# Analysis of an Ethereum Private Blockchain Network Hosted by Virtual Machines Against Internal DoS Attacks

João H. F. Battisti[1], Guilherme P. Koslovski[1], Maurício A. Pillon[1], Charles C. Miers[1](✉), and Nelson M. Gonzalez[2]

[1] Graduate Program in Applied Computing, Santa Catarina State University, Joinville, Brazil
joao.battisti@edu.udesc.br, {guilherme.koslovski,mauricio.pillon,charles.miers}@udesc.br

[2] IBM Watson Research Center, Yorktown Heights, USA
nelson@ibm.com

**Abstract.** The blockchain technology is increasingly adopted by distinct areas of knowledge and institutions filling administrative gaps related to data integrity, and audit procedures. In parallel, the adoption of resource virtualization is a reality to efficiently manage data centers. Consequently, the institutions have integrated blockchain networks into their own systems, hosting blockchain nodes atop virtual machines (VMs). In this sense, we present a resistance analysis of an Ethereum-based network hosted by VMs. The Denial of Service (DoS) attacks are used to identify the impact on the standard VM flavor. Specifically, we developed an environment for carrying out experiments using Ethereum configured with distinct consensus mechanisms: RAFT, Istanbul Byzantine Fault Tolerance (iBFT), and Proof-of-Authority (PoA). Each scenario is in-depth analyzed facing the DoS attack. Our results show that a private blockchain can suffer DoS when configured only to satisfy the default configuration defined by the application.

**Keywords:** Cloud computing · Blockchain · Security · DoS

## 1 Introduction

Blockchain consists of Peer-to-Peer (P2P) networks, cryptography, algorithms and a consensus mechanism [6]. Decentralization requires guarantees for networks to operate correctly. The consensus mechanism is a key component of a blockchain because it determines the organizations involved, how nodes interact, and their roles. Consensus mechanisms are designed for diverse and different contexts. Each application should identify the most appropriate consensus mechanism to meet its requirements while using the computing resources efficiently.

Blockchain requirements are usually identified according to the number of parties involved, types and number of transactions, how trust is managed among

L. Barolli et al. (Eds.): AINA 2022, LNNS 449, pp. 479–490, 2022.
https://doi.org/10.1007/978-3-030-99584-3_42

the parties involved, and what guarantees must be provided. Blockchains used for the general public typically have a significant number of users and transactions. As a result, they usually employ different consensus mechanisms than blockchains used internally by a company or a group of restricted parties. Furthermore, there are several ways to plan and deploy a blockchain. Our work focuses on the private model (or even consortium), in which one or more companies employ blockchain for their own use. Among the existing consensus mechanisms for this model, three stand out: RAFT, iBFT and PoA.

The use of blockchain in applications involves the configuration and deployment of nodes. These nodes allow the inclusion of new entries in the blockchain and implement the consensus process. Institutions adopting the private model create their blockchain nodes in VMs on their clouds [1,11]. The use of VMs implies defining a flavor to run the blockchain nodes. A VM for a blockchain application typically requires a modest configuration (*i.e.*, 2 vCPUS, 4 GB RAM, 20 GB storage, and 1 Gbps of networking) [4,7]. A lean flavor configuration avoids wasting resources and, according to the literature, supports most applications in terms of blockchain transactions. However, if there is an excessive number of transactions, this configuration may become a problem. Memory, processor and network resources can reach operational limits, making it difficult for the consensus engine to function.

In this context, the security of these cloud environments is a complex challenge. Users in the blockchain organization have access to the participating blockchain nodes and/or to the networks in which the VMs are hosted. In this scenario, blockchain nodes deployed in VMs are susceptible to unauthorized access, allowing a user to intentionally or accidentally initiate an attack against the institution. Regardless of motivation, this represents an attack surface for multiple security incidents. This work approaches the possibility of intentional or accidental attack by several requests which can subvert the system and generate a blockchain DoS attack. Thus, questions arise related to the environment in which the blockchain is inserted, as well as to the resistance and stability of the blockchain transactions while a DoS occurs. This work conducts a deep resistance analysis of a virtual machine based Ethereum private blockchain network against internal DoS attacks.

The work is organized as follows. Sections 2 and 3 explain the problem definition and related work. Section 4 presents our analysis proposal, as well as the experiments and scenarios. Section 5 presents the results and analysis.

## 2 Problem Definition

The growth of blockchain usage is remarkable [10,14], as well its applications demands and requirements [8]. Developers usually seek to improve their applications regarding efficiency and quality of the technology, but there are concerns related to the computational costs associated to performance and security. Thus, the nodes of a blockchain are deployed according to these requirements, e.g., number of transactions per minute they must process, efficiency, etc.

A blockchain consensus mechanism has a direct relationship to the amount of resources needed in the node. The choice of blockchain solution is based on the consensus mechanisms it supports. In this context, there are several consensus mechanisms (e.g., RAFT, iBFT, Practical Byzantine Fault Tolerance (pBFT), PoA, etc.) responsible for the validation processes of the blockchain network.

An important issue is the environment in which the nodes are created, highlighting computational clouds with their virtualization services. In general, the recommended flavor configurations for creating VMs, in private and consortium blockchain models, may change according to the platform/application. A platform which stands out in distributed applications and smart contracts is Ethereum. Ethereum has two types of recommendations for blockchain nodes run: *(i)* specific (4 vCPUS, 8 GB RAM and 1 Gbps network), and *(ii)* generalized (2 vCPUS, 4 GB RAM and 1 Gbps network). Each recommendation intends to meet the application needs (e.g., transactions processed per minute, latency). However, if the number of transactions submitted is above normal or if the resources are exhausted, this leads to a DoS scenario. To exemplify these situations, we define two scenarios illustrated in Figs. 1 and 2.

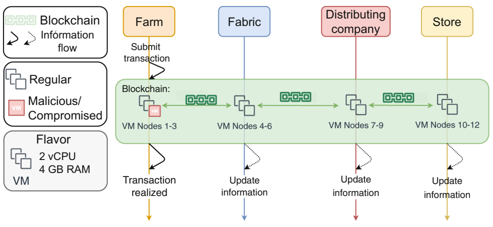

**Fig. 1.** Supply chain application scenario.

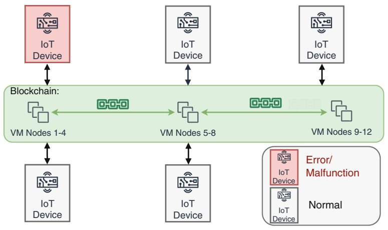

**Fig. 2.** Internet of Things (IoT) scenario.

Initially, we present a real and simple example of a supply chain, illustrated in Fig. 1. It is possible to observe an interactive environment and message exchanges between of the VMs hosting the blockchain network. Specifically, twelve VMs maintainers of the blockchain network are observed, responsible for the transaction validation process and insertion of new blocks. In this scenario a malicious user can DoS against the VM of one or more blockchain nodes. As the flavor of these nodes is not very robust in terms of processing power, it is very likely that one or two attacking VMs may generate enough requests for some other service running on the blockchain node (e.g., SSH is normally enabled for maintenance) and exhaust its memory or compute power. This attack, while not directly against the blockchain, exhausts the VM resources needed to carry out blockchain transactions and can lead to latency or denial of legitimate requests.

Figure 2 illustrates an environment composed of twelve VMs and five IoT devices that send transactions to the blockchain network. This scenario reflects a production environment, in which at a given moment one of the IoT devices fails and sends several transactions per second to the blockchain network (e.g., instead of sending a transaction per minute, it starts sending a transaction every 2 s). The blockchain network does not have the ability to distinguish only the true transactions that are sent by the device, producing an DoS with a reduction in bandwidth. This attack occurred unintentionally due to an unexpected failure that destabilized the application and the environment. Although this scenario illustrates an unintentional DoS, this could also be done intentionally by a malicious IoT device.

The analysis of these two scenarios leads to questions related to:

(i) the proper settings of the instance flavor,
(ii) the number of transactions needed to make the technology operational; and
(iii) the metrics and criteria for detecting issues on the blockchain.

These issues are important for the development of a blockchain network, as the number of transactions required and supported is directly related to the configuration of VMs.

## 3 Related Work

Based on the motivation and problem definition we define three functional requirements (FR) to be addressed by this work:

- (FR1) The environment must allow transactions to be carried out through API or automated mechanisms for blockchain networks;
- (FR2) We must collect the number of transactions sent and their hashes for verification; and
- (FR3) We must collect usage metrics such as CPU, memory, networking, and transaction latency.

We used a systematic review approach to identify related work. Table 1 presents a comparison between the identified works and our requirements.

**Table 1.** Related work *versus* Functional requirements.

| | [3] | [13] | [12] | [5] | [15] | [2] | [9] | [1] |
|---|---|---|---|---|---|---|---|---|
| FR1 | Yes | Yes, until 5k transactions | Yes, until 10k transactions | Yes, until 10k transactions | Yes | No | Yes | Yes |
| FR2 | No | No | No | No | No | No | No | No |
| FR3 | Partially, related metrics: packet overloading and consumption energy | Partially, related metrics: RAM and latency | Partially, related metrics: latency | Partially, related metrics: Latency, tx rate and number of transactions | Partially, related metrics: Latency, tx rate and runtime | No | Partially, related metrics: Latency and tx rate | Partially, related metrics: latency, time and cost of transactions |

Analyzing Table 1, it is perceptible that none of the identified works addresses all the FRs defined in our research. Although FR1 and FR3 are at least partially addressed in most of works, FR2 is absent. Works [1,3,5,9,12,13] assess the performance of blockchain platforms and their consensus mechanisms, carrying out on several transactions on the platforms and analyzing memory consumption, latency, transfer rate, processing time, and energy consumption. Work [2] performs a comparison between different platforms through their characteristics.

## 4 Proposed Approach

The examples from Sect. 2 illustrated a blockchain environment hosted by an Infrastructure-as-a-Service (IaaS) cloud. The nodes of the blockchain network are created on VMs. In a real environment in which an unintentional DoS occurs, there are two situations:

(i) The attack takes place from the accumulation of several transactions received by the platform, causing a high rate of network traffic that results in higher network latency, therefore delaying the validation and insertion of new blocks.
(ii) Similar to the previous item, but as a result of this higher latency and reduction of available network bandwidth, service unavailability and/or packets discarding may occur.

In addition, noting that in this environment the network is formed by known nodes, an intentional DoS occurs in the following situations:

(i) A malicious user belonging to the network carries out attacks against other healthy nodes to consume their resources, either to subvert the system or to reduce the number of participants in the validation process.
(ii) A malicious user sends a significant number of false/positive transactions to the network to increase traffic and latency, and consequently delays in the validation process and insertion of new blocks.

Based on these situations we propose a performance and security analysis in a private blockchain network during the execution of a DoS attack comprising the following steps:

- Analyze the number of transactions per minute that an instance can perform without service loss;
- Analyze the immutability and integrity of transactions; and
- Establish a relationship between the flavor of the instance and the intensity of DoS attacks.

Figure 3 illustrates the testbed setup. Each scenario consists of six instances running operating system (OS) GNU/Linux Ubuntu 20.04. The VMs were configured with a flavor of 2 vCPUs and 4 GB of RAM. This is the flavor recommended for generalized projects on the Ethereum platform [4]. We define three test scenarios with different consensus mechanisms: RAFT, iBFT and PoA. For each

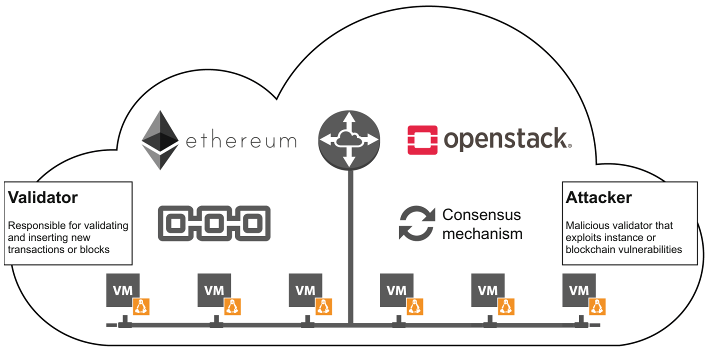
ethereum
openstack.
Validator
Responsible for validating and inserting new transactions or blocks
Consensus mechanism
Attacker
Malicious validator that exploits instance or blockchain vulnerabilities
VM
VM
VM
VM
VM
VM

**Fig. 3.** Testbed used to execute Scenarios I, II, and III.

scenario we verify two situations: (1) occurrence of an unintentional DoS attack; and (2) an intentional DoS attack. In all scenarios the swap is disabled on all VMs. Only main memory is used since the objective is to exhaust the resources. In this sense, the following experiments are performed in each of the scenarios: (I) A malicious user sends transactions directly to a node of a blockchain, here the purpose is consuming the instance's resources; (II) Carrying out several transactions of a user/customer on the blockchain network, in order to reduce network traffic or unavailability of service, aiming to identify, from the monitoring of resources and transactions, the stability of the network in the event of a DoS non-malicious attack. DoS attacks selected:

- Transaction Flood: Flood attack characterized by causing a large volume of traffic on the blockchain network, so that the bandwidth becomes congested; and
- SSH Flood: A protocol exploit attack, characterized by excessive consumption of target resources, attack generates SSH packets (TCP/22) trying to establish connection to the VM.

During the attack we monitored processing and memory resources of the VMs, network traffic, number of transactions and their latency for the blockchain platform.

## 5 Results and Discussion

Scenario I is based on the RAFT consensus algorithm, which is an Crash Fault Tolerance (CFT) algorithm. This algorithm has a certain degree of resilience in their protocol, allowing the system to correctly reach consensus even with failures in its components. However it is not resistant to Byzantine failures. Scenarios II (iBFT) and III (PoA) have algorithms based on Byzantine Fault Tolerance (BFT), which are resistant to component failures and also to Byzantine failures.

In all scenarios, two experiments were executed. The first experiment consists of an DoS attack that seeks to exploit vulnerabilities or flood ports. The second performs a blockchain transaction flood attack. During the entire process of executing the experiments, the resources of the instances and the blockchain network were monitored (based on System Activity Report - SAR - tools) to identify possible instabilities or oscillations in the network.

For the first experiment, attacks were carried out for 240 min. For all scenarios, no changes in processes, memory or any instability that could compromise the blockchain network were detected. For the second experiment, for each of the scenarios different results were identified. Figure 4 and 5 illustrate the results of the second experiment in Scenario I, using RAFT consensus algorithm.

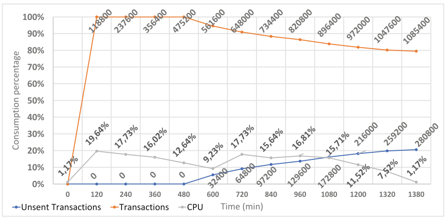

**Fig. 4.** Processor consumption - RAFT.

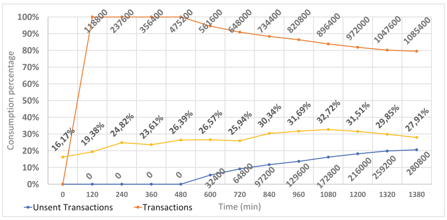

**Fig. 5.** Memory consumption - RAFT.

Figures 4 and 5 present three curves each. Both Figures show the number of transactions completed (and not completed) during the experiment. Figure 4 also presents the CPU consumption during the experiment. In contrast, Fig. 5 presents the memory consumption during the experiment. Analyzing CPU consumption we observe that the RAFT algorithm is relatively efficient and does not require a high computational power. During the entire execution of the attack it remained stable and with few changes in its consumption. Regarding memory consumption the main peaks of the occurrence of a reduction in the amount of transactions sent are clear, as the memory consumption in these peaks is reduced. Transactions begin to drop after 480 min of execution of the attacks.

Figures 6 and 7 illustrate the results obtained from the execution of Scenario II using the consensus algorithm iBFT.

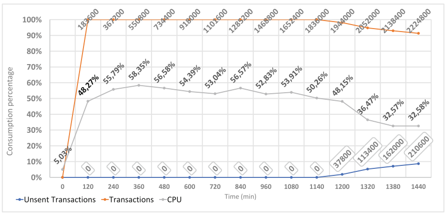

**Fig. 6.** Processor consumption - iBFT.

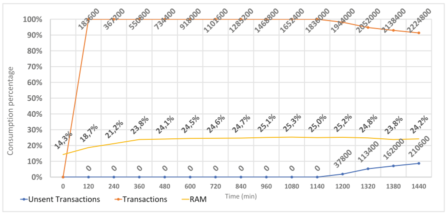

**Fig. 7.** Memory consumption - iBFT.

Regarding the use of processing (Figs. 6 and 7), we observe the difference between a CFT consensus algorithm compared to BFT – due to the complexity and security issues, the use of computational power is greater. However, after 1,200 min of execution of the attack the system remained stable in the use of processing, and as the reduction in processor consumption, the greater the number of transactions not performed by the platform. In a BFT consensus algorithm, the processing reduction means the transaction processing speed reduction, allowing the gradual increase of the process queue. Regarding transactions, the stability of this consensus algorithm against the Transaction Flood attack is notorious. Only after 1,200 min the transactions start to drop.

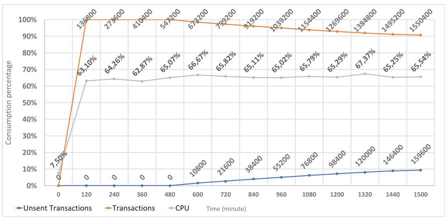

**Fig. 8.** Processor consumption - PoA.

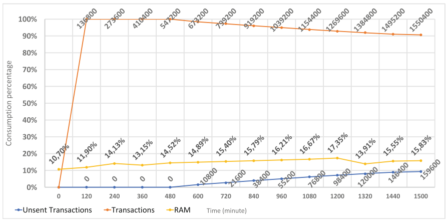

**Fig. 9.** Memory consumption - PoA.

Figures 8 and 9 illustrate the experiments in Scenario III. Regarding processor consumption, since it is a BFT consensus algorithm we observe high processing consumption due to mathematical calculations that require greater computational power. Consumption remained stable during the experiment. Regarding transactions, we observe the stability of the consensus algorithm during the DoS attack. The first occurrence happened during the 600-min time of the attack execution but the growth of unsent transactions occurred in a contained way, compared to Scenarios I and II. In order to better understanding, we develop a compilation of these results (Figs. 10 and 11).

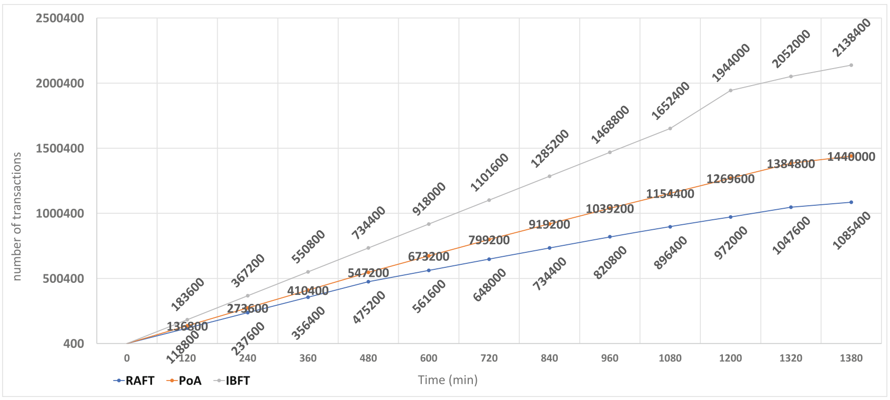

**Fig. 10.** Total transactions sent.

Figure 10 compares the number of transactions sent during the Transaction Flood attack for Scenario I (RAFT), Scenario II (iBFT), and Scenario III (PoA). The iBFT algorithm, which is a BFT algorithm, shows the best performance in terms of number of transactions sent. The worst case is RAFT, which is a CFT algorithm.

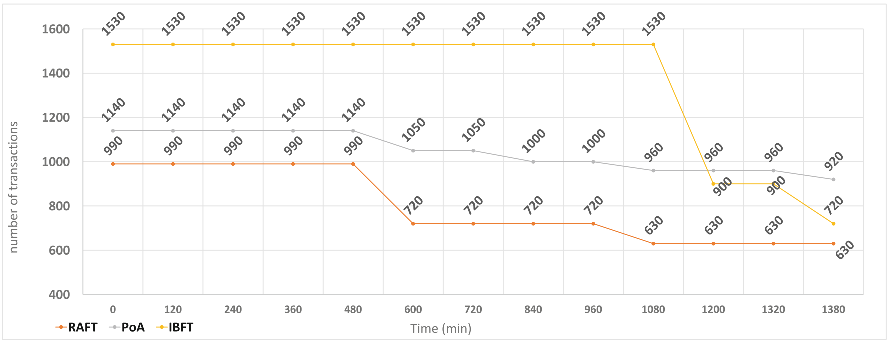

**Fig. 11.** Transactions per minute.

Figure 11 shows the comparison between all these scenarios in the comparison of sending transactions per minute. Regarding the RAFT algorithm scenario, since the transaction flood execution, it had the worst performance in sending transactions per minute, starting with 990 transactions per minute and ending with a sending rate of 630 transactions per minute. The iBFT consensus algorithm, from the beginning, revealed, in comparison with the others, a higher transaction sending rate, this being 1530 transactions per minute. However, with the processing loss and its high transaction buffer, these numbers started to drop, indicating a higher percentage drop in transaction per minute rates, reaching 720 transactions per minute by the end of the experiment. On the other hand, the PoA consensus algorithm showed the best behavior in relation to the amount of transactions sent per minute, which started with 1140 transactions, and since 600 min of attack execution, delays or not sending transactions started, but it kept up a good behavior, reaching the end with a rate of 920 transactions per minute, superior to RAFT and iBFT.

Table 2 presents a summary of all information collected, being separated between the consensus algorithms, the flavor of the VMs, and the number of transactions that can be supported by the instance without catastrophic impairment of the application's functioning.

**Table 2.** General aspects of flavor against the tested consensus algorithms.

| Consensus | RAFT | iBFT | PoA |
|---|---|---|---|
| Processor | 900 Transactions/min<br>20% of 2 vCPU | 1500 Transactions/min<br>55% of 2 vCPU | 1100 Transactions/min<br>67% of 2 vCPU |
| RAM Memory | 900 Transactions/min<br>33% of 4 GB RAM | 1500 Transactions/min<br>58% of 4 GB RAM | 1100 Transactions/min<br>60% of 4 GB RAM |
| Transactions | 900 non-continuous<br>700 continuous | 1500 non-continuous<br>900 continuous | 1000 continuous |

Analyzing the results of Table 2, it is important to highlight some issues: (i) when BFT consensus mechanisms are applied, it is necessary to understand the needs of blockchain applications and according to the complexity and number of transactions, in parallel with the higher processing consumption, this analysis of the application becomes indicated to choose the best *flavor*; and (ii) CFT algorithms usually need more memory than BFT applications, with this it is indicated that this analysis is done so that the choice of flavor is the best possible. Furthermore, concerns about the security and proper functioning of environments and their applications have become increasingly worrying, especially with the growing number of vulnerabilities and attackers. In order to expose more comprehensively, Table 3 provides a comparison between consensus mechanisms, resources, and transactions at the time that DoS attacks started to affect blockchains and at the end of the attack.

**Table 3.** Flavor (2 vCPU, 4 GB RAM, 1 Gbps network): Consensus mechanisms regarding transactions/resources and DoS.

| Metrics | RAFT | iBFT | PoA |
|---|---|---|---|
| DoS start time | 600 | 1200 | 600 |
| End of DoS | 1380 | 1380 | 1380 |
| Start DoS Transactions rate | 720/min | 900/min | 1050/min |
| Transaction rate at the end of DoS | 630/min | 720/min | 920/min |
| Percentage of processor use at the start of DoS | 9,23% | 48,15% | 66,67% |
| Percentage of processor use at the end of DoS | 7,52% | 32,58% | 65,54% |
| Percentage of memory usage at the start of DoS | 26,57% | 25,2% | 14,89% |
| Percentage of memory usage at the end of DoS | 29,86% | 24,2% | 15,83% |

Table 3 establishes a relationship between the flavor of the instance and the intensity of the attacks DoS, in relation to the computational resources used. The results presented have a comparison between the beginning of the occurrence of DoS attacks and the final monitoring, in relation to the scenarios and their respective consensus mechanisms. Among the data, the first highlight is the RAFT algorithm which, like the PoA, had the start of DoS after 600 min of execution, revealing a greater weakness in the face of these attacks in relation to the transactions, but not about the use of your resources. The iBFT consensus algorithm showed the best performance in relation to the amount of transactions sent, however, after the floods started, there was a reduction in its processing, causing the largest differential in transaction rates. PoA showed, in general, the best performance in the ratio of the resources of its flavors vs DoS, keeping in balance, mainly in the use of its resources, with the smallest differential transaction fees. Based on these assessments, it is important to apply good security practices in a private blockchain network, monitoring network traffic, transactions and processing constantly. In addition, that all participating users and validators of the network, have the minimum confidence that is necessary for the proper functioning of the network.

## 6 Considerations and Future Work

The benefits of applications from a private or consortium blockchain in terms of economy, data management, and security are manifold. An important question is related to facilitating the creation of blockchain nodes in computational clouds, through virtualization. In these contexts, there is a concern with issues related to the security of the application and its development environment.

The analyzes performed by our work reinforce the concern about the need for security and continuous monitoring of a private blockchain network. Based on the results presented by our experiments, it is evident that an DoS attack, malicious or not, results in noticeable damage to the blockchain network and its users, which also results in the possibility that other vulnerabilities in blended attacks can be exploited. Regarding the flavor e.g., 2 vCPU, 4 GB RAM, and 1 Gbps network of the VMs we identified it is enough for the general needs of the blockchain platform and its application, but the choice of the consensus mechanism applied, and the number of transactions necessary to operationalize the application, are necessary for safe and optimized choices. This work allows to better understand about private/consortium blockchain applications from VMs in computational clouds, in particular those based on the Ethereum platform.

Future works includes these research to be extended in order to analyze other flavor sizes, and the use of other blockchain platforms/consensus mechanisms, applied within computational clouds.

**Acknowledgements.** This work was supported by Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC), Santa Catarina State University (UDESC), and Laboratory of Parallel and Distributed Processing (LabP2D).

This work was supported by Ripple's University Blockchain Research Initiative (UBRI).

This work received financial support from the Coordination for the Improvement of Higher Education Personnel - CAPES - Brazil (PROAP/AUXPE).

## References

1. Abreu, A.W.d.S., Coutinho, E.F., Bezerra, C.I.M.: Performance evaluation of data transactions in blockchain. IEEE Latin America Trans. **20**(3), 409–416 (2021). https://latamt.ieeer9.org/index.php/transactions/article/view/5429. Number: 3
2. Chowdhury, M.J.M., et al.: A comparative analysis of distributed ledger technology platforms. IEEE Access **7**, 16,7930-16,7943 (2019)
3. Dorri, A., Kanhere, S., Jurdak, R., Gauravaram, P.: Blockchain for IoT security and privacy: the case study of a smart home. In: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 618–623 (2017). https://doi.org/10.1109/PERCOMW.2017.7917634
4. Ethereum Foundation: Nodes and clients - ethereum. Technical report, Ethereum.org (2021). https://ethereum.org/en/developers/docs/nodes-and-clients/
5. Hao, Y., Li, Y., Dong, X., Fang, L., Chen, P.: Performance analysis of consensus algorithm in private blockchain. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 280–285 (2018)
6. Lin, I.C., Liao, T.C.: Survey of blockchain security issues and challenges. Int. J. Netw. Secur. **19**, 653–659 (2017)
7. Linux Foundation: An introduction to hyperledger (2018). https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf
8. Loch, W.J., Koslovski, G.P., Pillon, M.A., Miers, C.C., Pasin, M.: A novel blockchain protocol for selecting microservices providers and auditing contracts. J. Syst. Softw. **180**, 111,030 (2021). https://doi.org/10.1016/j.jss.2021.111030. https://www.sciencedirect.com/science/article/pii/S0164121221001278
9. Monrat, A.A., Schelén, O., Andersson, K.: Performance evaluation of permissioned blockchain platforms. In: 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pp. 1–8 (2020). https://doi.org/10.1109/CSDE50874.2020.9411380
10. Needham, M.: Global spending on blockchain solutions forecast to be nearly $19 billion in 2024, according to new idc spending guide (2021). https://www.idc.com/getdoc.jsp?containerId=prUS47617821
11. Ogiela, M.R., Majcher, M.: Security of distributed ledger solutions based on blockchain technologies. In: 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), pp. 1089–1095 (2018). https://doi.org/10.1109/AINA.2018.00156
12. Pongnumkul, S., Siripanpornchana, C., Thajchayapong, S.: Performance analysis of private blockchain platforms in varying workloads. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1–6 (2017). https://doi.org/10.1109/ICCCN.2017.8038517
13. Rouhani, S., Deters, R.: Performance analysis of ethereum transactions in private blockchain. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 70–74 (2017)
14. Taylor, P.J., Dargahi, T., Dehghantanha, A., Parizi, R.M., Choo, K.K.R.: A systematic literature review of blockchain cyber security. Digital Commun. Netw. **6**(2), 147–156 (2020). https://doi.org/10.1016/j.dcan.2019.01.005. https://www.sciencedirect.com/science/article/pii/S2352864818301536
15. Vatcharatiansakul, N., Tuwanut, P.: A performance evaluation for internet of things based on blockchain technology. In: 2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST), pp. 1–4 (2019)