An Extensible Monitoring Tool for Blockchain-Based Software Applications

Abstract—**The use of support tools that monitor software application processes is required to identify, measure, and assess their performance and resolve any anomalies that impede their full functionality. Similarly, applications that use Blockchain technology require analysis and monitoring tools for the resources and functionality provided by their smart contracts. This paper presents a computational tool that monitors hardware resources and evaluates transaction processing requests in Blockchain networks. Called MonitorChain, the solution presented here has software components that allow the connection with different Blockchain networks, the configuration of the workload, the monitoring of consumed hardware resources, and the evaluation of requests processed over Blockchain networks. To evaluate the operation of MonitorChain, we developed a Rest API that uses the components specified in the proposed solution and processed a set of transactions on different Blockchain networks to analyze the logs of hardware resource metrics and client-server requests. The results indicated that MonitorChain can be used by other software applications, and the reports generated are useful for understanding the resources needed to run software systems on Blockchain networks.**

Index Terms—**Blockchain, Smart Contracts, performance, monitoring.**

I. Introduction

The software industry has created computer systems that use Blockchain technology in several areas, including the Internet of Things, streaming services, the health sector, and public safety [1], [2]. Blockchain technology, viewed as disruptive, has transformed traditional business models due to its ability to provide security in data transactions, transparency and traceability in contracts, and cost reduction by eliminating a third party to mediate transactions between the stakeholders involved.

Blockchain technology is used in the development of computer systems via decentralized applications (DApps). DApps are open source, run on a peer-to-peer network, and use smart contracts to interact with a Blockchain network [3]. A key feature of a decentralized application is that it executes transactions autonomously and without the involvement of central mediators [4]. In this case, smart contracts provide this feature by directly managing rules on the Blockchain network.

Blockchain technology in software applications requires significant processing power of computational resources and high availability of Internet traffic. Furthermore, the increased electricity consumption in transaction mining and the high carbon emission during this process are highlighted as impending challenges that necessitate advances in the state of practice. Recent research has centered on developing computational solutions to assist stakeholders in determining the resources required to run a software application on Blockchain networks [5]. In the state of the practice, solutions such as Hyperledge Benchmark [6], Distributed Ledger Performance Scan (DLPS) [7], and BlockBench [8], support different consensus protocols and offer custom testing and comprehensive performance measurement of Blockchain applications. Computational tools for this purpose enable evaluating the processing performance and hardware resource consumption when a set of instructions is executed over a Blockchain network. As a result, processing failures and bottlenecks can be identified, and architectural changes made to the evaluated software application. However, the solution proposed in this paper addresses additional features that have yet to be implemented by the tools in the state of practice, such as interface extensibility for connecting to different Blockchain networks and calculating Gas units used in processing Blockchain transactions.

This paper presents a computational tool that monitors hardware resources and evaluates transaction processing requests in Blockchain networks. Called MonitorChain, the solution presented here has software components that allow the connection with different Blockchain networks, the configuration of the workload, the monitoring of consumed hardware resources, and the evaluation of requests processed over Blockchain networks. In addition, MonitorChain provides additional graphical visualization resources for the functionalities described above. To evaluate the operation of MonitorChain, we developed a Rest API that used the components specified in the proposed solution and processed a set of transactions on different Blockchain networks to analyze the logs of hardware resource metrics and client-server requests. The results indicated that MonitorChain could be used by other software applications, and the reports generated help understand the resources needed to run software systems on Blockchain networks.

II. Background

A. Blockchain Technology

In recent years, Blockchain technology has become one of the most impactful innovations on the technology scene. Originating as the underlying infrastructure of the Bitcoin cryptocurrency, Blockchain technology quickly spread beyond cryptocurrencies and found application in various areas, such as the Internet of Things (IoT), streaming services, healthcare, and public safety [9], [10]. A distinctive feature of Blockchain technology is its ability to create decentralized, immutable, and transparent records of transactions, which are stored in a distributed network of nodes. Thus, blockchain technology provides an additional layer of security and trust compared to traditional systems of records and transactions.

Blockchain is a distributed ledger technology based on a decentralized consensus system to create a shared, secure, immutable record of transactions or events [11]. This architecture allows information to be recorded in blocks, which are linked sequentially in a chain and maintained by a decentralized network of validation nodes. Blockchain's underlying mechanism makes it resistant to tampering with records since any attempt to modify data requires the agreement of the majority of nodes in the network. This immutability and transparency of data, together with eliminating intermediaries in validation processes, make Blockchain a standout technology, with wide application in various areas, from finance to logistics, and the potential to profoundly transform recording and transaction practices across the entire spectrum of human activity.

In Blockchain systems that use the Ethereum Virtual Machine (EVM) to execute smart contracts [12], the "Gas unit" represents the primary cost indicator in the network. Gas is an abstract measure that quantifies the computational effort required to carry out operations on smart contracts. This unit is crucial for understanding and regulating transactions and is paid in Ether based on the exchange rate called gas price.

An essential subdivision within Gas is the gwei, a smaller unit of Ether used to express Gas prices. 1 gwei is equivalent to 0.000000001 Ether, offering a finer granularity in transactions, especially useful in more complex smart contracts that require more detailed precision in measuring costs. In mining operations, miners have the autonomy to ignore transactions with gas prices considered too low, directly influencing the inclusion or exclusion of transactions in blocks. In the context of public blockchains, such as Ethereum, Gas is also used by miners to evaluate the performance and resource consumption of smart contracts, providing a fundamental metric for the economy and efficient operation of the Blockchain network.

B. Monitoring applications

Monitoring systems play a crucial role in analyzing and continuously improving the performance characteristics of computer systems. Monitoring systems are designed to collect, record, and analyze performance data and operational metrics from running applications [13]. These systems allow real-time visibility of the system's state, essential for identifying problems, performance bottlenecks, and anomalies that may arise during operation. In this sense, monitoring systems differ from benchmarks [14], which are developed as a set of metrics and tests designed to evaluate computer systems' performance in a wide variety of scenarios and workloads. Thus, they are used in benchmarking environments based on static performance metrics. In contrast, monitoring systems are essential for the dynamic and real-time observation of system behavior in a production environment.

The continuous and high-quality operation of an application is critical for meeting the needs of its users. The use of support tools that provide monitoring of the processes of these applications is required to identify, measure, and assess their performance, as well as to resolve any anomalies that impede their full functionality. Similarly, applications that use Blockchain technology require analysis and monitoring tools for the resources and functionality provided by their smart contracts.

Blockchain networks, both public and private, enable organizations to manage their supply chains, contracts, payments, identity, and resources in novel ways. Blockchain technology-enabled changes in business models generate significant monetary value in daily transactions. In Blockchain applications, tracking these transactions, ensuring their validity, and monitoring the consumption of computational resources to execute these processes is critical. To contribute to this vital research field, this paper presents a computational tool enabling DApps to monitor hardware resources, evaluate client-server requests, provide interfaces for connections to different Blockchain networks, and perform workloads to execute smart contract functionalities.

III. Related Works

This section presents studies on using Blockchain technology and monitoring tools in different application contexts. Blockchain technology has been applied in different application scenarios, and its characteristics allow the consumption of information necessary for the operation of the solutions found in the state of the art. In [15], an event-driven software architecture that includes a messaging structure is designed for managing the operations of collecting and changing states in a blockchain network. In turn, [16] proposes a service-oriented architecture (SOA) that integrates edge computing technology into a streaming platform. SOA architecture and edge computing enable higher quality and more efficient data processing and transmission.

In the context of streaming platforms, researchers such as [17] specify a structure that uses edge computing, in which the block size in the network is adaptable for video streaming applications. In this same line of research, [18], [19] propose the development of a decentralized streaming platform that makes use of the InterPlanetary File System (IPFS) protocol and employs a cryptographic reward incentive system for content creators based on the popularity of their videos.

Research into runtime monitoring tools has found applications in a variety of domains. Due to the plurality of existing approaches, [13] specified a comparison framework to mitigate the complexity associated with selecting such tools. This framework is based on an analysis of existing literature and taxonomies applicable to monitoring languages and standards. With the advent of rapid technological developments, new monitoring tools have emerged that can respond immediately to demands for real-time analysis. In this context, runtime software monitoring (SRM) tools have gained prominence, which ensures the quality, performance, and reliability of complex software systems. The study by [20] provides a comprehensive and detailed overview of SRM, covering its chronology and classification and the metrics that allow the performance of such tools.

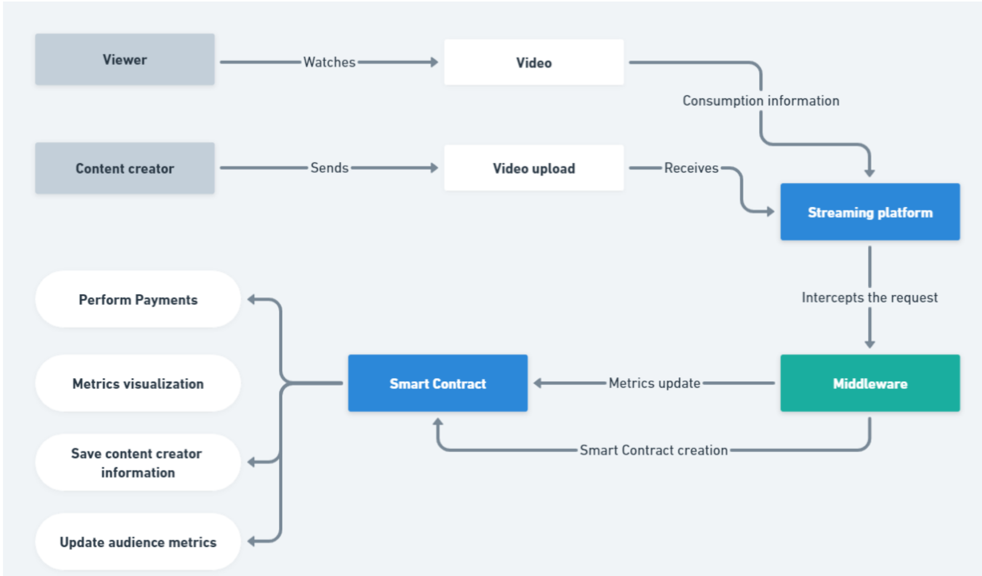

Fig. 1. MonitorChain Evaluation scenario.

In this category of work, the research of [20] identifies and describes two predominant types of SRM: the first, known as code-based, focuses on analyzing the source code of a system, seeking to identify potential problems, vulnerabilities, and inefficiencies, which allows developers to take proactive measures to improve the quality and security of the software. The second category, called execution-based, takes a more dynamic approach, monitoring the system's behavior in real-time during its execution. This approach allows for detecting real problems that may only arise under specific operating conditions, providing a complete overview of system performance in production environments. In [21], the authors address the challenges inherent in monitoring and debugging distributed programs in real-time. These challenges are mainly due to the specific characteristics of operating environments in distributed systems, which can present considerable complexities, making monitoring and debugging solutions complex and challenging. In the Blockchain application ecosystem, [22] presents a comprehensive analysis of the security-related challenges that impact blockchain systems. The study addresses a variety of attacks targeting blockchain systems, encompassing those that target the network layer, the consensus layer, and the smart contract layer. The authors highlight the inherent complexity of security in blockchain systems but maintain that implementing monitoring systems can substantially mitigate security-related risks in this context.

A performance monitoring framework for Blockchain systems is proposed in [23]. The authors specified three main components for data collection, processing, and performance visualization. The results showed that the framework identifies performance bottlenecks and optimizes the performance of software applications.

The research mentioned in this section represents an advancement in software application monitoring. However, our solution advances the state of the art by providing a tool that monitors software systems in a flexible architectural model that allows the use of different Blockchain networks. Section IV presents the software architecture and main functionalities developed to support software applications based on Blockchain technology.

IV. Proposed Solution

MonitorChain is a computational tool that monitors hardware resources and evaluates transaction processing requests in Blockchain networks. Designed to allow the evaluation of different Blockchain networks available in the market, MonitorChain uses an architectural pattern that helps extend or develop new components without compromising the operation of the application. Figure 1 illustrates the software architecture of the proposed solution, and the definition of its four main components is given below.

Hardware Resource Monitoring calculates the RAM usage, processing stack, and processor usage metrics of the computing resource running the proposed evaluation system. The values discovered for the metrics above are updated every second, and the results are written to a log file. The client-server request monitoring component can evaluate HTTP or HTTPS transactions. As a result of request monitoring, the following information is measured and written to a log file: the address of the route used in the request, the response status codes (HTTP or HTTPS), the response time in milliseconds, the total number of successful and failed requests, the number of gas units used in transactions, cryptocurrency used in transactions and the estimated cost in dollars made in transactions.

Estimating the cost (EC) of Blockchain transactions is determined by valuing the cryptocurrency used. This process involves taking into account the price of the gas unit in Gwei (GP), multiplied by the total gas used in the transaction (GE)

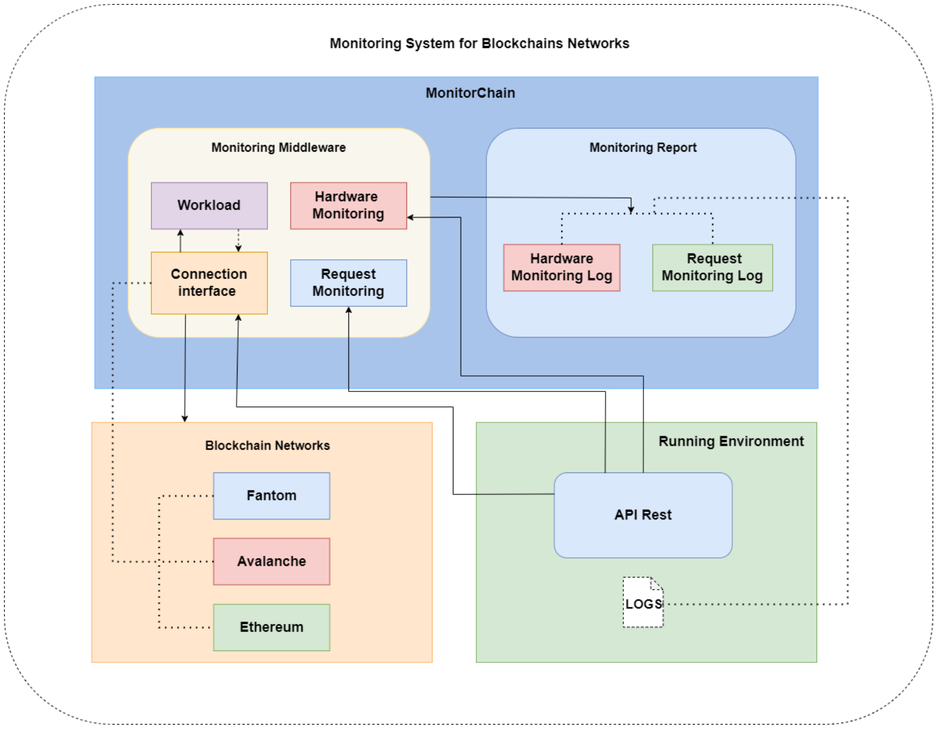

Fig. 2. MonitorChain Software Architecture

associated with the selected digital currency, and multiplying it by the cryptocurrency's exchange rate in dollars (ER). The following equation explains the mathematical equation used for this calculation.

$$EC = \left(\left(\frac{GP}{1{,}000{,}000{,}000}\right) \times GE\right) \times ER$$

To allow the connection with different Blockchain networks, a component that manages communication interfaces with the main DApps platforms available in the market was developed. The interfaces available in MonitorChain have connection parameters, persistence method signatures, and subscription contracts for business layer classes of software applications that need to communicate with different Blockchain networks. MonitorChain already has communication interfaces with Fantom[1], Avalanche,[2] and Ethereum[3] networks, and if needed, new communication interfaces with other networks can be developed following the pattern implemented in the proposed architecture. Finally, the workload component allows the configuration of the data load that will be processed over the Blockchain networks available in MonitorChain.

The computational solution proposed in this paper was created by combining the NodeJs cross-platform environment and the JavaScript programming language. A Rest API was created to use the components described in the software architecture and analyze the processing logs of hardware resource metrics and client-server requests processed over Blockchain networks for evaluating the operation of MonitorChain.

To use MonitorChain, software applications must import the tool's files and call the functions they wish to use in their application. The monitoring functionalities work as middleware and must be instantiated in the application's main source code file. Once the application is configured, the connection interfaces to the Blockchain networks available in MonitorChain can be used or extended to new networks. Finally, a file (e.g., *.ts) containing the workload execution instructions must be configured for processing on the chosen Blockchain network using the routes available in the application.

V. Evaluation

As mentioned in the previous section, a software application was developed to evaluate the solution proposed in this work, which uses blockchain technology to analyze how a traditional business model can benefit from disruptive technology. The evaluation scenario used in this study refers to using blockchain technology in the streaming platform ecosystem, including transferring financial resources between two digital wallets and offering transparency mechanisms between the stakeholders involved.

The computational solution developed in this evaluation scenario preserves the basic functionalities of streaming services (e.g., content creation, visualization, advertisements). It proposes an intermediate layer to make audience control and payments between the parties involved transparent. This intermediary component plays a key role in updating the

[1]https://fantom.foundation/
[2]https://www.avax.network/
[3]https://ethereum.org/pt-br/

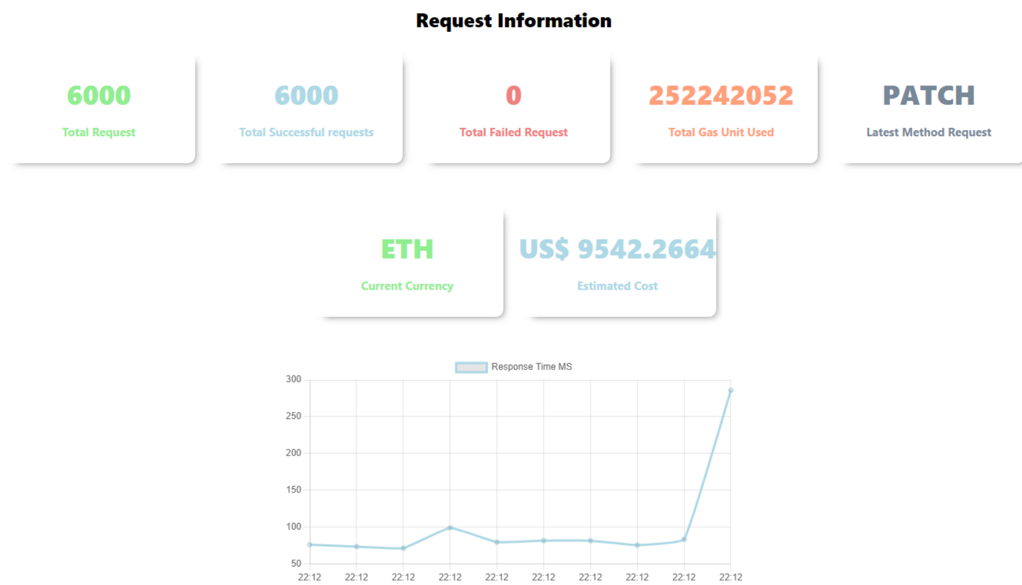

Fig. 3. Softwre Metrics Report

audience information associated with each video on the platform. Each video is accompanied by a specific smart contract, generated at the time of upload, storing audience information and adding the platform's monetary transfer system to the Blockchain environment without compromising the platform's core operation. Figure 1 shows the flow of creating and updating a contract in the environment described.

A feature of the proposed solution is the decentralization of the payment system. In the traditional model adopted by streaming platforms, financial institutions are used as third parties to facilitate payment between the interested parties involved. In the evaluation scenario, the financial institution is removed from the business model, and the entire payment process is carried out by the business rules specified in smart contracts. Based on this evaluation scenario, we integrated MonitorChain with the developed software application and evaluated the software monitoring functionalities that use Blockchain technology. As illustrated in Figure 1, the API represents the blockchain application that we want to investigate: i) the monitoring of hardware resources used during the execution of smart contracts functionalities; ii) the monitoring of requests made to the application; iii) the analysis of the financial cost associated with the processing of transactions on the blockchain network.

The evaluation was conducted in a computing environment with a local area network and composed of workstations with 16 GB of DDR4 RAM and an I7 1165G7 processor. After setting up the computational domain, the following activities were performed: i) the Ethereum network, operated locally through Ganache, was chosen as the primary connecting blockchain network. Furthermore, the Fantom network was utilized remotely as the connecting blockchain network for the second round of testing. ii) three distinct data loads were configured, totaling 10,000 transactions. The data loads were divided in both tests: i) 100 transactions for contract creation, ii) 6000 transactions for contract information updates, and iii) 3900 transactions for contract information access.

During the evaluation, it was possible to monitor the resources consumed during the execution of the data loads using the graphical interface provided by the MonitorChain tool. As shown in Figure 3, the results obtained offer various information regarding the requests made at the end of this process. In the evaluation context, Figure 3 illustrates the results of the 6,000 transactions carried out to update the audience metrics in the contract linked to the video. This data covers not only the total number of requests but also discriminates between successful and failed transactions, as well as shows the amount of gas used in all transactions, the cryptocurrency used, and the estimated costs related to the transactions.

In addition to the metrics already mentioned, it is worth noting that the collection of information on the cost of transactions also included a detailed analysis of the transaction fees associated with each operation and a description of the transactions according to their complexity and priority. This data provided a more comprehensive view of operating costs, enabling the identification of transactions with higher costs and providing input for strategies to optimize them. With this analysis, the evaluation can provide information for informed decision-making and continuous improvement of transaction processes in the monitored environment.

In the Ethereum network infrastructure, using the ETH cryptocurrency as a reference was based on a market price of US$ 1891.49 per unit of ETH, while the gas price was set at 20 gwei. In this context, the average transaction processing cost was calculated at US$ 00.97, resulting in a total cost of

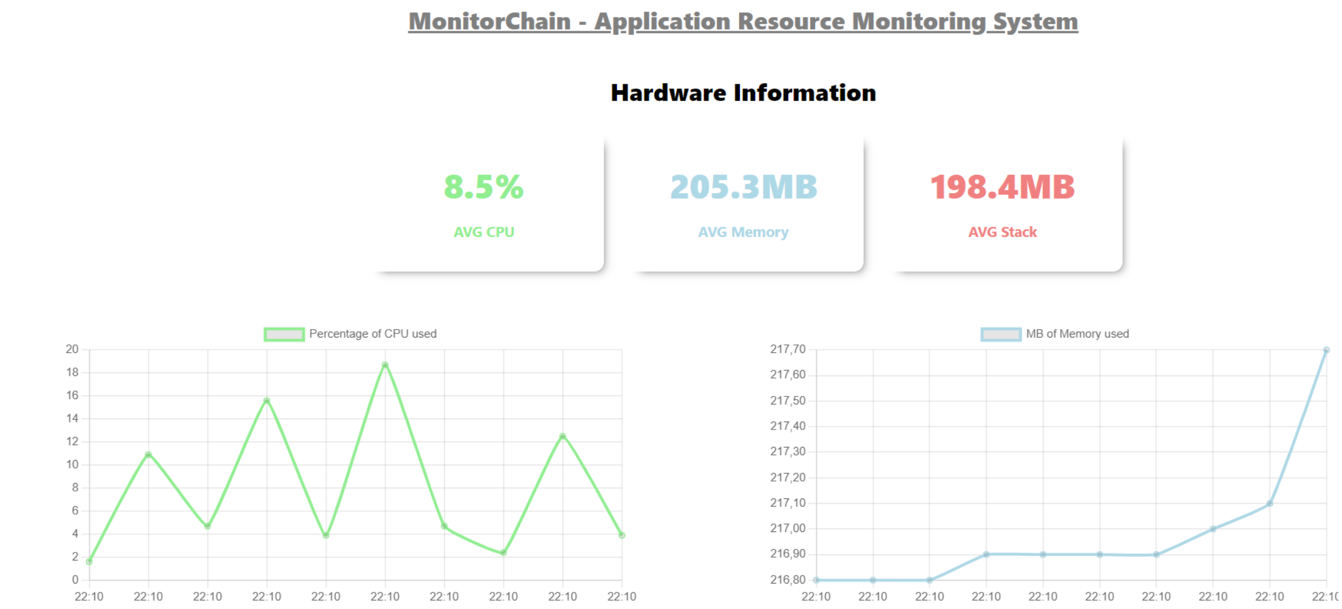

Fig. 4. Hardware Metrics Report

TABLE I
EXECUTION COSTS FOR 10,000 TRANSACTIONS

Network	Cryptocurrency	Market Price (US$)	Gas Price (gwei)	Average Cost per Transaction (US$)	Total Cost (US$)
Ethereum	ETH	1891.490	20	00.97	9742.76
Fantom	FTM	00.32	3.5	0.0001	01.62

US$ 9742.76. On the other hand, in the Fantom network, the reference cryptocurrency was FTM, whose market price was US$ 00.32 per unit of FTM, and the gas price was set at 3.5 gwei. Under these conditions, the average cost of processing transactions was equivalent to US$ 0.0001, resulting in a total cost of US$ 01.62. Similar to the Ethereum network, the price of gas on the Fantom network played an essential role in determining operating costs and is vital to a comprehensive understanding of the financial aspects inherent in executing transactions. The data above on the total cost of executing 10,000 transactions on the two networks is summarized in Table 1, providing a comparative view of the average costs per transaction and total costs on both platforms.

In addition, Figure 4 shows information on the hardware resources used throughout the evaluation process. This data includes average processor usage, RAM usage, and the processing stack. The graphs relevant to this stage illustrate the average use of processor capacity and the average amount of RAM used in the last 10 seconds, providing a more in-depth view of the performance of hardware resources during the execution of operations.

VI. CONCLUSION

This paper presents a computational tool that monitors hardware resources and evaluates transaction processing requests in Blockchain networks. Called MonitorChain, the solution presented here has software components that allow the connection with different Blockchain networks, the configuration of the workload, the monitoring of consumed hardware resources, and the evaluation of requests processed over Blockchain networks. Graphical visualization resources were developed in MonitorChain to facilitate the interpretation of results, and logs can be downloaded after processing execution.

One of MonitorChain's features is its ability to monitor Blockchain applications operating on different networks while allowing support for new networks through an extensible and flexible architecture. In addition to this functionality, MonitorChain offers a set of metrics that make it possible to assess the technical and financial viability of a software application based on Blockchain technology. The results indicated that MonitorChain can be used by other software applications, and the reports generated are useful for understanding the resources needed to run software systems on Blockchain networks.

In future work, we plan to insert new metrics into MonitorChain, such as energy consumption, and develop features responsible for running unit and integration tests on the entire developed tool.

REFERENCES

[1] Soares, R., Araújo, A., Rodrigues, G., & Alencar, C. (2023, April). Streaming Platforms Based on Blockchain Technology: A Business Model Impact Analysis. In International Conference on Information

Technology-New Generations (pp. 143-149). Cham: Springer International Publishing..

[2] Lima, Í., Araújo, A., Souza, R., Couto, H., & Times, V. (2021). A service-based software architecture for enabling the electronic health record storage using Blockchain. International Journal of Computer Applications in Technology, 65(3), 222-234.

[3] Metcalfe, W. (2020). Ethereum, smart contracts, DApps. Blockchain and Crypt Currency, 77.

[4] Johnston, D., Yilmaz, S. O., Kandah, J., Bentenitis, N., Hashemi, F., Gross, R., ... & Mason, S. (2014). TheGeneralTheoryofDecentralizedApplications, DApps.

[5] Saingre, D., Ledoux, T., & Menaud, J. M. (2020, November). BCTMark: a framework for benchmarking blockchain technologies. In 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA) (pp. 1-8). IEEE.

[6] Nakaike, T., Zhang, Q., Ueda, Y., Inagaki, T., & Ohara, M. (2020, May). Hyperledger fabric performance characterization and optimization using goleveldb benchmark. In 2020 IEEE international conference on blockchain and cryptocurrency (ICBC) (pp. 1-9). IEEE.

[7] Sedlmeir, J., Ross, P., Luckow, A., Lockl, J., Miehle, D., & Fridgen, G. (2021). The DLPS: a new framework for benchmarking blockchains.

[8] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K. L. (2017, May). Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM international conference on management of data (pp. 1085-1100).

[9] Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. International journal of web and grid services, 14(4), 352-375..

[10] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K. L. (2017, May). Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM international conference on management of data Couto, H., Araújo, A., Soares, R., & Rodrigues, G. (2022, May). The use of blockchain technology in electronic health record management: an analysis of state of the art and practice. In ITNG 2022 19th International Conference on Information Technology-New Generations (pp. 179-185). Cham: Springer International Publishing.

[11] Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf, 4(2), 15.

[12] Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151 (2014).

[13] Rabiser, R., Guinea, S., Vierhauser, M., Baresi, L., & Grünbacher, P. (2017). A comparison framework for runtime monitoring approaches. Journal of Systems and Software, 125, 309-321.

[14] Bradley, D., & Tyrrell, A. (2002, May). A hardware immune system for benchmark state machine error detection. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) (Vol. 1, pp. 813-818). IEEE.

[15] Wöhrer, M., Zdun, U., & Rinderle-Ma, S. (2021, September). Architecture design of blockchain-based applications. In 2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS) (pp. 173-180). IEEE.

[16] Zhang, H. (2022). Online platform construction of SOA architecture and streaming media technology under edge computing. International Journal of System Assurance Engineering and Management, 13(Suppl 3), 1154-1163.

[17] Liu, M., Yu, F. R., Teng, Y., Leung, V. C., & Song, M. (2018). Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. IEEE Transactions on Wireless Communications, 18(1), 695-708.

[18] Naik, H. M., Vaity, S. A., D'mello, S. N., & Patil, S. D. (2020). DecenTube: A decentralized video streaming platform (dApp) using Ethereum and IPFS. International Journal of Next-Generation Computing, 11(2).

[19] Barua, S., & Talukder, D. (2020, December). A blockchain based decentralized video streaming platform with content protection system. In 2020 23rd International Conference on Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.

[20] Gao, L., Lu, M., Li, L., & Pan, C. (2017, November). A survey of software runtime monitoring. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 308-313). IEEE.

[21] Dodd, P. S., & Ravishankar, C. V. (1992). Monitoring and debugging distributed realtime programs. Software: Practice and Experience, 22(10), 863-877.

[22] Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2020). A survey on the security of blockchain systems. Future generation computer systems, 107, 841-853.

[23] Zheng, P., Zheng, Z., Luo, X., Chen, X., & Liu, X. (2018, May). A detailed and real-time performance monitoring framework for blockchain systems. In Proceedings of the 40th international conference on software engineering: software engineering in practice (pp. 134-143).