundancy which, again, is creating resource-demanding platforms [33–35]. Both erasure coding and data replication, which are the most common methodologies for ensuring availability and integrity, require additional nodes that are tasked with holding the replicated data and coordinating the data reading and recovery efforts. On the other hand, the systems that focus on high resource efficiency are usually bypassing data security altogether, focusing only on the data transfer and storage between the nodes, not taking into account the resources needed to secure the data packets transferred through the internet or the communication links between the nodes of the edge network [36–38]. These networks are often designed and evaluated with the assumption that data and network security are handled in another level of the data transfer and storage that is just out of their scope. Despite the fact that security is a major issue in every IoT system, cyber-risk regulations and assessment are still in their infancy. For that reason, the authors in [39] presented an analysis of cyber-risk assessment approaches in complex IoT systems and developed an epistemological analysis that enables the assessment of uncontrollable risk states in such systems.

The performance of IoT active devices can be improved by sharing their communication and computation resources. However, most works in the literature focus on either communication cooperation or computation cooperation. In [40], the authors proposed an energy-efficient resource allocation scheme in a wireless-powered MEC system, by leveraging a joint communication/computation cooperation among users. This joint strategy has been proven to reduce overall energy consumption compared to other state-of-the-art works. As far as QoS is concerned, it is difficult for users to select the services with the highest quality. Over the years, many studies have been conducted for QoS prediction in edge computing environments. In [41], the authors proposed a QoS prediction approach by employing and extending the ARIMA model. Finally, in [42], Vehicular Edge Computing (VEC) is presented as a mechanism for improving the QoS, where a volunteer-assisted model is utilized for computation offloading.

3. Storage Systems

The three evaluated storage systems belong to different types of storage. More specifically, MinIO is an object storage system, IPFS is file storage system, while BigchainDB is a blockchain database. In general, the different storage formats hold, organize, and present data in different ways, each with unique capabilities and constraints. File storage utilizes a hierarchy of files in folders, block storage divides data into arbitrarily organized, evenly sized volumes, while object storage links data with the associated metadata. Blockchain is a type of shared database that stores data as signed blocks which link to each other, creating a chain of immutable interconnected data entries. A high-level taxonomy of the different storage types is presented in Table 1.

Table 1. A taxonomy of the different storage types.

Feature	Block	Object	Filesystem	Blockchain
Data Access Method	Filepaths (usually)	Content Queries	Filepaths	Transactions
Storage Mode	Binary blocks	Documents	Files	Signed blocks
Scalability	Limited	Full	Not innate	Challenged
Metadata	No	Yes	Limited	Yes
Main Strengths	Distributed and Fast	Unstructured and Scalable	Simple and Secure	Security, Immutability, and Transparency

3.1. MinIO

MinIO https://min.io/, accessed on 5 July 2022, is an open-source framework created by IBM. It is an inherently decentralized and highly scalable P2P solution. It is designed to be cloud native, and by supporting a hierarchical structure, it is able to form federations of clusters. As data and metadata are written together as objects, there is no need for a metadata database. MinIO is in fact a combination of object storage and block storage, as it preserves the lightweight distributed nature of block storage while providing a plethora of metadata and the easy usage of object storage. Finally, MinIO is able to deliver the high-performance object storage that is required by modern big data applications.

3.2. BigchainDB

BigchainDB https://www.bigchaindb.com/, accessed on 5 July 2022, is a blockchain database that supports both blockchain (decentralization, immutability, and owner-controlled assets) and database properties (high transaction rate, low latency, indexing, and structured data querying) [43]. BigchainDB supports two transaction operations: CREATE and TRANSFER. A BigchainDB transaction is a JSON string that conforms to BigchainDB Transactions Specification. Each BigchainDB instance is a virtual concept consisting of three parts: (i) a MongoDB database which is used for the data storage locally, (ii) a BigchainDB server that is responsible for sending and processing requests, permission controls, encryption, decryption, transaction verification, and so on, and (iii) a Tendermint communication node that is a Byzantine Fault-Tolerant (BFT) middleware [44] for networking and consensus.

3.3. InterPlanetary File System

The InterPlanetary File System (IPFS) https://docs.ipfs.io/, accessed on 5 July 2022, is a P2P-distributed file system, designed for storing versioned file data in a decentralized manner [22]. The IPFS has been built on top of the BitTorrent protocol [45] and the Kademlia DHT [46]. BitTorrent is a widely used P2P filesharing system, which in the IPFS enables the efficient relocation of objects between peers composing the infrastructure. The Kademlia DHT is a popular DHT that is used for the management of the metadata. As a P2P system, there are not privileged nodes and no single point of failure. The IPFS nodes store objects in their local storage and maintain a DHT that is used to search the network address of the other peer.

4. Experimental Evaluation

The experimental evaluation presented below was performed on an RPi with a Quad core (Cortex-A72 (ARM v8) 64-bit SoC 1.5 GHz) and 8 GB of RAM (LPDDR4-3200 SDRAM), running Raspberry Pi OS with Python 3.6. The behavior of each database system is evaluated using a collection of small to medium binary files ranging from 15 KB to 10 MB, which form the evaluation dataset that is stored in the examined systems.

Experimental Results

The evaluation metrics utilized are divided into two categories: (i) resource consumption (total, used)—CPU, RAM, HDD, and network, and (ii) performance—throughput, data request response time, and network time.

The performance evaluation was executed through Locust https://locust.io/, accessed on 5 July 2022, an open-source load-testing framework that enables the definition of user behavior and supports running load tests distributed over multiple machines. In addition, it is able to simulate millions of concurrent user requests. For the purposes of the experiments, 20 users were set performing distributed query requests from four different machines. The results of the experiments are shown in Figure 1, where the average response time in milliseconds of each storage system is recorded. Figure 1a,b visualize the average response time of a single request of read and write operations, respectively. On the other hand, Figure 1c,d illustrate the average response time for all users' requests. The standard deviation of the response time is also illustrated in each figure in a stacked barplot manner on top of each average response time. Overall, as indicated in the above figures, MinIO presents the best performance in both the read and write operations. BigchainDB follows MinIO in all cases except for the write operation of a single request (Figure 1b), where the IPFS outperforms BigchainDB. Due to the object store nature of MinIO, it can be observed that the write operation of all requests is more time-consuming compared to BigchainDB. Moreover, the IPFS exhibits a high standard deviation, indicating a more unstable performance, thus yielding the worst performance out of the three storage systems.

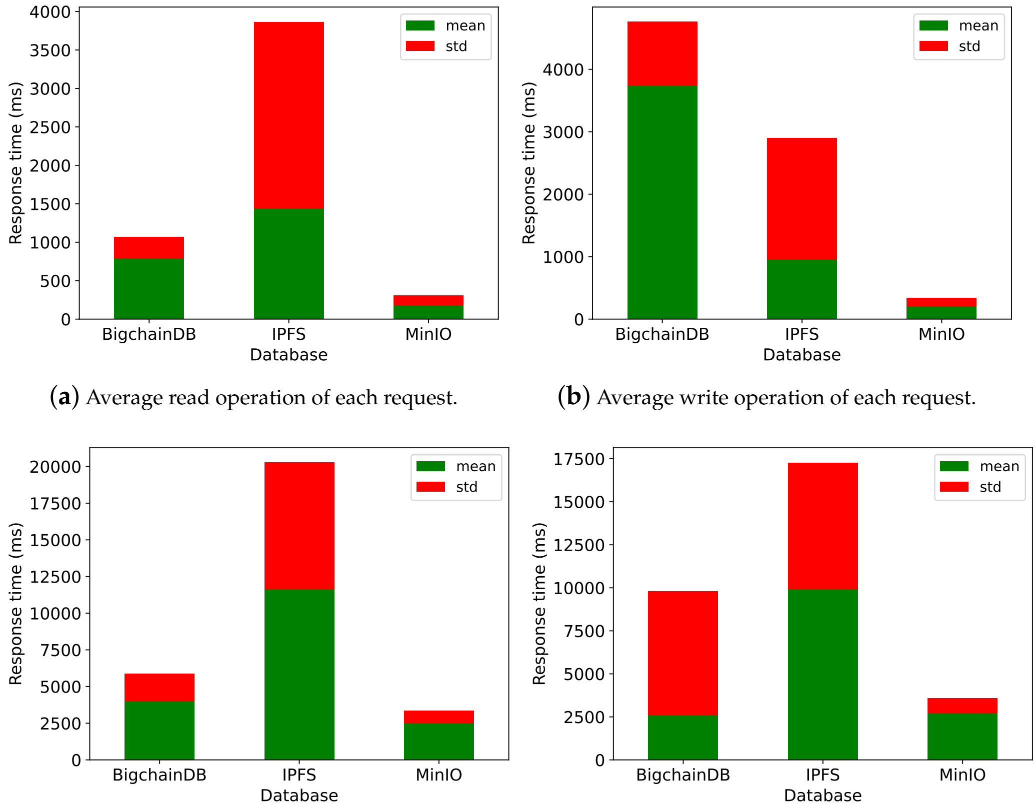
To further evaluate the storage systems, we also measured the RAM usage, the disk latency, and the disk IO time for a single user's request and for all users' requests, similar to the previous figures. The CPU was also recorded but not plotted because its usage was negligible. This proves that the storage is lightweight enough to be deployed on most edge devices, including the Raspberry Pis used for the evaluation. Figures 2 and 3 illustrate the statistics for the read and write operations, respectively. Figures 2a and 3a indicate the percentage of the RAM usage where, as depicted, MinIO consumes the least amount of RAM in each case. In addition, BigchainDB follows MinIO, only in the case of a single request, with the IPFS is ahead of BigchainDB in all users' requests.

In the rest of the figures where the disk latency (Figures 2b and 3b) and the disk IO time (Figures 2c and 3c) are presented, MinIO achieves the best performance followed by BigchainDB, while the IPFS yields the worst performance results. The disk metrics are increased by a larger degree, proving intense I/O activity.

Finally, Figure 4 demonstrates the transaction rate (TR) achieved by each storage system. The transaction rate can be defined as:

$$\mathrm{TR} = \frac{\sum \mathrm{transactions}}{\mathrm{total\_time}} \tag{1}$$

As the results suggest, MinIO achieves the highest TR followed by BigchainDB, while the IPFS exhibits the worst results. For instance, the TR obtained by MinIO is 3.3 and 1.3 times larger compared to the IPFS and BigchainDB, respectively.



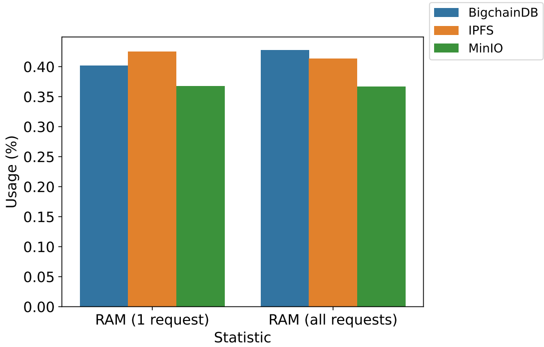
(**a**) Average read operation of each request.

(**b**) Average write operation of each request.

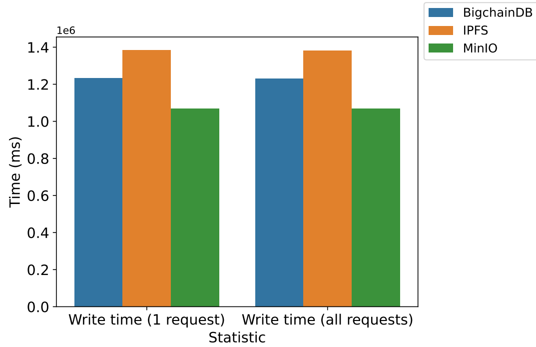
(**c**) Average read operation of all requests.

(**d**) Average write operation of all requests.

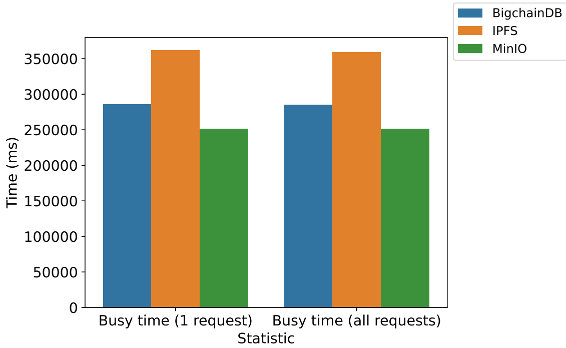
Figure 1. Performance of read/write operations of each database.



(**a**) RAM.

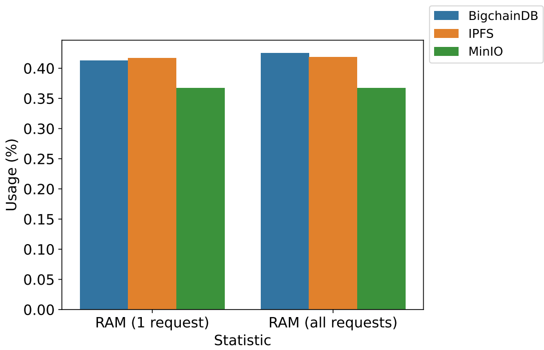


(**b**) Disk latency.

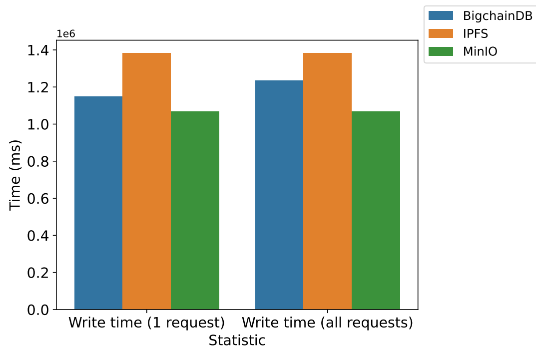


(**c**) Disk IO time.

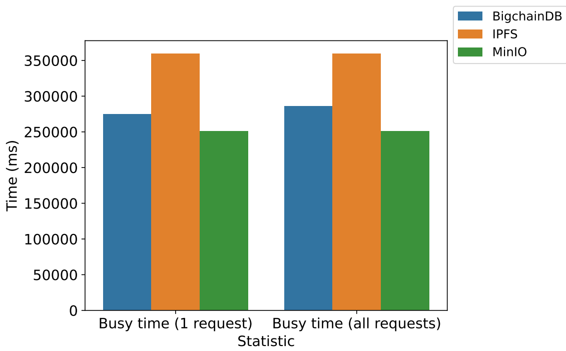
Figure 2. Statistics for the read operation of each database.



(**a**) RAM.



(**b**) Disk latency.



(**c**) Disk IO time.

Figure 3. Statistics for the write operation of each database.

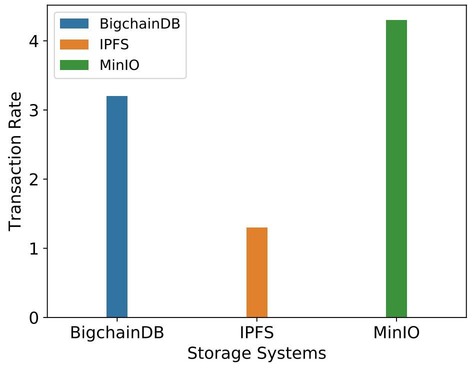


Figure 4. Transaction rate achieved by each storage system.

5. Conclusions and Future Work

The distributed and heterogeneous nature of the edge and its limited resource capabilities pose challenges in implementing or choosing an efficient edge storage system. In this work, we present an overview and a performance analysis of three different storage systems, namely MinIO, BigchainDB, and the IPFS. The effectiveness of each storage system is evaluated by employing a number of QoS and resource utilization metrics. Each storage system is deployed and installed on an RPi, which serves as an edge device, able to optimize the overall efficiency with minimum power and minimum cost. The experimental results demonstrated that MinIO yields the best performance in every setting while BigchainDB comes second in most cases. Furthermore, although the IPFS has a relatively low response time, it also exhibits a large variation in the response time between each operation, resulting in a high standard deviation from its average performance. Moreover, it is worth noting that the response times of each storage system are comparable to each other, and more workload and stress testing is required to further support the findings of our research.

One of the fundamental requirements for bulk data processing in cloud and edge infrastructures is database scalability. Therefore, as future work, we plan to examine the scalability of the evaluated storage systems by deploying them in the distributed mode. As already mentioned, each system belongs to a different type of storage; therefore, the distributed mode differs significantly. For instance, MinIO in the distributed mode sets up a highly available storage system with a single-object storage deployment, while the IPFS Cluster provides data orchestration across a swarm of IPFS daemons by allocating, replicating, and tracking a global pinset distributed among multiple peers. In addition, BigchainDB creates a network with no single point of control/failure, and the decentralized control is operated via a federation of voting nodes composing a P2P network. Finally, we intend to consider some additional metrics for the evaluation, such as the input/output operations per second and cache hit ratio.

Author Contributions: A.M., E.P., I.K., T.T. and K.T. worked on the conceptualization of the work. A.M., E.P., I.K., S.N.X. and T.T. worked on the methodology and, with K.T., developed the final solution. A.M., I.K., S.N.X., T.T. and E.P. worked on the experiments and the original draft preparation. K.T. helped with the writing—review and editing—of the final manuscript and performed the overall project supervision. All authors read and agreed to the published version of the manuscript.

Funding: This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101016509 (CHARITY) and No. 871793 (ACCORDION). Furthermore, this work was supported the SmartShip Project through the European Union's Horizon 2020 research and innovation program under the Marie-Sklodowska Curie Grant Agreement No. 823916. The work reflects only the authors' view, and the EU Agency is not responsible for any use that may be made of the information it contains.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Cisco, U. *Cisco Annual Internet Report (2018–2023) White Paper*; Cisco: San Jose, CA, USA, 2020.
2. Chiang, M.; Zhang, T. Fog and IoT: An overview of research opportunities. *IEEE Internet Things J.* **2016**, *3*, 854–864. [CrossRef]
3. Hao, Z.; Novak, E.; Yi, S.; Li, Q. Challenges and software architecture for fog computing. *IEEE Internet Comput.* **2017**, *21*, 44–53. [CrossRef]
4. Hu, Y.C.; Patel, M.; Sabella, D.; Sprecher, N.; Young, V. Mobile edge computing—A key technology towards 5G. *ETSI White Pap.* **2015**, *11*, 1–16.
5. Patel, M.; Naughton, B.; Chan, C.; Sprecher, N.; Abeta, S.; Neal, A. Mobile-edge computing introductory technical white paper. *White Pap.-Mob.-Edge Comput. (Mec) Ind. Initiat.* **2014**, *29*, 854–864.
6. Satyanarayanan, M. The emergence of edge computing. *Computer* **2017**, *50*, 30–39. [CrossRef]
7. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
8. Korontanis, I.; Tserpes, K.; Pateraki, M.; Blasi, L.; Violos, J.; Diego, F.; Marin, E.; Kourtellis, N.; Coppola, M.; Carlini, E.; et al. Inter-operability and Orchestration in Heterogeneous Cloud/Edge Resources: The ACCORDION Vision. In Proceedings of the 1st Workshop on Flexible Resource and Application Management on the Edge, Virtual Event, 25 June 2020; pp. 9–14.
9. Theodoropoulos, T.; Makris, A.; Violos, J.; Tserpes, K. An Automated Pipeline for Advanced Fault Tolerance in Edge Computing Infrastructures. In Proceedings of the 2nd Workshop on Flexible Resource and Application Management on the Edge, Minneapolis, MN, USA, 1 July 2022; pp. 19–24.
10. Ferrucci, L.; Mordacchini, M.; Coppola, M.; Carlini, E.; Kavalionak, H.; Dazzi, P. Latency preserving self-optimizing placement at the edge. In Proceedings of the 1st Workshop on Flexible Resource and Application Management on the Edge, Virtual Event, 25 June 2020; pp. 3–8.
11. Makris, A.; Boudi, A.; Coppola, M.; Cordeiro, L.; Corsini, M.; Dazzi, P.; Andilla, F.D.; Rozas, Y.G.; Kamarianakis, M.; Pateraki, M.; et al. Cloud for Holography and Augmented Reality. In Proceedings of the 2021 IEEE 10th International Conference on Cloud Networking (CloudNet), Cookeville, TN, USA, 8–10 November 2021; pp. 118–126.
12. Theodoropoulos, T.; Makris, A.; Boudi, A.; Taleb, T.; Herzog, U.; Rosa, L.; Cordeiro, L.; Tserpes, K.; Spatafora, E.; Romussi, A.; et al. Cloud-based XR Services: A Survey on Relevant Challenges and Enabling Technologies. *J. Netw. Netw. Appl.* **2022**, *2*, 1–22. [CrossRef]
13. Zhang, Y.; Zhang, H.; Cosmas, J.; Jawad, N.; Ali, K.; Meunier, B.; Kapovits, A.; Huang, L.K.; Li, W.; Shi, L.; et al. Internet of radio and light: 5G building network radio and edge architecture. *Intell. Converg. Netw.* **2020**, *1*, 37–57. [CrossRef]
14. Sittón-Candanedo, I.; Alonso, R.S.; Corchado, J.M.; Rodríguez-González, S.; Casado-Vara, R. A review of edge computing reference architectures and a new global edge proposal. *Future Gener. Comput. Syst.* **2019**, *99*, 278–294. [CrossRef]
15. Makris, A.; Tserpes, K.; Varvarigou, T. Transition from monolithic to microservice-based applications. Challenges from the developer perspective. *Open Res. Eur.* **2022**, *2*, 24. [CrossRef]
16. Xu, X.; Li, H.; Xu, W.; Liu, Z.; Yao, L.; Dai, F. Artificial intelligence for edge service optimization in Internet of Vehicles: A survey. *Tsinghua Sci. Technol.* **2022**, *27*, 270–287. [CrossRef]
17. Sandhu, A.K. Big data with cloud computing: Discussions and challenges. *Big Data Min. Anal.* **2022**, *5*, 32–40. [CrossRef]
18. Psomakelis, E.; Aisopos, F.; Litke, A.; Tserpes, K.; Kardara, M.; Campo, P.M. Big IoT and Social Networking Data for Smart Cities. In Proceedings of the 6th International Conference on Cloud Computing and Services Science, Rome, Italy, 23–25 April 2016; Volumes 1 and 2, pp. 396–405.
19. Rausch, T.; Rashed, A.; Dustdar, S. Optimized container scheduling for data-intensive serverless edge computing. *Future Gener. Comput. Syst.* **2021**, *114*, 259–271. [CrossRef]
20. Makris, A.; Psomakelis, E.; Theodoropoulos, T.; Tserpes, K. Towards a Distributed Storage Framework for Edge Computing Infrastructures. In Proceedings of the 2nd Workshop on Flexible Resource and Application Management on the Edge, Minneapolis, MN, USA, 1 July 2022; pp. 9–14.
21. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [CrossRef]
22. Kumar, S.; Bharti, A.K.; Amin, R. Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions. *Secur. Priv.* **2021**, *4*, e162. [CrossRef]
23. Hou, W.; Jiang, Y.; Lei, W.; Xu, A.; Wen, H.; Chen, S. A P2P network based edge computing smart grid model for efficient resources coordination. *Peer-Peer Netw. Appl.* **2020**, *13*, 1026–1037. [CrossRef]
24. Tracey, D.; Sreenan, C. How to see through the Fog? Using Peer to Peer (P2P) for the Internet of Things. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 47–52.
25. Daniel, E.; Tschorsch, F. IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 31–52. [CrossRef]
26. Subathra, G.; Antonidoss, A.; Singh, B.K. Decentralized Consensus Blockchain and IPFS-Based Data Aggregation for Efficient Data Storage Scheme. *Secur. Commun. Netw.* **2022**, *2022*, 3167958. [CrossRef]

27. Zhang, L.; Zeng, W.; Jin, Z.; Su, Y.; Chen, H. A Research on Traceability Technology of Agricultural Products Supply Chain Based on Blockchain and IPFS. *Secur. Commun. Netw.* **2021**, *2021*, 3298514. [CrossRef]
28. Confais, B.; Lebre, A.; Parrein, B. An Object Store Service for a Fog/Edge Computing Infrastructure Based on IPFS and a Scale-Out NAS. In Proceedings of the 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain, 14–15 May 2017. [CrossRef]
29. Nyamtiga, B.W.; Sicato, J.C.S.; Rathore, S.; Sung, Y.; Park, J.H. Blockchain-based secure storage management with edge computing for IoT. *Electronics* **2019**, *8*, 828. [CrossRef]
30. Radanliev, P.; de Roure, D. Review of Algorithms for Artificial Intelligence on Low Memory Devices. *IEEE Access* **2021**, *9*, 109986–109993. [CrossRef]
31. Ren, Y.; Leng, Y.; Cheng, Y.; Wang, J. Secure data storage based on blockchain and coding in edge computing. *Math. Biosci. Eng.* **2019**, *16*, 1874–1892. [CrossRef] [PubMed]
32. Yuan, L.; He, Q.; Chen, F.; Zhang, J.; Qi, L.; Xu, X.; Xiang, Y.; Yang, Y. CSEdge: Enabling Collaborative Edge Storage for Multi-Access Edge Computing Based on Blockchain. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 1873–1887. [CrossRef]
33. Javed, A.; Heljanko, K.; Buda, A.; Främling, K. CEFIoT: A fault-tolerant IoT architecture for edge and cloud. In Proceedings of the 2018 IEEE 4th world forum on internet of things (WF-IoT), Singapore, 5–8 February 2018; pp. 813–818.
34. Wu, J.; Li, Y.; Ren, F.; Yang, B. Robust and auditable distributed data storage with scalability in edge computing. *Ad Hoc Netw.* **2021**, *117*, 102494. [CrossRef]
35. Nijim, M.; Albataineh, H. Secure-Stor: A Novel Hybrid Storage System Architecture to Enhance Security and Performance in Edge Computing. *IEEE Access* **2021**, *9*, 92446–92459. [CrossRef]
36. Liu, J.; Curry, M.L.; Maltzahn, C.; Kufeldt, P. Scale-out Edge Storage Systems with Embedded Storage Nodes to Get Better Availability and Cost-Efficiency At the Same Time. In Proceedings of the 3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 20), Santa Clara, CA, USA, 6 February 2020.
37. Jin, H.; Luo, R.; He, Q.; Wu, S.; Zeng, Z.; Xia, X. Cost-Effective Data Placement in Edge Storage Systems with Erasure Code. *IEEE Trans. Serv. Comput.* **2022**, 1. [CrossRef]
38. Elgazar, A.E.; Aazam, M.; Harras, K.A. SMC: Smart media compression for edge storage offloading. In Proceedings of the 2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19), Renton, WA, USA, 9 July 2019.
39. Radanliev, P.; Roure, D.D.; Burnap, P.; Santos, O. Epistemological Equation for Analysing Uncontrollable States in Complex Systems: Quantifying Cyber Risks from the Internet of Things. *Rev. Socionetwork Strateg.* **2021**, *15*, 381–411. [CrossRef]
40. Mao, S.; Wu, J.; Liu, L.; Lan, D.; Taherkordi, A. Energy-efficient cooperative communication and computation for wireless powered mobile-edge computing. *IEEE Syst. J.* **2020**, *16*, 287–298. [CrossRef]
41. Yan, C.; Zhang, Y.; Zhong, W.; Zhang, C.; Xin, B. A truncated SVD-based ARIMA model for multiple QoS prediction in mobile edge computing. *Tsinghua Sci. Technol.* **2021**, *27*, 315–324.
42. Zeng, F.; Chen, Q.; Meng, L.; Wu, J. Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3247–3257. [CrossRef]
43. McConaghy, T.; Marques, R.; Müller, A.; De Jonghe, D.; McConaghy, T.; McMullen, G.; Henderson, R.; Bellemare, S.; Granzotto, A. Bigchaindb: A Scalable Blockchain Database. White Paper BigChainDB 2016. Available online: https://git.berlin/bigchaindb/site/raw/commit/b2d98401b65175f0fe0c169932ddca0b98a456a6/_src/whitepaper/bigchaindb-whitepaper.pdf (accessed on 5 July 2022).
44. Kwon, J. Tendermint: Consensus without Mining. Draft v. 0.6 Fall. 2014. Volume 1. Available online: https://tendermint.com/static/docs/tendermint.pdf (accessed on 5 July 2022).
45. Cohen, B. Incentives build robustness in BitTorrent. In Proceedings of the Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, 21–22 February 2003; Springer: Berlin/Heidelberg, Germany, 2003; Volume 6, pp. 68–72.
46. Maymounkov, P.; Mazieres, D. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*; Springer: Berlin, Germany, 2002; pp. 53–65.