Scalable Distributed Hyperledger Fabric

Subjects: Computer Science, Hardware & Architecture

Contributor: Houshyar Honar Pajooh, Mohammad A. Rashid, Fakhrul Alam, Serge Demidenko

Blockchain technology, with its decentralization characteristics, immutability, and traceability, is well-suited for facilitating secure storage, sharing, and management of data in decentralized Internet of Things (IoT) applications. Despite the increasing development of blockchain platforms, there is still no comprehensive approach for adopting blockchain technology in IoT systems. This is due to the blockchain's limited capability to process substantial transaction requests from a massive number of IoT devices. Hyperledger Fabric (HLF) is a popular open-source permissioned blockchain platform hosted by the Linux Foundation.

Keywords: blockchain ; Hyperledger Fabric ; performance ; throughput

1. Introduction

Blockchain, originating from Bitcoin [1], encompasses a list of continuously growing data and transaction records, called blocks that are cryptographically linked and secured. Peers maintain the blockchain in a peer-to-peer (P2P) transaction platform, where transactions are recorded in a period of time and packaged into a block by peers to join the blockchain ledger. Blockchain offers a decentralized network with records being tamper-resistant and traceable. Numerous blockchain-based decentralized applications have emerged with the widespread development of this technology. The emergence of Internet of Things (IoT) technology is making large-scale sensor deployment possible at an unprecedented scale for a variety of applications like air quality monitoring [2], healthcare [3], smart homes and smart buildings [4], agriculture [5] and many industrial applications [6]. In all such applications, the typical IoTs are the connected smart devices usually comprising one or more sensors, data gathering and processing controllers with memory, and air or wired interface to the communication channels. As a secure and unalterable architecture, blockchain is a promising paradigm to address the needs of availability, confidentiality, and integrity for IoT applications [7]. The integration of the blockchain to the Internet of Things (IoT) is a challenging enrichment that can guarantee the privacy, security, trust, and data reliability of conventional IoT applications. The feasibility of such blockchain-based IoT systems has been extensively explored recently [8][9]. Nonetheless, the time-consuming consensus process is the primary bottleneck of adopting blockchain technology in IoT applications. The IoT systems are varied in terms of the number of generated requests where applications generate thousands of transactions per second (tps).

Distributed ledger technologies (DLTs) enable the storage of information securely and accurately using a set of cryptographic primitives. Once stored, the information becomes immutable. Hyperledger Fabric (HLF) is a form of a permissioned DLT. It helps enterprises to build their specific DLT solutions more efficiently and securely. The HLF system performance is enhanced by implementing a highly modular framework and pluggable consensus. It can also provide privacy for a broad range of implementation solutions (e.g., IoT networks) while meeting the specific needs of IoT applications. Furthermore, a pluggable consensus approach improves the latency of finality and confirmation. Achieving scalability, throughput, robust cryptographic security, latency, and resource consumption are some of the major challenges while moving from traditional DLT to HLF solutions. The security arrangements of distributed IoT systems can be established and maintained by deploying HLF. HLF offers the implementation of restricted networks and controlled access to user data within the IoT systems.

Permissioned blockchains are those that run a blockchain among a group of known and identifiable members. A permissioned blockchain secures transactions between a set of organizations that have a common aim but do not completely trust each other, such as firms that exchange payments, commodities, or information. A permissioned blockchain can employ classic Byzantine-fault-tolerant (BFT) consensus by relying on the identities of the peers. HLF is the first distributed open-source operating system for deploying permissioned blockchains. Blockchain applications have gained significant attention from both the industry and academia in recent years [10][11]. Such applications are noticeable in different domains ranging from public services [12], finance [13], smart hospitals [14], smart manufacturing [15], supply chains

[16], energy trading [17], etc., to the new era of IoT [18]. Despite the development and implementation of many blockchain projects, there are still concerns associated with the blockchain platforms' throughput, latency, and ability to scale [19].

Performance evaluation presents a significant challenge for current blockchain systems [20][21], particularly during the execution of complex smart contracts. Technical challenges in adopting blockchain systems are associated with parameters such as throughput, latency, scalability, size, bandwidth, security, wasted resources, usability, as well as versioning, and hard forks [22]. Therefore, it is crucial to evaluate the real-time performance of the blockchain platforms. The performance of the blockchain system can be considered overall and as a detailed performance. The overall performance, including the throughput and latency, can help to find the ideal blockchain system that could fit real-world application scenarios. However, the detailed performance computation reveals performance bottlenecks and provides detailed information about the entire process. Blockchain parameters affect the performance, security, and adaptability of the system. This becomes more complicated when choosing an optimal configuration in the IoT systems with vast amounts of small and resource-constraint devices. The parameters need to be validated and tested before deploying the blockchain IoT systems to determine the limitations and possible bottlenecks.

The evaluation of the overall performance of blockchain systems has been studied widely in the literature [23][24][25][26]. However, various process stages still need more detailed performance measurements. There is a lack of metrics to measure and monitor the detailed performance of blockchain systems. Moreover, the scalability and real-time monitoring overhead of the framework need to be comprehensively studied. It is important to be able to monitor the system performance in a large-scale distributed environment. Thus, the way of performance monitoring and the selection of metrics for doing it are the main challenges for the blockchain performance measurements.

HLF network is orchestrated by various components, including endorsers, ordering services, and committers. It constitutes different transaction processing phases consisting of the endorsement, ordering, validation, and commit phases. Therefore, HLF encompasses various configurable parameters such as block size, channels, endorsement policy, and state databases. Finding the right set of values for these parameters is the main challenge in adapting an efficient blockchain system. A comprehensive performance analysis needs to find out an optimal block size to achieve higher throughput and lower latency while considering a more efficient type of endorsement policy in a distributed platform.

2. Configuration Parameters and Key Metrics

The primary focus is on studying the overall performance from the peer's perspective. At the same time, the Orderer and Gossip effects on the experiment were eliminated as they were kept static. The overall system under the test (SUT) and related components are presented in **Figure 1**. The model includes a single-channel HLF network with one client running benchmarking tools and one anchor peer.

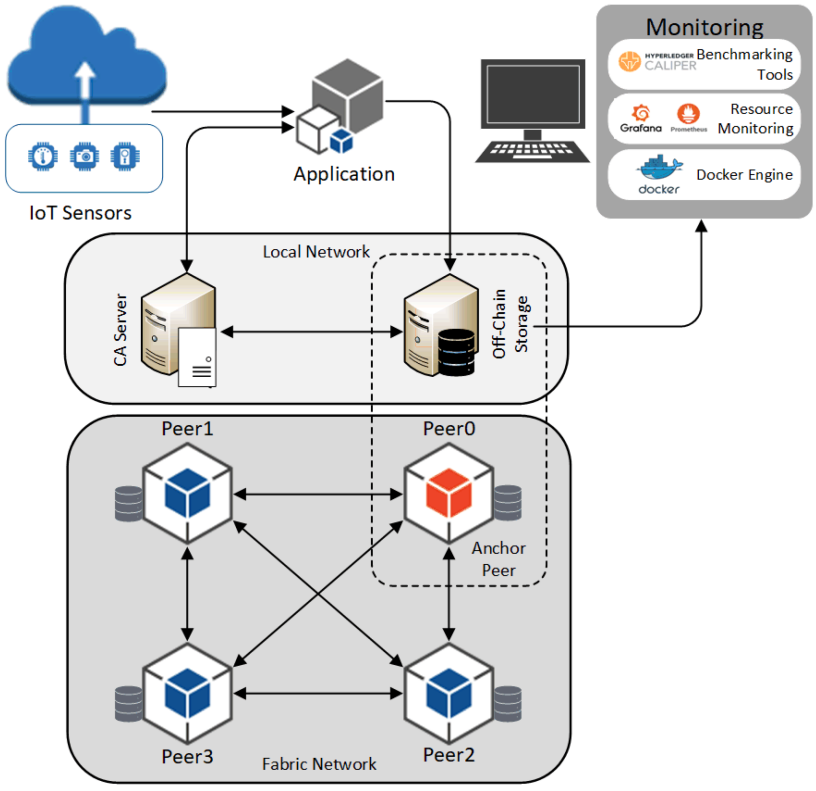

Figure 1. HLF-based distributed system model (the peers and off-chain storage are separated into a scalable platform).

2.1. Key Parameters Definition

Several key parameters are considered in this study. The first of them is the block size. An Orderer orchestrates transactions in batches. It then delivers them to peers in a block with the aid of the Gossip protocol. Each peer processes one block of received transactions at a time. The Orderer performs the cryptographical process per block to verify the Orderer signature, while the endorsement signature verification process is handled per transaction. The block size variation influences the throughput and latency. Therefore, this study investigates the effect of various block sizes in conjunction with transaction sending rates. Note that it is assumed here that all transactions are of the same complexity and are independent of each other.

Endorsement policies play a vital role in controlling the number of executions of a transaction and signing the transactions before submission to the Orderer. So, the transaction can successfully be validated by the VSCC phase. The validation confirms that transaction endorsements meet the endorsement policy for that Chaincode (i.e., read/write set does not conflict with simultaneous updates that were committed before.) The time required for the endorsement policy to collect and evaluate transactions is affected by its complexity.

Channel provides an environment where a group of peers creates a separate transaction ledger accessible only by members. However, a peer can join multiple channels and therefore maintain various ledgers. The channels process orders and delivers transactions independently (even though on the same peers). The number of the channels and their functionality directly impact system performance and scalability.

The routine verification process and signature computation by peers as a part of the system Chaincodes need significant CPU and network resources. Running user-defined Chaincodes by endorsing peers during transaction submissions creates extra loads on the system. In the presented case, the design considers a network having low latency and high bandwidth.

2.2. Performance Metrics

The Hyperledger Performance and Scalability Working Group developed a document [27] providing precise performance metrics applicable across various DLT platforms. It was used in the reported experiments and analysis.

2.2.1. Transaction Throughput

Deployment, execution, and invoking of smart contracts in different blockchain systems occur at different speeds. It is needed therefore to monitor the transaction throughput. It is measured as the rate of committing valid transactions by the HLF network in a defined period. For the HLF network with a single channel, the measurement at a single peer is considered. However, in the reported model and analysis the experiments were further extended to multiple peers (up to 100). The formal mathematical description of the transaction throughput can be obtained as:

$$TPS_i = \frac{Count\ (T_x\ in\ (t_s, t_e))}{t_e - t_s} \quad (1)$$

where, *T*x is the total number of submitted transactions, *t*e is the last block commit time, and *t*s is the initial transaction submission time. The transaction throughput of *N* peers is calculated by taking the median:

$$\overline{TPS} = \frac{\sum_i TPS_i}{N} \quad (2)$$

2.2.2. Transaction Latency

When the transaction is sent to the network, it takes some time to be confirmed by the system. Transaction latency is the amount of time taken from the point the transaction is submitted to the point when the transaction is confirmed and committed with the result being available across the network. This metric is measured per transaction. However, in most cases, the experiment provides various statistics on overall transactions such as high, average, low, and standard deviations. In the reported analysis, the transaction confirmations at a single peer and multiple peers with various load levels were checked. The computed end-to-end latency consists of three components: endorsement latency, ordering

latency, and commit latency [28]. During a period started at *t*s and ended at *t*e, the transaction sent to the peer is shown by *Tx*input, and *Tx*confrimed. The average latency of the peer *i* can be computed using the following equation:

$$AL_i = \frac{\sum_{T_x} (t_{Txconfirmed} - t_{Txinput})}{Count\ (T_x\ \ in\ \ (t_s, t_e))} \tag{3}$$

The latency of all smart contracts is calculated by taking the median:

$$\overline{AL} = \frac{\sum_i AL_i}{N} \tag{4}$$

2.2.3. Network Size and Scalability

The implemented HLF network's ability to support increasing the number of participants is computed in this study. Network size indicates the number of validating peers participating in consensus in the SUT. Network size is presented to show the total number of nodes actively participating in the HLF blockchain network.

2.2.4. Block Size

Block size presents the number of transactions per block, and it is described by three variables: the maximum transaction count, absolute maximum byte, and preferred maximum bytes. The transactions are batched as a block. The study further expands the analysis to include multiple blocks (10 blocks and 50 blocks) in batches. It also studies the effects of different batch sizes on HLF systems.

2.3. Test Environment

The primary goal of this study is to benchmark the performance of the distributed HLF implemented on multiple machines. Therefore, an in-depth study of HLF core components and benchmark performance for IoT applications was conducted.

Performance assessment and scalability evaluation of HLF were conducted by deploying different sets of parameters including transaction sending rate, block size, size of the network, and network traffic delivery. To perform the performance evaluation, various metrics have been considered such as network throughput, average transaction latency, and resources. Scalability was measured based on variations in throughput and transaction latency by increasing the size of the network. The test results show the impact of a specific parameter on the performance of the HLF blockchain network, discover the bottlenecks, and illustrate how the adjustments can be deployed to enhance the performance.

Figure 2 depicts the experimental setup model that was used in all experiments. A permissioned HLF network was set up with one organization that includes several peers in each scenario. The ordering service was run on a separate node, and a single channel was implemented. The Chaincode was deployed on the channel to facilitate the assigned tasks.

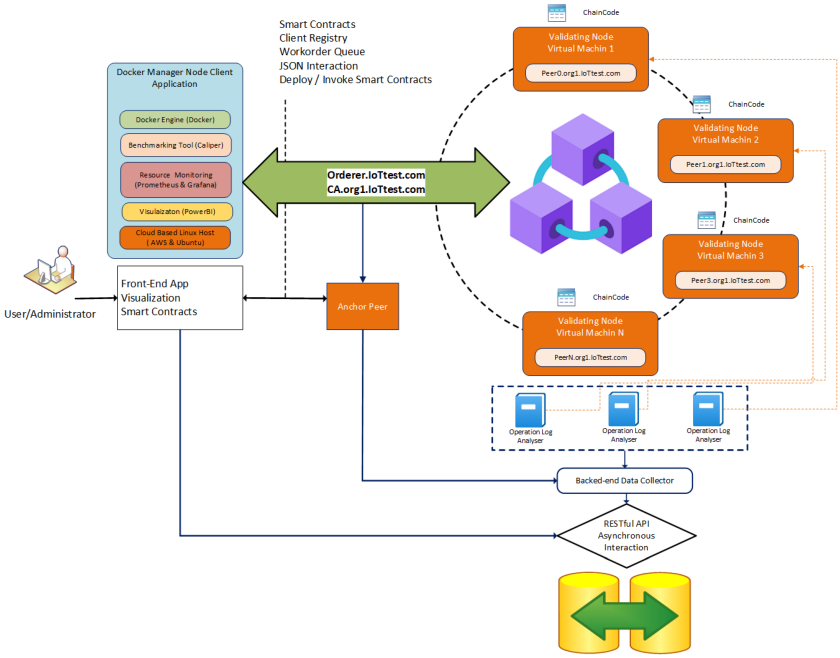

Figure 2. Experimental setup and components for performance evaluation.

The setup deployed a private HLF blockchain network in a controlled distributed environment. To achieve realistic results, several Amazon AWS EC2 instances were deployed as an underlying network of nodes. Their parameters are given in **Table 1**. Each instance was run on its Virtual Machine (VM). All VMs belonged to the same subnetwork to diminish the effect of network latencies within the experiments. The same investigation was conducted several times with different values of peers and nodes. KV denotes a Key Value to be sent as a transaction to the blockchain network.

Table 1. SUT parameters and metrics.

Parameters	Values
Transactions	**1 KV write (1-w) of size 20 bytes**
Channels	**1 Channel**
World StateDB	**LevelDB**
Peer Resources	**Up to 100 vCPUs, 3.3 GHz, 10 GiB, Low to Moderate Network Performance**
Block Size	**30 transactions per block**
Batch Timeout	**1000 ms**
Tx Sending Rate	**5–500 (tps)**
Number of Blocks	**10, 50**

IoT gateways were modelled as EC2 instances in AWS. Various message transactions were implemented within the IoT system as blockchain transactions. AWS EC2 instance having 2vCPUs, 3.0 GHz Intel Xeon Platinum processors and 4GB RAM was used to run the test benchmark platform. The AWS EC2 instance ran Ubuntu 18.04 LTS and peers, CA, OS, and Caliper with Hyperledger Fabric release v1.4. That test environment was used to investigate the impact of the hardware selection (i.e., CPU and RAM) on the throughput, latency, and scalability of the implemented blockchain network.

Virtual machines as IoT edge (with the exception of the HLF system) aid in reducing traffic load interference generated by numerous systems. The Docker running on VMs (Virtual Machine) are spread over many computers. The number of allowed computers was restricted, even when some Docker systems were installed on a host since a high number of CLIs were necessary for issuing transactions. Each VM had dedicated resources for processing transactions.

The Hyperledger Fabric (version 1.4) framework was deployed to run the blockchain application. It is an open-source permissioned blockchain platform for enterprise applications. Virtual machine instances host Hyperledger Caliper [27], a benchmark tool to measure multiple blockchain performances. The Caliper also runs on client and monitoring instances to broadcast transactions on the HLF channel. The network consisted of numerous peers (from 5 peers per organization and up to a maximum of 100 peers) that were run on scalable network infrastructure. The blockchain components were deployed as a Docker container. Docker Swarm was used to orchestrate and manage the containers spread across the network of VMs. All nodes had the Ubuntu 18.04 LTS operating system.

The Hyperledger Caliper was deployed as a standard open-source benchmarking tool recommended by the Hyperledger community. Further analysis of collected log files and data was conducted using Microsoft PowerBI [29] and Origin Pro [30]. The Prometheus and Grafana [31] were used to monitor Hyperledger Fabric Docker Containers.

The Proof of Work (PoW) consensus shows its robustness. Due to its pseudoanonymous nature, it has been considered the most secure option for cryptocurrency applications. However, in the enterprise ecosystems such as IoT networks and telecom environments, it appeared to be redundant as the blockchain participants are already known to each other. Therefore, permissioned blockchains are designed for enterprise systems that use more straightforward and less resource-consuming consensus protocols, such as Raft [28], that was implemented in this study.

To evaluate the performance of SUT, the following approach is employed. Transactions by the client application are submitted. They change the World State of an HLF network. It leads to testing the performance of consensus. When the application client delivers a transaction proposal to the endorsing peers, this step begins. The endorsement policy determines which endorsing peers are chosen. It is worth noting that the ledger operation (required to activate consensus mechanisms) influences an HLF network's performance. Therefore, simple transactions have to be implemented to

minimize this performance impact. Based on the system time set in Chaincode, each transaction creates a number, which is appended to a value. Each of those values generates a key–value pair including a constant value. The Chaincode is used to write the transactions to the ledger. It runs in a secure and segregated docker container and allows peers to create the transactions. Every transaction is a write transaction that modifies the World State. The endorsers then execute the Chaincode separately, construct a transaction response depending on the execution results, and sign the response. Finally, the application receives the signed transaction proposal response. Because the key's randomization makes it difficult to write a key that has already existed in the ledger, the transactions are secured against being invalid.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical Report. 2019. Available online: https://bitcoin.org/en/bitcoin-paper (accessed on 10 November 2020).
2. Ali, S.; Glass, T.; Parr, B.; Potgieter, J.; Alam, F. Low cost sensor with IoT LoRaWAN connectivity and machine learning-based calibration for air pollution monitoring. IEEE Trans. Instrum. Meas. 2020, 70, 1–11.
3. Dwivedi, A.D.; Srivastava, G.; Dhar, S.; Singh, R. A decentralized privacy-preserving healthcare blockchain for IoT. Sensors 2019, 19, 326.
4. Ghayvat, H.; Mukhopadhyay, S.; Gui, X.; Suryadevara, N. WSN-and IOT-based smart homes and their extension to smart buildings. Sensors 2015, 15, 10350–10379.
5. Jayaraman, P.P.; Yavari, A.; Georgakopoulos, D.; Morshed, A.; Zaslavsky, A. Internet of things platform for smart farming: Experiences and lessons learnt. Sensors 2016, 16, 1884.
6. Sanchez-Iborra, R.; Cano, M.D. State of the art in LP-WAN solutions for industrial IoT services. Sensors 2016, 16, 708.
7. Boudguiga, A.; Bouzerna, N.; Granboulan, L.; Olivereau, A.; Quesnel, F.; Roger, A.; Sirdey, R. Towards better availability and accountability for iot updates by means of a blockchain. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, France, 26–28 April 2017; pp. 50–58.
8. Honar Pajooh, H.; Rashid, M.; Alam, F.; Demidenko, S. Hyperledger Fabric Blockchain for Securing the Edge Internet of Things. Sensors 2021, 21, 359.
9. Honar Pajooh, H.; Rashid, M.; Alam, F.; Demidenko, S. Multi-layer blockchain-based security architecture for internet of things. Sensors 2021, 21, 772.
10. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. Future Gener. Comput. Syst. 2020, 107, 841–853.
11. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. Int. J. Web Grid Serv. 2018, 14, 352–375.
12. Chettri, L.; Bera, R. A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. IEEE Internet Things J. 2019, 7, 16–32.
13. Guo, Y.; Liang, C. Blockchain application and outlook in the banking industry. Financ. Innov. 2016, 2, 24.
14. Jamil, F.; Ahmad, S.; Iqbal, N.; Kim, D.H. Towards a remote monitoring of patient vital signs based on IoT-based blockchain integrity management platforms in smart hospitals. Sensors 2020, 20, 2195.
15. Shahbazi, Z.; Byun, Y.C. Integration of Blockchain, IoT and Machine Learning for Multistage Quality Control and Enhancing Security in Smart Manufacturing. Sensors 2021, 21, 1467.
16. Shahbazi, Z.; Byun, Y.C. A Procedure for Tracing Supply Chains for Perishable Food Based on Blockchain, Machine Learning and Fuzzy Logic. Electronics 2021, 10, 41.
17. Jamil, F.; Iqbal, N.; Ahmad, S.; Kim, D. Peer-to-Peer Energy Trading Mechanism based on Blockchain and Machine Learning for Sustainable Electrical Power Supply in Smart Grid. IEEE Access 2021, 9, 39193–39217.
18. Christidis, K.; Devetsikiotis, M. Blockchains and smart contracts for the internet of things. IEEE Access 2016, 4, 2292–2303.
19. Pilkington, M. Blockchain technology: Principles and applications. In Research Handbook on Digital Transformations; Edward Elgar Publishing: Cheltenham, UK, 2016.
20. Dannen, C. Introducing Ethereum and Solidity; Springer: New York, NY, USA, 2017; Volume 318.
21. Hyperledger Fabric. Available online: http://hyperledger-fabric.readthedocs.io/en/release-1.4/ (accessed on 5 January 2021).

22. Swan, M. Blockchain: Blueprint for a New Economy; O’Reilly Media, Inc.: Newton, MA, USA, 2015.
23. Dinh, T.T.A.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.L. Blockbench: A framework for analyzing private blockchain s. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2 017; pp. 1085–1100.
24. Fan, C.; Ghaemi, S.; Khazaei, H.; Musilek, P. Performance Evaluation of Blockchain Systems: A Systematic Survey. IE EE Access 2020, 8, 126927–126950.
25. Dabbagh, M.; Choo, K.K.R.; Beheshti, A.; Tahir, M.; Safa, N.S. A survey of empirical performance evaluation of permissi oned blockchain platforms: Challenges and opportunities. Comput. Secur. 2021, 100, 102078.
26. Dabbagh, M.; Kakavand, M.; Tahir, M.; Amphawan, A. Performance Analysis of Blockchain Platforms: Empirical Evaluat ion of Hyperledger Fabric and Ethereum. In Proceedings of the 2020 IEEE 2nd International Conference on Artificial Int elligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 26–27 September 2020; pp. 1–6.
27. Hyperledger Performance and Scale Working Group. Hyperledger Blockchain Performance Metrics. Available online: ht tps://www.hyperledger.org/wpcontent/uploads/2018/10/HL-Whitepaper-Metrics-PDF-V1 (accessed on 10 November 20 20).
28. Ongaro, D.; Ousterhout, J. In search of an understandable consensus algorithm. In Proceedings of the 2014 USENIX A nnual Technical Conference, Berkeley, CA, USA, 19–20 June 2014; pp. 305–319.
29. Microsoft Power Platform; Microsoft: Redmond, WA, USA, 2020.
30. OriginPro. Available online: https://www.originlab.com/ (accessed on 30 January 2021).
31. Reinartz, F.; Volz, J.; Rabenstein, B. Prometheus–Monitoring System & Time Series Database. 2022. Available online: http://prometheus.io (accessed on 10 November 2020).
