SANTA CATARINA STATE UNIVERSITY - UDESC
COLLEGE OF TECHNOLOGICAL SCIENCE - CCT
GRADUATE PROGRAM IN APPLIED COMPUTING - PPGCAP

VITOR EMANUEL BATISTA

ANALYSIS OF PERFORMANCE AND SECURITY OF GENERATION OF NFTS APPLIED TO FTS OF ENVIRONMENTAL PRESERVATION AREAS

JOINVILLE

2024

VITOR EMANUEL BATISTA

ANALYSIS OF PERFORMANCE AND SECURITY OF GENERATION OF NFTS APPLIED TO FTS OF ENVIRONMENTAL PRESERVATION AREAS

Master thesis presented to the Graduate Program in Applied Computing of the College of Technological Science from the Santa Catarina State University, as a partial requisite for receiving the Master's degree in Applied Computing.

Supervisor: Dr. Charles Christian Miers

JOINVILLE

2024

Ficha catalográfica elaborada pelo programa de geração automática da Biblioteca Universitária Udesc,
com os dados fornecidos pelo(a) autor(a)

Batista, Vitor Emanuel
Analysis of performance and security of generation of NFTs applied to FTs of environmental preservation areas / Vitor Emanuel Batista. -- 2024.
93 p.

Orientador: Charles Christian Miers
Dissertação (mestrado) -- Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação , Joinville, 2024.

1. NFT. 2. FT. 3. Blockchain. 4. Performance. I. Miers, Charles Christian. II. Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação . III. Título.

Vitor Emanuel Batista

Analysis of performance and security of generation of NFTs applied to FTs of environmental preservation areas

Master thesis presented to the Graduate Program in Applied Computing of the College of Technological Science from the Santa Catarina State University, as a partial requisite for receiving the **Master’s degree in Applied Computing**.

Joinville, 30th July 2024

To my loving wife, family, friends, advisor and mentor, this work is dedicated to your unwavering support.

ACKNOWLEDGMENTS

I dedicate this thesis to all those who have supported and inspired me throughout this challenging journey.

To my wife Anagiulia, my son Vicente and all my family, whose unwavering love and encouragement have been my guiding light. Your belief in my abilities has fueled my determination to succeed, and I am forever grateful for your constant support.

To my friends, who have stood by my side and provided the much-needed laughter and distraction during the most stressful times. Your presence has reminded me of the importance of balance and self-care.

To my professors and mentors, especially my advisor Charles C. Miers, who have imparted their knowledge and expertise, shaping me into the researcher I am today. Your guidance and patience have been instrumental in my academic growth.

I am also grateful to the Carbon 21 Project team and ULBRI, especially Prof. Marcos Antonio Simplício Jr., doctoral student Marco Antonio Marques, master's student Pedro Barcha H. Correia and all team members from both USP and UDESC.

To the participants of my study, who generously shared their time and insights, contributing to advancing knowledge in this field. Your willingness to participate in my research has made this thesis possible.

Lastly, I thank myself for the countless hours of hard work, perseverance, and sacrifices made along the way. This thesis represents a culmination of my dedication and passion for learning, and I am proud of my journey.

May this thesis serve as a testament to the collective effort and support that has brought me to this point. Thank you all for being a part of my academic pursuit and for believing in me.

ABSTRACT

Cloud computing has become the dominant choice for hosting various systems and services, and public cloud service spending has grown substantially. This trend has extended to blockchain technology, which offers decentralized solutions for diverse applications. Concurrently, there is an increasing focus on business models incorporating Environmental, Social and Governance (ESG) aspects. One such initiative is Carbon 21, a platform generating tokens in response to reforestation actions and the carbon credit market. This article examines the performance aspects of generating Non-Fungible Tokens (NFTs) in a blockchain environment under Denial of Service (DoS) attack conditions. The Hyperledger Caliper was the benchmark tool used in experiments conducted to analyze the blockchain's resilience and stability in a Virtual Machine (VM). The experiments were executed in a local and cloud environment, using different Transactions Per Second (TPS) configurations to identify how the system behaves. Furthermore, our linear regression models showed a strong positive correlation between memory usage and transaction amount. These results highlight the need for precise cloud resource sizing and robust monitoring mechanisms to prevent service degradation under high transaction loads.

Keywords: NFT, FT, blockchain, performance.

RESUMO

A computação em nuvem tornou-se a escolha dominante para hospedar vários sistemas e serviços, com um crescimento substancial dos gastos com serviços em nuvens do tipo pública. Essa tendência se estendeu à tecnologia blockchain, que oferece soluções descentralizadas para diversas aplicações. Ao mesmo tempo, há um foco crescente em modelos de negócios que incorporam aspectos Environmental, Social and Governance (ESG). Uma dessas iniciativas é o Carbon 21, plataforma que gera tokens em resposta a ações de reflorestamento e ao mercado de créditos de carbono. Este artigo examina os aspectos de desempenho da geração de NFTs em um ambiente blockchain sob condições de ataque DoS. O Hyperledger Caliper foi a ferramenta de benchmark utilizada em experimentos realizados para analisar a resiliência e estabilidade do blockchain em uma VM. Os experimentos foram executados em ambiente local e em nuvem, com diferentes configurações de TPS para entender como o sistema se comportaria. Os modelos de regressão linear mostraram uma forte correlação positiva entre o uso de memória e o total de transações. Esses resultados destacam a necessidade de dimensionamento preciso de recursos nas nuvens, bem como mecanismos robustos de monitoramento para evitar a degradação do serviço sob altas cargas de transações.

Keywords: NFT, FT, *blockchain*, desempenho.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
C	Smart Contract
CA	Certification Authority
Cetesb	Environmental Company of the State of São Paulo
CID	Content Identifier
DB	Database
DoS	Denial of Service
EDA	Exploratory Data Analysis
EIP	Ethereum Improvement Proposal
ESG	Environmental, Social and Governance
FR	Functional Requirements
FT	Fungible Token
IaaS	Infrastructure as a Service
IBAMA	Brazilian Institute of Environment and Renewable Natural Resources
iBFT	Istanbul Byzantine Fault Tolerance
ICO	Coin Offering
IO	Input/Output
IPFS	Interplanetary File System
IQR	Interquartile Range
L	Landowner
NFR	Non-Functional Requirements
NFT	Non-Fungible Token
P2P	Peer-to-Peer
PoA	Proof-of-Authority
PoS	Proof of Stake
PoW	Proof of Work
SSH	Secure Shell
SUT	System Under Test
TPS	Transactions Per Second
UDESC	Universidade do Estado de Santa Catarina
VM	Virtual Machine

CONTENTS

1 INTRODUCTION

Cloud computing has become the prevailing choice for hosting systems and services in various domains, with worldwide end-user spending on public cloud services forecast to grow 20.4% to total $675.4 billion in 2024, up from $561 billion in 2023 (LoDolce e Howley 2024). Based on this, the Infrastructure as a Service (IaaS) model offers on-demand access to fundamental computing, storage, and networking resources, with virtualization technology empowering users to control these resources. Meanwhile, blockchain technology is growing, comprising Peer-to-Peer (P2P) networks, cryptography, algorithms, and consensus mechanisms, presenting decentralized solutions with diverse applications (Tinu 2018).

On the other hand, the business community has shown an increasing interest in developing business models that incorporate ESG aspects (Filho et al. 2021; Beck R. Agerskov 2024). The recent wave of digital asset tokenization using decentralized technologies such as blockchain presents a significant opportunity to create projects; one considered in this work is called Carbon 21, a platform that generates tokens in response to reforestation actions and the carbon credit market. This platform aims to operate the official carbon market, encompassing broader aspects beyond valuing forest areas. This potential market for tokens generated by reforestation activities can provide currently unobserved economic incentives, making it an attractive alternative to land leasing for agricultural purposes.

Institutions adopting private or consortium blockchain models have created nodes using VMs within their private or hybrid computing clouds. blockchain solution developers often provide pre-configured VM images on their websites, optimized for ease of use (Hyperledger 2023). However, organizations can create their own VMs or containers housing blockchain nodes for their applications. We focus on using VMs, considering their widespread adoption, while leaving experimentation with containers for future investigations.

This work addresses the potential risks posed by intentional or accidental attacks, specifically through a series of requests that can subvert the system and result in a blockchain DoS situation, especially with operations to create NFTs and Fungible Tokens (FTs) in the context of the Carbon 21 project. The focus is on understanding the blockchain's environment, evaluating its resilience, and assessing the stability of blockchain services and transactions in the face of a DoS attack. Therefore, the objective is to analyze the performance and security aspects of generating NFTs and FTs on cloud infrastructure to ensure the effectiveness and reliability of such applications.

We identified some related works, but none with a specific level of detail, metrics, and extensive results within a DoS attack environment for NFT blockchain. Furthermore, understanding the interactions between blockchain, VMs, and cloud infrastructure with the proper VM configuration (flavor) is critical to optimizing performance, scalability, and security in cloud projects.

The main contributions of this master thesis are:

- Detailed examination of the Hyperledger Caliper performance and scalability of blockchain networks (Raft consensus mechanism) in generating NFTs and FTs, specifically under DoS attack conditions.
- Identification of memory and other metrics consumption trends, and their impact on the stability and performance of blockchain services.
- Resource Optimization: Recommendations for optimal VM configurations and resource allocations to enhance the reliability and efficiency of blockchain environments.
- Development and customization[1] of experimental tools using Hyperledger Caliper to gather comprehensive performance metrics captured every second.
- Use data analysis techniques, including Box Plots and regression models, to interpret experimental results and uncover significant patterns.

The method used in this work consists of a referenced research carried out to develop the theoretical foundation, followed by applied research to verify the feasibility and functionality of the proposed solution. The organization of the work is as follows. Chapter 2 introduces the theoretical foundation, approaching how the concept of digital asset, blockchain, NFT and FT are used in the Carbon 21 project. Based on the theoretical foundation, the basis for identifying the proposed research problem and its motivations is established. We also detail the Functional Requirements (FR) and Non-Functional Requirements (NFR) guiding our experiments and scenarios. Chapter 3 presents our adopted research method, as the search, inclusion, and exclusion criteria, academic search mechanisms used, and how the identified related works meet or do not meet the established FR. Then, the proposed solution for the identified research question is detailed, addressing individually how the FR will be attended. Based on the definition of the proposed solution, the chapter also details the test plan and setup to run the analysis of performance and security of generation of NFTs in the Carbon 21 project. Chapter 4 presents the results descriptively and provides a detailed analysis. We discuss our experimental setup and the data analysis techniques, such as box

[1] Available on https://github.com/vitorebatista/caliper

plots and regression models, to interpret the performance metrics collected during the experiments. The findings are then analyzed to identify patterns, trends, and insights into the system's behavior under different conditions. This analysis helps to understand the strengths and weaknesses of the Carbon 21 project's implementation, providing a foundation for future improvements and optimizations.

2 FUNDAMENTAL CONCEPTS

The emergence of NFTs has transformed how people perceive and interact with digital assets. Cloud computing has revolutionized the business landscape, allowing organizations to leverage on-demand computing resources and adopt a modular software architecture composed of independent services (Tianfield 2011). While this paradigm shift presents exciting opportunities, it also challenges optimizing performance, resource utilization, and cost-effectiveness. Thus, there are opportunities to explore the need for an architecture that can efficiently process requests while maximizing computational resources' performance and cost benefits. The availability of on-demand computing resources has opened up new horizons for businesses (Pahl et al. 2017). Instead of investing in expensive hardware infrastructure, organizations can now access computational power and storage as needed, paying only for what they consume (Al-Roomi et al. 2013). This flexibility enables scalability, agility, and cost savings, making cloud computing attractive for businesses of all sizes.

In order to understand the problem presented in this work, it is necessary to go over relevant aspects related to systems using new technologies and concepts. Thus, this chapter presents digital assets, which, unlike traditional physical assets, digital assets exist solely in digital form, allowing for increased accessibility, ease of distribution, and potential for creative expression (Section 2.2). Then, the blockchain ecosystem embraced the cause with a decentralized and immutable digital ledger that records transactions across multiple computers, ensuring transparency, security, and accountability (Section 2.3). With the need for digital assets and the potential of blockchain, NFT emerged to create unique digital assets that are indivisible and cannot be exchanged on a one-to-one basis like cryptocurrencies, providing creators with the ability to establish ownership rights, define royalties, and offer limited edition or exclusive digital assets (Section 2.4).

Based on this, the growth of the developer community, and new projects using the mentioned technologies, there is a research opportunity to ensure optimal performance, efficient resource utilization, and cost-effectiveness in cloud computing environments; an architecture must be carefully designed. It should consider workload distribution, load balancing, scalability, fault tolerance, and resource allocation. By intelligently managing these aspects, organizations can achieve better responsiveness, reduced latency, improved throughput, and ultimately enhance the overall user experience (Section 2.5).

In the context addressed, so far, few researchers and practitioners have delved into the realm of performance analysis using NFTs. There is a relevant research potential to evaluate the performance of a NFT application considering standard environment settings stipulated by the community, allowing the obtained results to be considered by active developers and future stakeholders who wish to join through a project and want to know the proper environment configurations according to minimum requirements of the application (Section 3.2).

2.1 SCENARIO CONTEXT

The business community has been increasingly concerned with building business models considering aspects such as the Environmental, Social and Governance (ESG) (Filho et al. 2021). Accompanying this movement, several platforms have emerged aimed precisely at promoting this concept.

One example is systems that facilitate the calculation of individuals' and companies' carbon footprints while allowing actions to be taken to offset this footprint. Although promising, these platforms are commonly aimed at the voluntary compensation market (Moss 2023) or only at entities with a vast planted area (Carbon 2023). In addition, the calculation involved in accounting for carbon credits generated by a project involves a large number of variables, requiring specialized consulting, which is often seen as a factor that hinders its adoption by landowners who do not have an excellent knowledge of this market (P.H. et al. 2015). Finally, as several existing projects have an international origin, they are not necessarily well adapted to Brazilian legislation (e.g., current national laws prohibit local deforestation from being compensated abroad, which reduces the interest of competitors that deal with international areas) (P.H. et al. 2015).

On the other hand, following the current wave of tokenizing assets using decentralized technologies such as blockchain, there is excellent potential to create a platform that generates tokens in response to reforestation actions. This can be done and independently (and even complementary) to the official carbon market, given that the latter seeks to be much broader than just valuing forest areas. As a result, assuming it is possible to create a market of people interested in the generated tokens, the reforestation activity can gain an economic interest that has not been observed yet, becoming an attractive alternative to land leasing for agricultural activities.

In this architecture, at least two types of tokens are foreseen:

1. Non-Fungible Token (NFT): represents a specific reforestation area. More than one actor can share ownership proportionally, stimulating cooperation between investors and landowners in the reforestation process. Shared NFTs pay FTs proportionally to the share of each of the owners.

a) Utility 1: can be sold as a "reforestation token". In this case, a reclaimed area (or with a commitment to be reclaimed) can be used for environmental compensation purposes (e.g., by contractors). Under current legislation, such an area would need to be recovered a *posteriori*, i.e., after registering a commitment to reforest; this involves registering the land with a degraded area recovery plan). However, with the proposed system, it is technically feasible to allow the landowner to do this registration themselves and then sell the reforestation token to a third party who needs to make the compensation, provided that such token has not yet been used for environmental compensation (i.e., such use would be single-use on the blockchain). It is vital to note that such functionality does not allow something like "compensating without actually reforesting", as only areas that could be used for compensation are registered in the system (e.g., areas already considered forest areas could not benefit from this functionality). Therefore, what is sought in this scenario is essential to encourage reforestation even before the need for compensation, i.e., the planting of trees is "in advance", generating credit for future use.

b) Utility 2: generation of "preservation dividends" in the form of fungible tokens, representing ownership of the area for a specified period (e.g., each hectare would annually generate a fungible token). This generates an extra benefit for those who carry out forest compensation through the platform: it is not just an environmental liability but also an asset that generates fungible tokens. Again, this mechanism encourages landowners to get ahead of the reforestation process.

c) Utility 3: a new use for environmental compensation as an understory area. Such use is very similar to reforestation tokens, as it allows an area to be used for forest compensation. However, according to the current legislation, an interval of about 32 years is necessary for an originally reforested area to be used for new compensation in this modality.

2. Fungible Token (FT): Refers to tokens generated by an area registered as a non-fungible token. While these tokens serve a similar role to those generated in the carbon credit market, the aim is for them to be used as "preservation credits" or "water generation credits", operating independently of the carbon market. This approach aims to avoid the carbon credit market's inherent complexity while allowing landowners to enter it for additional profits. Despite being fungible, relevant metadata such as the fact of being in a specific biome (e.g., Cerrado, Atlantic Forest, Amazon, etc.) can be included in the records; similarly to what happens with engravings placed on physical coins of Real, this metadata in principle does not

change the financial value of the coin, but it can bring additional value to the tokens from the point of view of collectors.

As the system deals with the tokenization of real-world assets, mechanisms must be established to validate the registered data, preventing fraud in which changes in the real asset are not reflected in the blockchain. Some relevant situations are listed below by way of example (the production of an exhaustive list is part of the scope of the project):

1. The existence of a specific area and its owner: notarial documents can verify these characteristics when registering an area in the system as a NFT. It is likely that such documents, at least initially, do not have standardized formats or digital signatures, making their automated analysis (e.g., by smart contracts) difficult. Thus, a manual check may be required. A point of expansion of the project would be to promote such digitalization of land registrations, for example, via a partner company in the area of tokenization of properties (Tech 2021). In any case, documents proving existence and possession (whether they are automatically verifiable or not) must be part of the respective records.
2. The area's state regarding forest cover: this item requires constant monitoring. So, validation can involve manual analyses, such as city hall audits and satellite audits, with automated deforestation verification algorithms. These last analyses may involve registering the area's coordinates with systems that provide frequent updates (e.g., Google Earth) and may rely on the assistance of specialized partners - suggested names include ESALQTec, Embrapa Satelite, INPE, and MapBiomas. In terms of process, the initial registration may rely on a digital signature from a trusted entity, while subsequent registrations would involve periodic re-endorsements with low granularity (e.g., one new verification per year) or non-endorsement alerts at any time (e.g., deforestation noticed by some internal mechanism or an accredited partner generates an alert registered in the blockchain, requiring re-analysis). To allow re-verification by any interested entity, the record's metadata must include the area's geographic coordinates and instructions for obtaining a satellite image.

The operating costs of this system refer essentially to the maintenance of the miners' infrastructure, which translates into: investment in computational resources, such as processing, memory and communication bandwidth; basic operating costs, such as electricity and maintenance of physical space; and personnel expenses. To cover these expenses, an initial reserve is required for cash flow purposes.

An alternative for raising these funds consists of launching an Coin Offering (ICO) of the platform’s FT token, a helpful strategy to stimulate investor interest and generate liquidity. Another possibility involves the payment of referral bonuses, in which the person who acquires the token during the ICO via referral receives and generates a bonus in FT for himself and the person who referred him.

During system operation, on the other hand, fees are provided for the following system operations:

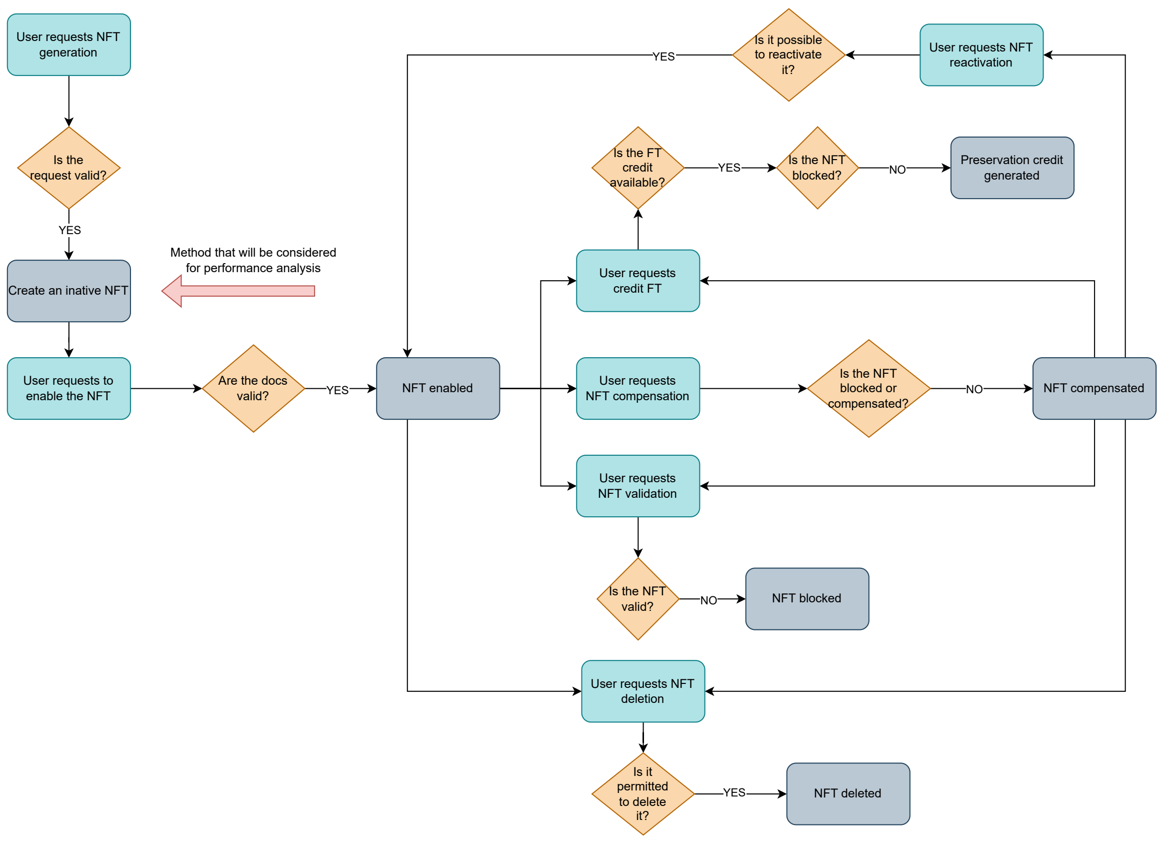
1. Generation of Fungible Token by Non-Fungible Token: when registering an area in the system in the form of a NFT, there must be a global smart contract that regulates the percentage of participation of the system in the generation of FTs by that NFT. For example, if the rate is 1%: when A registers an NFT that, according to the system’s rules, generates 100 FTs annually, the owner of the FT receives 99 FTs, and one FT is credited to the system. Therefore, the system obtains FTs even without having a preserved area that allows its automatic generation, being able to sell them for their market value.

2. Non-Fungible Token transactions: the market for buying and selling NFTs involves financial fees reverted to the system in the form of FTs that need to be included in the transaction as remuneration. For example, to record a NFT sale transaction from A to B, the transaction must also include the FTs that will be passed on to the system for the transaction to be valid (whether these FTs are from A, B, or both).

3. Fungible Token transactions: the market for buying and selling FTs involves financial fees reverted to the system. For example, if the fee is 1%: when A buys 100 FTs from B, recording this transaction on the blockchain leads B to acquire 99 FTs, and 1 FT is registered as “system fee”, going into his possession.

Considering these requirements, the main actions involved in a potential “user journey” in the Carbon 21 system can be seen in Figure 1, which explores the project life cycle showing the actions sequence necessary to request the creation of a NFT and other sub-operations.

Figure 1 – Life cycle of Carbon 21.


User requests NFT generation
Is the request valid?
YES
Create an inative NFT
Method that will be considered for performance analysis
User requests to enable the NFT
Are the docs valid?
YES
NFT enabled
Is it possible to reactivate it?
YES
User requests NFT reactivation
Is the FT credit available?
YES
Is the NFT blocked?
NO
Preservation credit generated
User requests credit FT
User requests NFT compensation
Is the NFT blocked or compensated?
NO
NFT compensated
User requests NFT validation
Is the NFT valid?
NO
NFT blocked
User requests NFT deletion
Is it permitted to delete it?
YES
NFT deleted

Adapted from (Carbono21 2023)

Each of the operations has its due importance within the flow of the Carbon 21 project, which are described below:

1. Registration: Landowner (L) registers an area equivalent to X hectares in the system, presenting supporting documentation that the land (e.g., notary documents) is in his possession and that it can be reforested (e.g., that it is not an area already occupied by forest cover). The request is registered on a blockchain pending approval.

2. Validation: After evaluating the request, which can be done manually or automatically depending on the documents required and presented, the request generates X NFTs for L, associated with the applicable Smart Contract (C).

3. Dividends: After a period defined in C, the X NFTs generate Y FTs for L, registered in the blockchain according to the distribution defined in the C. Although the distribution must involve L mostly, it can also include the system (by way of fees) and some promoter (by way of remuneration for the system's promotion action). Distributions may differ from time to time (e.g., there may be contracts where the promoter only receives FTs the first time they are generated by a given NFT).

4. Compensation via NFTs: At any time, an investor interested in making forest compensation can acquire the compensation rights associated with the NFT, paying with several FTs negotiated between the investor and the NFT's owner. This transaction is registered on the blockchain, blocking the use of that NFT for new environmental compensation for a period defined in a smart contract, in accordance with current legislation.

5. Purchase/sale of NFTs: At any time, an investor may purchase NFTs from any entity in the system that owns such tokens. For example, the investor can be a speculative, who buys FTs believing in their future appreciation; another possibility is that the investor is an entity that needs to acquire FTs in order to be able to pay for forest compensation rights associated with some NFT. Furthermore, it is possible that the investor wants to acquire the NFTs related to a newly acquired rural property, or even from third parties, in search of the "dividends" generated in the form of FTs. In either case, the purchase price must be negotiated between the parties, for example, using a mechanism for listing bids and offers similar to that used on stock exchanges. Once the purchase is completed, authorized by the digital signature, and with due payment of fees to the system, the NFTs in question passes into the investor's possession, a transaction registered on the blockchain. It should be noted that the purchase and sale of FTs will likely be more common in the system, given that ownership of NFTs will normally (although not necessarily) remain with the owner of the land to which that NFT refers.

The goal of the Carbon 21 initiative is to make reforestation a profitable and enticing business for owners of small to large scale properties. Essentially, it provides a tokenization platform that leverages the opportunities created by existing ecological compensation laws, micro-generation of carbon credits, the timber market, and any other ESG related actions, while also generating an opportunity of its own with the minting of digital coins backed by real-world forested areas. To benefit from the Carbon 21 initiative, landowners need only to register their lands on the platform, showing proof that the competent authorities have approved the intended land usage for growing trees. Then, they can already start profiting. Instead of just "yet another tokenization platform", Carbon 21 aims to be a force of change for promoting planetary health. Indeed, like trees, it is continuously growing and evolving. While doing so, new use cases and additional opportunities are prone to appear, including collaborations with other platforms that, like ours, aim to create a better and more sustainable world.

Several components, configurations, and workflow decisions contribute to the overall performance of your network (Hyperledger 2023). Managing these variables, such as the number of channels, chain code implementations, and transaction policies, can be complex, with multiple participating organizations contributing their own hardware and networking infrastructures to the environment.

2.2 DIGITAL ASSETS

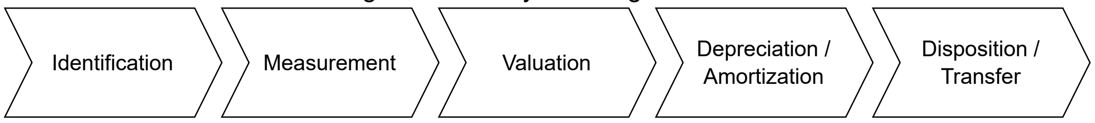
Digital assets have become increasingly essential due to the shift towards a digital economy. The historical need for digital assets can be traced back to the rise of the Internet in the late 20th century, which led to a proliferation of digital data and online transactions (Committee 2022).

One of the earliest and most vital digital assets was the digital currency, first introduced as e-gold in the late 1990s, a digital gold currency operated that allowed users to open an account on their website denominated in grams of gold or other precious metals and that lets users make instant transfers of value to other e-gold accounts. This currency allowed for secure, peer-to-peer transactions without intermediaries like banks or other financial institutions (Linden e Shirazi 2023).

As the Internet grew and became more ubiquitous, the need for secure and efficient ways of storing and transferring digital assets increased. This led to the development of blockchain technology, which provides a decentralized, tamper-proof ledger for recording transactions (Committee 2022). Digital assets are intangible assets that exist only in a digital form. They are often created, stored, and transferred using digital technology, including cryptocurrencies, digital art, music, videos, ebooks, software, etc. (Ankenbrand et al. 2020).

These assets are stored on digital networks, such as the Internet or blockchain, and can be accessed and transferred by anyone with the necessary digital credentials. It can potentially transform how people store, transfer, and value assets. For example, cryptocurrencies like Bitcoin and Ethereum allow for decentralized, secure, and transparent transactions without intermediaries like banks or payment processors. This can lead to faster and cheaper transactions and increased financial freedom for individuals and businesses (Linden e Shirazi 2023). From the perspective of a specific digital asset, its lifecycle starts from identification, goes through measurement, valuation, depreciation/amortization, and ends with disposition and/or ownership transfer as shown in Figure 2.

Figure 2 – Lifecycle of digital asset.



Adapted from Committee 2022

According to Committee 2022, the definition of each of these steps is:

1. Identification: The lifecycle begins with identifying a digital asset. This stage involves recognizing and categorizing assets based on their nature, whether they are physical assets that have been digitized or purely digital assets.
2. Measurement: Once a digital asset is identified, its properties and relevant data are measured and logged. This includes capturing information about the asset's characteristics, ownership details, purchase price, maintenance requirements, and other pertinent information that helps define the asset's value.
3. Valuation: The digital asset's value is determined at this stage. The valuation considers projected revenue generation, anticipated operational expenses, and potential price fluctuations. Assessing these elements calculates the asset's estimated value, which is essential for financial reporting and decision-making.
4. Depreciation/Amortization: If the digital asset is derived from a physical asset, this stage involves accounting for depreciation, which reflects the decrease in value over time. For assets that are not mapped from physical counterparts, amortization may be considered to account for changes in value. Other asset properties may change over time and should be re-measured and updated accordingly.
5. Disposition/Transfer: At some point, the digital asset may end its useful life or become obsolete. This stage involves making decisions regarding the asset's disposition, such as disposing of it, marking it as obsolete, or transferring ownership to external partners. Records of ownership changes, transactions, final property updates, and custody transfers are maintained during this process.

Furthermore, digital assets also have the potential to democratize the creative economy, allowing artists, musicians, and other creators to monetize their work more efficiently and directly. Digital marketplaces for art, music, and other digital content are springing up, creating new opportunities for creators to showcase and sell their work to a global audience (Ankenbrand et al. 2020). Digital assets are essential because they offer new opportunities for innovation, creativity, and economic growth. As digital technology advances, digital assets will likely become an increasingly vital part of the economy and society. It can be an opportunity for a project for several reasons (Zhu et al. 2018):

- Diversification of assets: By incorporating digital assets into a project, the project can diversify its assets and reduce reliance on traditional assets like cash, real estate, or stocks. Digital assets, such as cryptocurrencies or digital tokens, can provide new ways of raising funds, investing, or storing value.

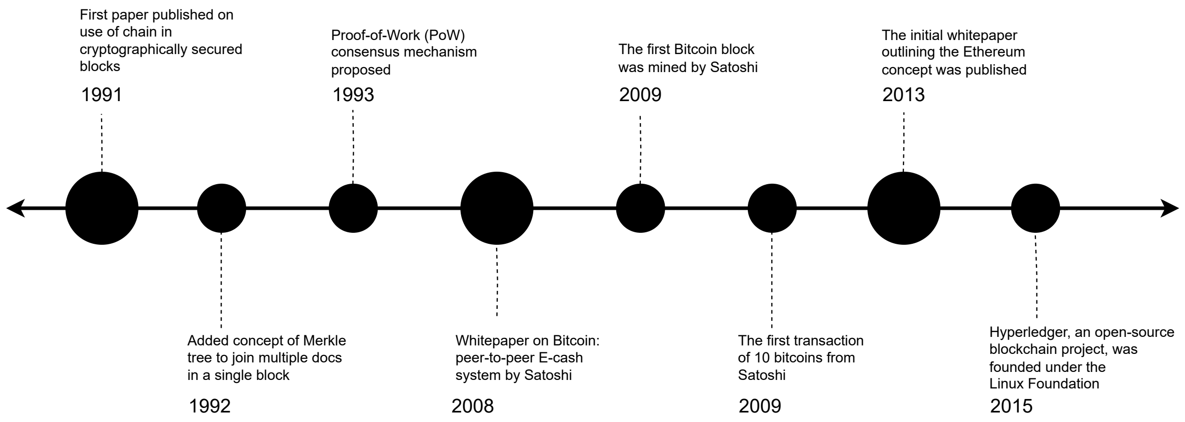
- Increased liquidity: Digital assets can be highly liquid, meaning they can be easily traded and converted into cash or other assets. This can provide greater flexibility and efficiency in managing a project's finances.
- Lower transaction costs: Digital assets can have lower transaction costs than traditional assets, as they do not require intermediaries like banks or brokers. This can make transactions faster, cheaper, and more accessible.
- Innovative business models: Digital assets can enable new business models that were previously not possible. For example, a project could create its own digital token that can be used to incentivize users, reward contributors, or raise funds. This can create new revenue streams and value propositions for the project.
- Increased transparency and security: Digital assets, such as blockchain-based tokens, can provide greater transparency and security in transactions and ownership. This can reduce the risk of fraud, counterfeiting, or unauthorized access and increase stakeholder trust.

Digital assets have become increasingly essential in the digital economy due to their potential for diversification, liquidity, cost efficiency, innovative business models, and improved transparency and security. By embracing digital assets and leveraging their benefits, projects using blockchain technology can position themselves at the forefront of the digital revolution, unlock new growth opportunities, and stay competitive in a rapidly evolving landscape. One of the key advantages of digital assets is their ability to provide diversification. By incorporating digital assets into a project's portfolio, it can reduce its reliance on traditional assets and explore new avenues for growth and investment. This diversification can mitigate risks and enhance overall portfolio performance.

2.3 BLOCKCHAIN

The history of blockchain technology, briefly presented in Figure 3, dates back to the early 1990s, when Stuart Haber and W. Scott Stornetta first introduced its fundamental concepts (Girasa e Scalabrini 2022). They were working on a practical solution to keep the backup of digital documents. Afterward, they aim to make the timestamps of those documents more secure. Later, they published their first paper expressing the use of a chain to cryptographically secure blocks in order to protect the integrity of past information.

Figure 3 – The blockchain initial timeline.


First paper published on use of chain in cryptographically secured blocks
1991
Proof-of-Work (PoW) consensus mechanism proposed
1993
The first Bitcoin block was mined by Satoshi
2009
The initial whitepaper outlining the Ethereum concept was published
2013
Added concept of Merkle tree to join multiple docs in a single block
1992
Whitepaper on Bitcoin: peer-to-peer E-cash system by Satoshi
2008
The first transaction of 10 bitcoins from Satoshi
2009
Hyperledger, an open-source blockchain project, was founded under the Linux Foundation
2015

Source: The author

In 1992, they added Merkle trees, which help collect more documents in a single block. Thus, increasing the model's efficiency, in 1993, Proof of Work (PoW) mechanism was proposed to protect against spam and other network abuses. However, the breakthrough that led to the creation of blockchain as it knows it today occurred with the emergence of Bitcoin in 2008 when a person or group of people using the pseudonym Satoshi Nakamoto published a whitepaper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto 2008). The whitepaper introduced Bitcoin, a decentralized digital currency, and the underlying technology known as blockchain. One year later, the first Bitcoin block was mined, and the first transaction was made. The motivation behind the creation of Bitcoin and blockchain was to address some key challenges and limitations of traditional financial systems. Satoshi Nakamoto aimed to develop a system that would enable secure, peer-to-peer transactions without the need for intermediaries like banks or financial institutions. The goal was to establish a trustless and transparent financial network that would empower individuals to fully control their assets and eliminate the risk of censorship or manipulation (Nakamoto 2008). Nakamoto proposed a blockchain, a distributed ledger recording all transactions transparently and some immutable aspects to achieve this. The blockchain serves as a public database that stores transactional data across a network of computers (nodes) that participate in the system.

The design of the blockchain involves several key elements. Firstly, transactions are grouped into blocks, which are then linked together chronologically, forming a chain of blocks. Each block contains a reference to the previous block, creating an interconnected sequence that ensures the integrity of the data. Secondly, the blockchain employs a consensus mechanism to validate and agree upon the order of transactions. In the case of Bitcoin, this mechanism is called PoW, where participating nodes compete to solve complex mathematical puzzles. The first node to solve the puzzle earns the right to add the next block to the chain and receives a reward in the form of newly created Bitcoins.

The combination of decentralized consensus and cryptographic techniques ensures the security and immutability of the blockchain. Once a block is added to the chain, altering or tampering with the recorded data becomes extremely difficult. This feature makes blockchain an ideal technology for recording and verifying transactions, eliminating the need for trust in a centralized authority (Tinu 2018).

Various industries, including finance, healthcare, and supply chain management, use the technology to improve efficiency, security, and trust in their processes. Therefore, it has gained popularity because it allows for creating tamper-proof and immutable records, making it difficult for fraudsters to manipulate or corrupt data (Yaga et al. 2018).

The different types of blockchains available differ in terms of their permission model and consensus mechanism. The combination of these variables results in blockchain models with distinct applications. Thus, it is necessary to analyze the characteristics of each of them to identify the most suitable blockchain model and consensus mechanism for the proposed solution. It became evident that blockchain could be used as a decentralized and secure platform for various other use cases, such as supply chain management, identity verification, and voting systems, and in the last years, there have been challenges with the use of NFT and FT in projects.

2.4 NFT AND FT

The Non-Fungible Tokens (NFTs) have emerged as a groundbreaking concept within the realm of blockchain technology (Wang et al. 2021). Their unique properties and ability to represent ownership of digital assets have revolutionized various industries, including art, gaming, collectibles, and more (Ethereum 2021). The concept of NFTs can be traced back to the early discussions surrounding the tokenization of digital assets on blockchain platforms.

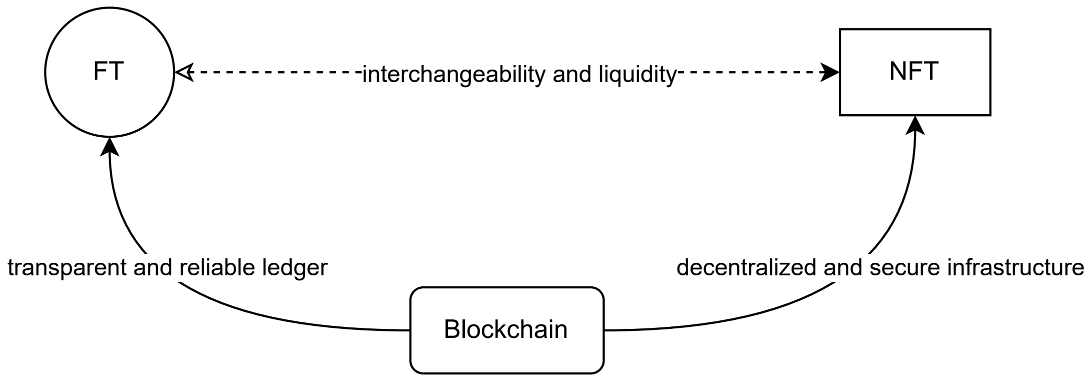
However, it wasn't until 2017 that the potential of NFTs gained significant attention with the introduction of CryptoKitties (Terry e Fortnow 2021). This blockchain-based game allowed users to collect, breed, and trade unique virtual cats using Ethereum's smart contract capabilities. The success of CryptoKitties led to widespread discussions about the possibilities of NFTs and their applications beyond gaming (Liu e Wang 2019).

To ensure interoperability and compatibility across different blockchain platforms, standards for NFTs have been established (Olsson 2022). The most notable standard is ERC-721, introduced in 2018 as an Ethereum Improvement Proposal (EIP). ERC-721 enabled creating and managing unique tokens, paving the way for the explosive growth of NFT-based projects (Wang et al. 2021).

Tokens are classified into FTs and NFTs in terms of fungibility, which means whether one token can be replaced with another token of the same type and quantity. FTs are interchangeable with other FTs because they have the same value as others, and can be divided into smaller units. NFTs are not interchangeable with other NFTs because every NFT is a unique and indivisible unit itself (Hong et al. 2020).

blockchain technology is fundamental in both NFTs and FTs, as shown in Figure 4. NFTs are unique digital assets that rely on blockchain's decentralized and transparent infrastructure. Each NFT is stored as a distinct token on the blockchain, ensuring provable scarcity and ownership rights. The blockchain ledger tracks the ownership and transaction history of NFTs, providing a secure and immutable record that can be easily verified and traced. Thus, this enables creators and collectors to confidently buy, sell confidently, and trade NFTs, knowing that their authenticity and ownership are securely established.

Figure 4 – Relation of NFT and FT with blockchain.


FT
interchangeability and liquidity
NFT
transparent and reliable ledger
decentralized and secure infrastructure
Blockchain

Source: The author

On the other hand, FTs can be freely exchanged on the blockchain. blockchain technology facilitates the secure and transparent transfer of FTs, eliminating the need for intermediaries and enabling peer-to-peer transactions. By leveraging blockchain's decentralized ledger, FTs can be easily traded, stored, and accounted for, providing users with an accurate record of their token holdings. The interchangeability between NFTs and FTs within the blockchain ecosystem allows for seamless interaction. FTs can be used as a medium of exchange for purchasing NFTs, enhancing liquidity and accessibility in NFT marketplaces. Integrating NFTs and FTs within blockchain technology opens up new digital asset ownership, creation, and monetization possibilities.

Consensus mechanisms play a crucial role in validating transactions and maintaining the integrity of NFT-based blockchain networks (Nguyen e Kim 2018). Most NFT projects leverage established consensus mechanisms such as PoW or Proof of Stake (PoS). PoW, used by Bitcoin, requires miners to solve complex mathematical puzzles to add new blocks to the blockchain. PoS, employed by networks like Ethereum 2.0, relies on validators who hold a certain amount of tokens to validate transactions (Ethereum 2021). The choice of consensus mechanism depends on factors such as scalability, energy efficiency, and security (Nguyen e Kim 2018). There are several other consensus mechanisms, but only a few are mentioned here because they are the most publicly known due to Bitcoin and Ethereum (Nguyen e Kim 2018; Tinu 2018).

Performance analysis is paramount in evaluating the effectiveness and sustainability of NFT-based blockchain projects (Wang et al. 2021). It involves assessing various key metrics, including transaction speed, scalability, gas fees, and environmental impact. As NFT popularity surged, blockchain networks faced scalability challenges and high transaction fees, highlighting the need for performance optimization (Nguyen e Kim 2018). Performance analysis enables developers and stakeholders to identify bottlenecks, enhance user experiences, and make informed decisions regarding protocol upgrades or network migrations. It also facilitates the identification of potential security vulnerabilities and optimizing resource allocation within the blockchain infrastructure.

2.5 PROBLEM AND OPPORTUNITIES

Private or public computing clouds are the predominant platforms for hosting systems and services (Hong e Chang 2022). The IaaS model has emerged as a prominent option among the various systems and services offered by computing clouds. Unlike other models, IaaS does not require consumers to manage or control the underlying cloud infrastructure directly. Instead, leveraging virtualization technologies like VMs (Sakiz Burcu 2021) grants them control over operating systems, storage, and applications.

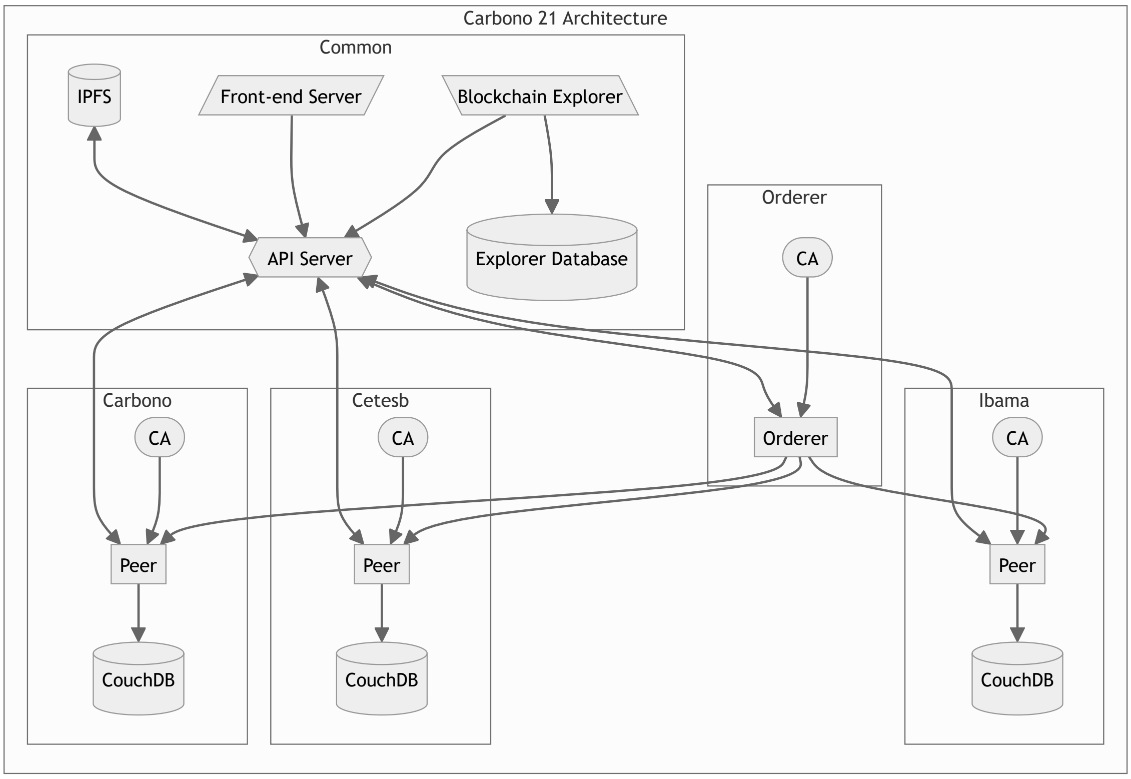
In a cloud computing scenario, in which services are developed and offered to customers on an outsourced platform, monitoring and managing this environment is an essential aspect of the infrastructure. Thus, it makes it possible to improve the scalability and services distribution, detection, failure prevention, and performance analysis, among others (Jiménez et al. 2015). The growth in the use of blockchain is remarkable, as are the demands and requirements of its application. Developers generally seek to improve their applications in terms of efficiency and technology quality, but there are concerns related to computational costs associated with performance and security (Rasolroveicy e Fokaefs 2022). Thus, blockchain nodes are deployed according to these requirements, e.g., the number of transactions per minute they must process, efficiency, etc.

A critical issue is an environment in which nodes are created, highlighting computing clouds with their virtualization services (Hong e Chang 2022). In general, recommended configurations for creating VMs, both in private and consortium blockchain models, may change by platform and application (Hyperledger 2021). Each recommendation is intended to meet application needs (e.g., transactions processed per minute, latency). However, if the number of transactions sent is above average or resources are exhausted, it leads to DoS scenario.

Hyperledger has two types of recommendations for running blockchain nodes: (i) specific (4 vCPUS, 8 GB RAM and 1 Gbps network) and (ii) generalized (2 vCPUS, 4 GB RAM and 1 Gbps network (Hyperledger 2020)). Each recommendation is intended to meet the application's needs (e.g., transactions processed per minute, latency) (Hyperledger 2021). However, suppose the number of submitted transactions is above normal, or the resources are exhausted. In that case, this leads to a DoS scenario (Rasolroveicy e Fokaefs 2022), understanding how the system will behave within this scenario and what is the application limit until resources are exhausted. Thus, rejecting transactions entirely is the objective of this research.

For the Carbon 21 project, the architecture designed for the functioning of the blockchain is based on services and their containers in Docker, as can be seen in Figure 5. Our goal is to evaluate each container's performance by visualizing metrics such as memory consumption, CPU usage, storage I/O, and network activity. In this master's thesis, we will focus on analyzing CPU usage, memory consumption, and transactions (successful, unfinished, failed, and total) over time to statistically understand the correlation between these factors. Each organization (Carbon, Cetesb, Ibama) has its own Certification Authority (CA), peers, and Database (DB). The CAs are responsible for managing the digital identities of the entities in their respective organizations. The peers maintain the ledger and execute smart contracts (chaincode). They interact with the blockchain network to validate and endorse transactions. Each peer is associated with a database that stores the ledger data and the world state. These databases ensure that each peer can independently validate and store the blockchain transactions.

Figure 5 – Carbon 21 architecture.



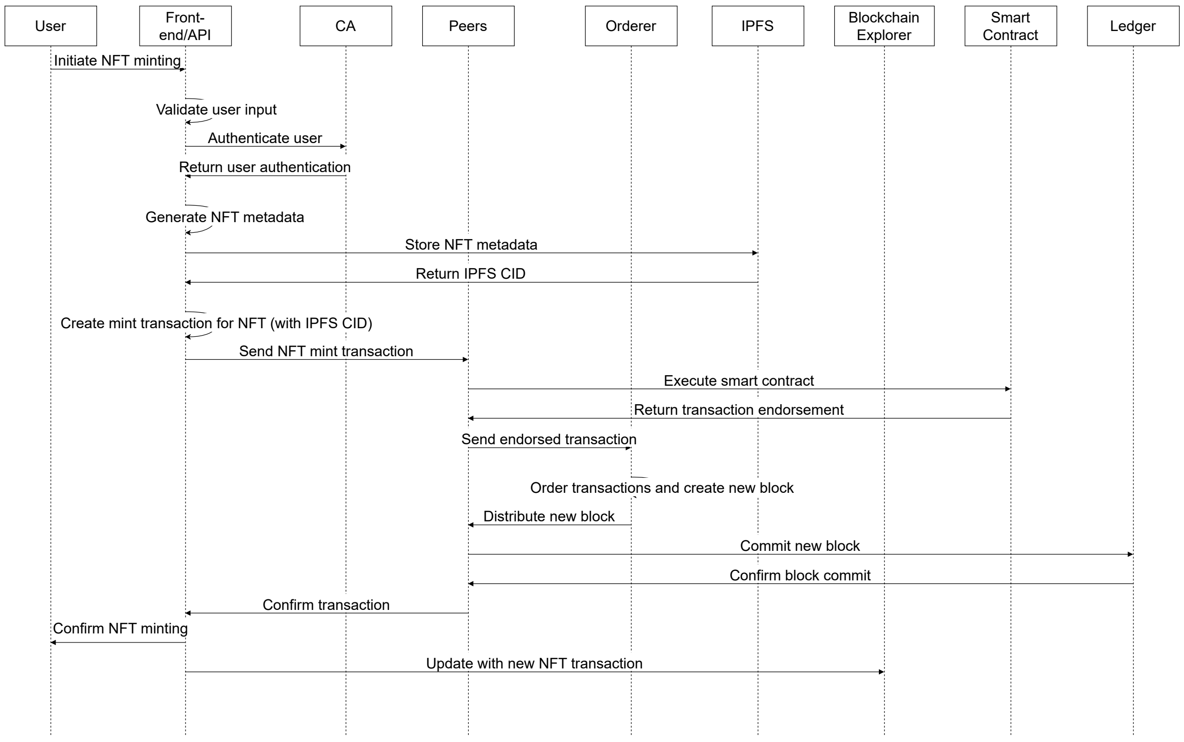
Source: The author.

The orderer is a critical Hyperledger Fabric component, ordering transactions and packaging them into blocks. It ensures the blockchain's consistency and reliability by determining the transaction sequence. The core blockchain network is built on Hyperledger Fabric. This permissioned blockchain platform provides the necessary infrastructure for deploying and managing the distributed ledger, smart contracts, and consensus mechanism.

The API and front-end components interact with the blockchain network to provide a user interface and application programming interface for users and external systems. They facilitate the submission of transactions and retrieval of data from the blockchain. Interplanetary File System (IPFS) is used for decentralized storage of files. It works alongside the blockchain to store large data files in a distributed manner, ensuring data availability and integrity. Finally, the blockchain explorer and explorer DB allows users to view and search the blockchain data. It provides a web-based interface to explore blocks, transactions, and smart contracts deployed on the network.

Each component in the system defined in Figure 7 plays a vital role in ensuring the security, authenticity, and integrity of the NFT minting process described in the sequence diagram in Figure 6. The experiments conducted to collect metrics and analyze results will be based on the execution of the minting process. All steps outlined in the sequence diagram will be treated as a transaction. Thus, by following these steps, the system guarantees that NFTs are created securely, transparently, and decentralized (Vairagade et al. 2022).

Figure 6 – Minting an NFT using Carbon 21 Architecture.



Source: The author.

Figure 6 illustrates the process sequence of minting an NFT using the Carbon 21 architecture, which can be summarized in 13 main steps.

1. Initiate NFT minting: The process begins when a user decides to mint a new NFT, submitting a minting request through the front-end application for this project using the direct Application Programming Interface (API).

2. Validate user input: The front-end application validates the user input to ensure all necessary information for minting the NFT is provided and correct.

3. Authenticate user & Return user authentication: The API communicates with the CA to authenticate the user's identity. This step ensures that only authorized users can mint NFTs.

4. Generate NFT metadata: Once authenticated, the API generates the metadata for the NFT, which includes unique identifiers and properties that define the NFT.

5. Store NFT metadata & Return IPFS Content Identifier (CID): The metadata is then stored on the IPFS, a decentralized storage network. This step ensures that the metadata is securely stored with immutable aspects.

6. Create mint transaction for NFT (with IPFS CID): The API creates a blockchain transaction that includes the IPFS CID and other relevant NFT data. This transaction is necessary to record the minting of the NFT on the blockchain.
7. Send NFT mint transaction: The minting transaction is sent to the peer nodes in the blockchain network for endorsement.
8. Execute smart contract & Return transaction endorsement: The peer nodes execute the smart contract associated with the NFT minting. This step involves validating and endorsing the transaction according to the predefined rules of the smart contract.
9. Send endorsed transaction: Once endorsed by the peers, the transaction is sent to the orderer, which is responsible for ordering transactions in the blockchain network.
10. Order transactions and create new block: The orderer collects and orders the transactions, then creates a new block. This step ensures that the transactions are processed in the correct sequence.
11. Distribute new block, Commit new block & Confirm block commit: The new block, containing the minting transaction, is distributed to all peer nodes. Each peer commits the block to its local ledger, updating the blockchain state.
12. Confirm transaction & Confirm NFT minting: The API confirms the successful minting of the NFT and provides the user with the transaction details, including the transaction ID and metadata.
13. Update with new NFT transaction: Finally, the blockchain explorer is updated with the new NFT transaction. This step ensures that the minted NFT is visible and accessible through the blockchain explorer, providing transparency and traceability.

Therefore, with the number of components and steps to be performed to ensure that the process is completed successfully, it is essential to understand the relationship between consuming resources versus transactions depending on the mode of operation of each blockchain to size the flavor and avoid under or over-sizing resources correctly. With the metrics identified and their results, computational resources (e.g., cloud VM flavor) can be sized according to specific needs and criteria.

2.6 CHAPTER CONSIDERATIONS

The computing cloud model revolutionizes access to computing resources by offering convenience, ubiquity, and scalability. Within this model, various actors play essential roles: consumers who utilize the services, providers responsible for delivering these services, and auditors who ensure effective audit processes. Additionally, this section revisits fundamental technological concepts like virtualization, which plays a pivotal role in enabling the creation of IaaS environments.

The increasing development of applications utilizing VMs underscores the significant impact of virtualization technology. blockchain stands out as one such application that greatly benefits from VMs. Moreover, the growing adoption of relatively new technologies like NFTs and FTs further highlights the importance of virtualization. Notably, institutions have successfully applied VM technology in implementing these innovative technologies. However, mitigating challenges associated with the simultaneous use of blockchain and VMs remains crucial.

Furthermore, this section presents diverse scenarios wherein different users, devices, or applications can intentionally or unintentionally execute DoS attacks on blockchain networks. These scenarios illustrate potential problems that may arise within blockchain networks, emphasizing the need for robust mitigation strategies. Consequently, understanding the relationship between resource consumption and transactions becomes paramount, particularly when aligning the mode of operation with each consensus mechanism. This understanding enables accurate resource sizing, preventing the pitfalls of under or over-sizing resources and ensuring optimal performance.

3 REQUIREMENTS AND PROPOSAL

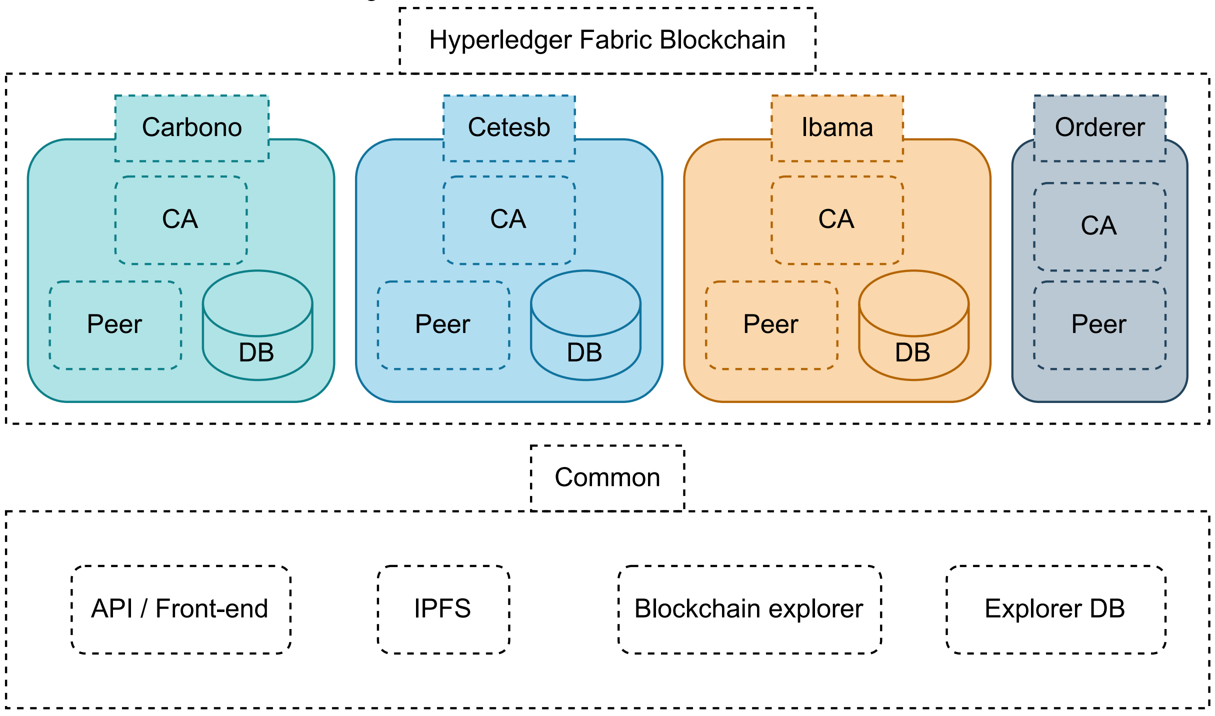
Blockchain technology possesses significant versatility, making it a subject of interest across various sectors. However, it also presents challenges primarily associated with computational costs, application environments, and security concerns (Rasolroveicy e Fokaefs 2022). These challenges vary depending on the intended use of the technology. Consequently, exploring the feasibility of applying complementary technologies alongside blockchain becomes crucial. This exploration should consider factors such as the analysis of the applied model, the consensus algorithm, the specific characteristics of the blockchain implementation, and the environmental factors that can enhance the benefits of this technology.

In order to understand how the Carbon 21 project is built using technologies such as blockchain, NFT, and FT (Section 3.1) presents more aspects and information about the scenario. Then, the requirements to propose a solution to the problem are presented in Section 3.2. Hypotheses and details of what happens when overloading with some types of requests (Section 3.3), as well as the key components (Section 3.4), will also be presented. Based on this, it will be possible to understand the test plan (Section 3.5) and testbed setup (Section 3.6) to run the experiments and collect the results.

3.1 SCENARIO

As described in Section 2.1, several parties could be involved in the development of the Carbon 21 blockchain, each playing a crucial role in the project's success, as seen in Figure 7. The primary organizations that comprise the Carbon 21 blockchain ecosystem include Carbon 21 itself and potential partners such as Cetesb (Environmental Company of the State of São Paulo) and IBAMA (Brazilian Institute of Environment and Renewable Natural Resources). However, it is essential to note that additional organizations may also be involved beyond these initial partners as the project progresses. Additionally, an independent orderer is implemented to handle consensus mechanisms, transaction ordering, and distribution, ensuring a decentralized and impartial approach to transaction management.

Figure 7 – Current architecture of Carbon 21.



Adapted from (Carbono21 2023)

Carbon 21 is a platform that harnesses existing legislation on ecological compensation, particularly by associating a "compensation capability" with NFTs representing land. This association requires the registration of the corresponding area at the administrative bodies responsible for enforcing compensation laws, such as IBAMA, and Cetesb in Brazil. These governmental bodies provide Carbon 21 users with the necessary legal proof to verify the validity of the compensation capability.

Initially, the Carbon 21 blockchain project may involve Carbon 21 as the main organization, ensuring the platform's security and reliability through robust log mechanisms. It periodically stores the tail of the blockchain in IPFS, which enables blockchain auditing, retrieval, and validation mechanisms. This brings the benefit of a decentralized and public network, as periodic blockchain information is stored on the IPFS public network, allowing anyone to consult and validate it. Although Carbon 21 operates a private blockchain, it incorporates auditing features commonly found in public blockchains. To establish the blockchain network, each participating organization, including Carbon 21, IBAMA, and Cetesb, runs three nodes:

1. Certification Authority (CA): The CA node is responsible for issuing and managing participants' identities within their respective organizations. It ensures the integrity and authenticity of the identities associated with the blockchain transactions.

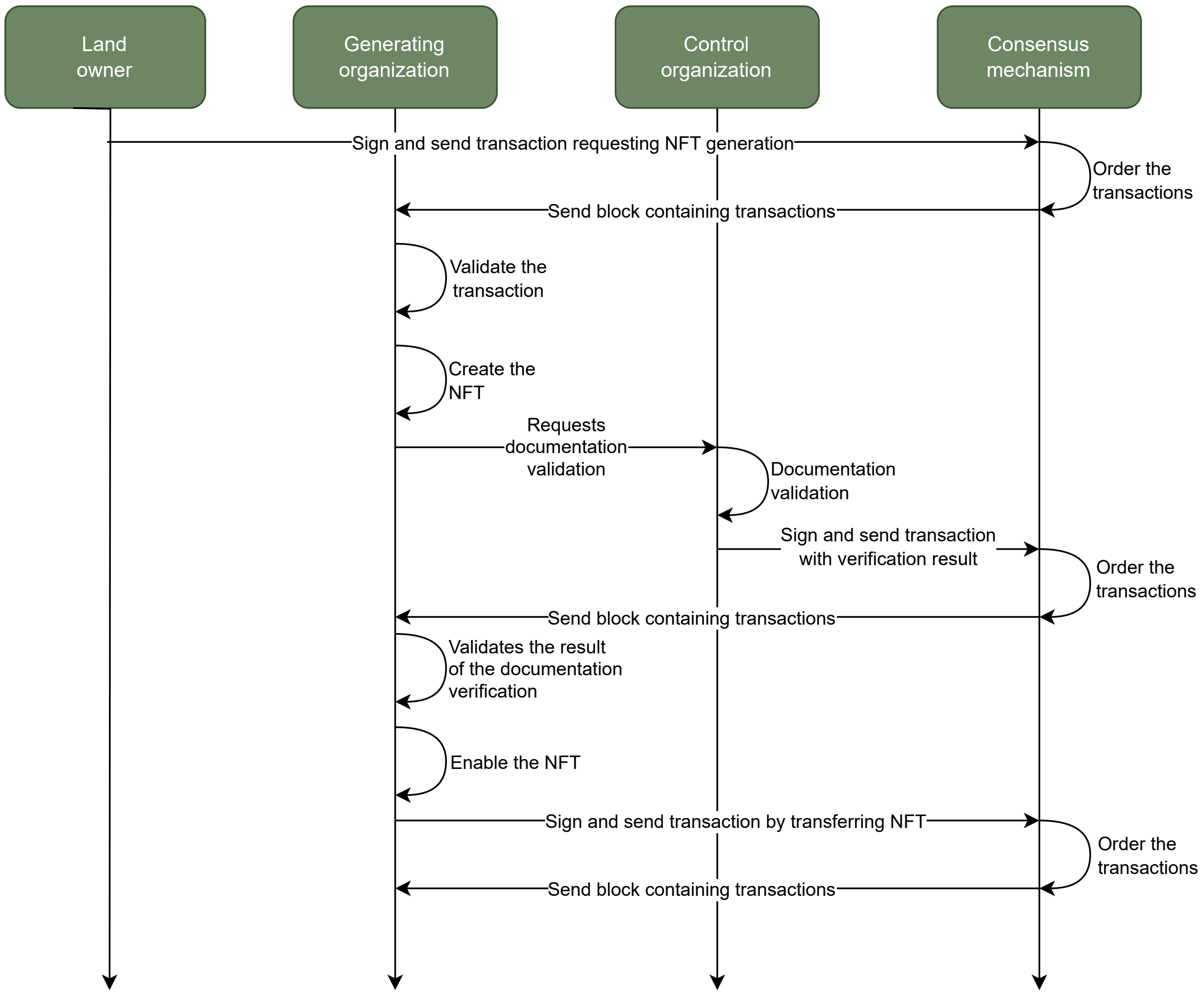
2. Peer: The Peer nodes validate and execute transactions within the blockchain network. They verify transactions' authenticity, consistency, and compliance against

predefined rules and smart contracts. Peers play a crucial role in achieving consensus within the network.

3. Database (DB): The DB node stores the current states of the blockchain, maintaining a record of all transactions and their associated data. It provides a reliable source of truth for the blockchain participants and ensures data consistency across the network.

By running these three nodes, each organization contributes to the overall security, transparency, and functionality of the Carbon 21 blockchain. The collaboration between Carbon 21, IBAMA, and Cetesb and potentially other organizations creates a robust and decentralized ecological compensation ecosystem, leveraging blockchain technology's power to drive environmental sustainability. Moreover, every organization involved in the blockchain will have a different responsibility in creating NFTs and other operations. Figure 8 shows the sequence diagram with all the operations and dependencies of the organizations to generate an active and valid NFT within the blockchain.

Figure 8 – Carbon 21's diagram.


Land owner
Generating organization
Control organization
Consensus mechanism
Sign and send transaction requesting NFT generation
Order the transactions
Send block containing transactions
Validate the transaction
Create the NFT
Requests documentation validation
Documentation validation
Sign and send transaction with verification result
Order the transactions
Send block containing transactions
Validates the result of the documentation verification
Enable the NFT
Sign and send transaction by transferring NFT
Order the transactions
Send block containing transactions

Adapted from (Carbono21 2023)

Effectively creating the NFT is one of the system's most crucial baseline operations. However, other operations are just as significant, performed in a considerably larger volume because they are public access operations. Below is a full list of operations grouped by type of user:

1. Land owners:
   a) Request NFT generation: Requests Carbon 21 and control organization to generate NFT, this operation will create the NFT with documentation validation pending status;
   b) Request NFT activation: Requests the NFT activation after its creation, so the control organization will validate the documentation to activate it or reject the operation;
   c) NFT list for sale: Operation to list all NFT available for sale on the Carbon 21 platform.
   d) Purchase NFT listed: The operation purchases an NFT that is available for sale, Carbon 21 will validate the operation and transfer the token to the new owner;
   e) Transfer FT/NFT: Direct operation for an owner to transfer the token to another user, without having to place the NFT for sale and the other user making the purchase. It is also possible to transfer FT so that another user has a positive balance to carry out operations on the platform, such as buying an NFT; and
   f) NFT revalidation: Requests revalidation of the NFT to ensure it is valid for auditing purposes.
2. Buyers (Companies that need compensation, investors, speculators, etc.):
   a) Request NFT compensation: Requests the compensation of the preserved area according to NFT to generate FT;
   b) NFT list for sale: Operation to list all NFT available for sale on the Carbon 21 platform;
   c) Purchase NFT listed: The operation purchases an NFT that is available for sale, Carbon 21 will validate the operation and transfer the token to the new owner;
   d) Transfer FT/NFT: Direct operation for an owner to transfer the token to another user, without having to place the NFT for sale and the other user making the purchase. It is also possible to transfer FT so that another user has a

positive balance to carry out operations on the platform, such as buying an NFT; and

e) NFT revalidation: Requests revalidation of the NFT to ensure it is valid for auditing purposes.

3. Control organization (e.g., Cetesb/IBAMA):

a) Endorse transactions: is a process where the beneficiary transfers ownership and rights of the NFT to a third party;

b) Execute RAFT node: ensures that all nodes eventually agree on the order and content of the replicated log, allowing for a consistent state across the distributed system;

c) Validate documentation: validates documentation to ensure that the NFT is valid, activating it on the platform;

d) Sign NFT issue and activation transactions: If the NFT is validated, it is possible to sign, activate and transfer ownership to the user who created it; and

e) Stores ledger: it refers to maintaining a distributed and immutable record of all transactions and data associated with a blockchain network. The ledger serves as a comprehensive and transparent history of transactions, ensuring the integrity and consistency of the data.

4. Carbon 21

a) Endorse transactions: is a process where the beneficiary transfers ownership and rights of the NFT to a third party;

b) Execute RAFT node: ensures that all nodes eventually agree on the order and content of the replicated log, allowing for a consistent state across the distributed system;

c) Issues NFTs: It effectively creates the NFT within the platform to be validated, activated, and traded later;

d) Issues FTs: Effectively creates the FT within the platform as per the compensation request process;

e) Sign transaction: it adds a digital signature to the transaction data to verify its authenticity and integrity. It ensures that the transaction has been authorized by the rightful owner of the associated digital assets and prevents tampering during transmission or storage; and

f) Stores ledger: it refers to maintaining a distributed and immutable record of all transactions and data associated with a blockchain network. The ledger serves as a comprehensive and transparent history of transactions, ensuring the integrity and consistency of the data.

In this way, since several operations are among them with public access, it is necessary to guarantee that the system behaves with stability in a large volume of simultaneous requests. Knowing the limits within the blockchain structure with a base configuration (flavor) is essential to configure the environment within the expectations of using the system.

Considering these operations available to be performed on Carbon 21, the main ones to be considered in experiments to analyze the performance of the project were identified and listed in Section 3.3. The objective is to guarantee the correct operability of the system in case of excessive execution of private or public operations due to the ease of method calling by anonymous users or users registered on the platform.

In order to propose a solution to the problem presented in Section 2.5, it is necessary to survey some prerequisites. In this sense, the following were identified:

- Have access to a private or public computing cloud IaaS, which has VMs with flavor settings defined in this work; and
- Set up a relevant scenario with VMs with privileges to collect network traffic and collect information about the instances.

From the adoption of the prerequisites, it is possible to determine the FR, which aims to present the functionalities that the system must perform, and the NFR, which presents the behavior of the system, to perform of the experiments and solution of the defined problem:

- FR1: The environment must allow transactions to be carried out through API or automated mechanisms for the blockchain network;
- FR2: The sending of transactions must be collected its quantity and its hash for eventual verifications;
- FR3: Collect VM metrics such as processing, memory, networks (IO transmitted and received), storage (read and write) and transaction latency;
- FR4: The operations called must be related to a NFT method, i.e., it must validate or create a NFT;
- NFR1: The system must provide means of parameterizing the system, allowing it to be adapted to the characteristics of the experiment; and

- NFR2: The techniques adopted to capture metrics and transactions should not affect performance.

These requirements serve as a foundation for the subsequent development and evaluation of the system, ensuring that it meets the necessary functionalities and performance criteria.

3.2 RELATED WORKS

Software performance analysis is a topic of significant importance from a technological, administrative, and business point of view. However, the performance analysis of blockchain solutions utilizing NFTs remains a relatively underexplored topic in academic literature, as well as in commercial and open-source solutions dedicated to this area. For the present work, the research method adopted to identify related work is systematic mapping, which aims to obtain an overview of the research area and identify/quantify evidence (Keele et al. 2007). Based on the mapping results, the aim is to identify the works related to the research area in which the proposed problem fits and analyze whether they meet the defined functional requirements.

3.2.1 Related Work Selection

The search for related works started with the definition of a set of keywords to be applied in scientific search engines was defined. The keywords definition is intended to highlight the focus on performance analysis in blockchain applications that use NFTs.

In order to format the search expression the logical operators *AND* and *OR* were used, in addition to elaborating the phrasing using parentheses. Thus, the keywords used were *NFT*, *blockchain*, and *performance analysis*, forming the search expression: *(NFT OR blockchain) AND performance analysis*.

The search engines used to carry out this research were: Google Scholar, ACM Digital Library, Elsevier, IEEE Xplore, and Springer Link. Google Scholar, for conducting a broader search, was the engine that presented the highest amount of results.

3.2.2 Inclusion and Exclusion Criteria

Among the works resulting from the search, a set of inclusion and exclusion criteria were applied to delimit the results obtained. Criteria are summarized in Table 1.

Table 1 – Inclusion and Exclusion Criteria.

Inclusion Criteria	Exclusion Criteria
Works written in English or Portuguese	”Gray” literature
Articles, abstracts, book chapters and technical reports	Publications prior to 2012
Analysis of performance and security	

Source: The author.

Firstly, works written in English or Portuguese were defined as inclusion criteria. The second inclusion criterion establishes that the selected works must be articles, extended abstracts, book chapters, or technical reports. Finally, the papers should address the topic related to the analysis of performance and security of generation of NFTs. In the case of the identification of duplicated works, only the most recent result should be considered.

Exclusion criteria were ”gray” literature works (i.e., publications in blogs and journals without scientific rigor) and productions in a language other than the predefined ones. Another defined exclusion criterion concerned the publication date. The present work presents a performance analysis of blockchain applications using NFT, only the results with a publication date since 2014, the year it was first discussed and introduced in a whitepaper.

3.2.3 Search Results

The search was carried out in four predefined academic search engines, using the elaborate search expression, resulting in an initial total of 23213 results. Filters referring to the publication date and other inclusion and exclusion criteria were applied, reducing the result to 1552 works. The analysis of the results found that many of these works did not yet correlate with the proposed theme. Thus, a sample was selected with the first 100 results from each search engine, ranked by relevance. Then, the titles and keywords of these works were analyzed, resulting in a set of works for reading the abstract. After reading the abstracts, ten papers related to the proposed theme were selected for a full read, one of which was identified as related work. In addition to the identified works, one open-source tool was also considered in this analysis: Caliper. Thus, the following list presents the related works and tools identified:

1. **Performance Analysis of Consensus Algorithm considering NFT Transaction Stability** (MinYoun-A e LimDong-Kyun 2022): The proposed work presents the performance of various blockchain consensus algorithms, comparing and analyzing as a method to increase the transaction cost and processing time during NFT transactions and to increase the transaction stability requirements that occur during smart contract execution.

2. **Caliper** (Hyperledger 2023): Caliper is a blockchain benchmark tool, it allows users to measure the performance of a blockchain implementation with a set of predefined use cases. Hyperledger Caliper will produce reports containing performance indicators to serve as a reference when using the following blockchain solutions.

3. **Analysis of an Ethereum Private blockchain Network Hosted by Virtual Machines Against Internal DoS Attacks** (Battisti et al. 2022): The proposed work presents a resistance analysis of an Ethereum-based network hosted by VMs with a DoS attacks used to identify the impact on the standard VM flavor. It developed an environment for experiments using Ethereum configured with distinct consensus mechanisms: Raft, Istanbul Byzantine Fault Tolerance (iBFT), and Proof-of-Authority (PoA).

4. **Performance analysis of the Raft consensus algorithm on Hyperledger Fabric and Ethereum on cloud** (Battisti et al. 2023): Performance analysis of the Raft consensus mechanism based on its implementation in Hyperledger Fabric and Ethereum blockchain solutions.

The identified works do not fully resolve the issue of performance analysis of NFT projects. The article by MinYoun-A e LimDong-Kyun 2022, written only in Korean, does not cover in detail the experiments and results identified in Section 2.5. The Hyperledger 2023 reference, on the other hand, only technically addresses a benchmark tool called Hyperledger Caliper, without actually applying it to a project demonstrating results and detailing the experiments. The work by Battisti et al. 2022 addresses the analysis of performance in blockchain Ethereum-based networks in different consensus mechanisms, not being addressed NFT and Hyperledger, being the closest to what is expected to be performed in this work. Table 2 compares the functional requirements of the proposed solution and the related works identified.

Table 2 – Related works vs. Functional Requirements.

	(MinYoun-A e LimDong-Kyun 2022)	(Hyperledger 2023)	(Battisti et al. 2022)	(Battisti et al. 2023)
FR1	Yes	Yes	Yes	Yes
FR2	No	Yes	Yes	Yes
FR3	Partially, metrics: network reliability, TPS, and consensus algorithm stability	Yes	Yes	Yes
FR4	Yes	Yes, considering it is an agnostic tool	No	No

Source: The author.

The FR1, which expects that the environment must allow transactions to be carried out through API or automated mechanisms for the blockchain network, is satisfied by all. The FR2, on the other hand, expects the sending of transactions to collect its quantity and its hash for eventual verification, which is not satisfied by

MinYoun-A e LimDong-Kyun 2022 since it is not addressed at any point in the article. In addition, it partially meets FR3, as it does not address processing metrics such as memory, networks, and storage. Finally, the work by Battisti et al. 2022 and (Battisti et al. 2023) does not satisfy the requirement because it does not present results using NFT.

3.3 PROPOSAL

Based on the number of actions available to be performed within the project, as mentioned in Section 3.1, there is an opportunity to perform a Denial of Service (DoS) attack with real requests instead of an Secure Shell (SSH) attack as presented by Battisti et al. 2022 and Battisti et al. 2023. These attacks can call public operations by any user, thus being a relevant analysis to identify the potential of the architecture using a predefined flavor.

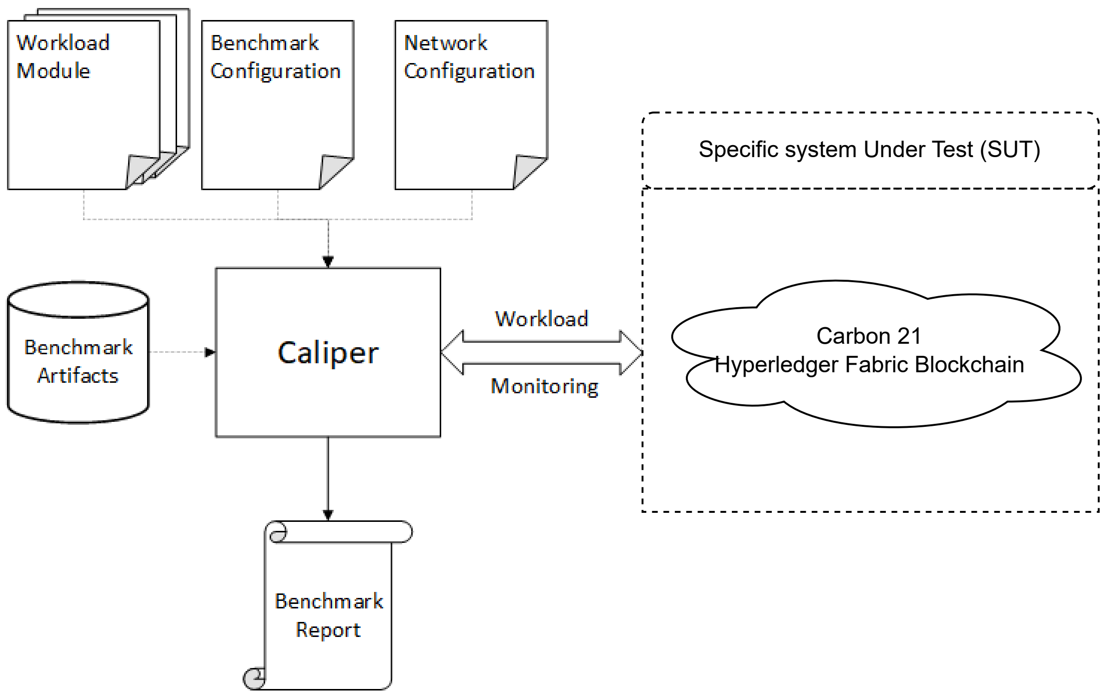
By calling these and other operations, it will be possible to identify the behavior of the system in Hyperledger and Interplanetary File System (IPFS), wherein a real environment where an intentional DoS attack occurs, two situations can occur as identified by Battisti et al. 2022:

1. It occurs from the accumulation of several transactions that the platform receives, which causes a high rate of network traffic and, as a result, latency occurs on the network. With the occurrence of latency in the blockchain network, consequently, delays occur in the validation and insertion of new blocks; and
2. This maintains the same line of thinking as the first situation. However, service unavailability or dropped packets that are queued for network traffic may result from this latency and reduced network traffic.

Although a blockchain network is not commonly applied in just one computing cloud, the choice was made due to the practicality of executing the experiments and isolation from other factors (e.g., background traffic) that make the analysis more subjective and complex. Furthermore, several parties could be involved in the development of the Carbon 21 blockchain, each playing a crucial role in the project's success, as seen in Figure 7. The primary organizations that comprise the Carbon 21 blockchain ecosystem include itself and potential partners such as Environmental Company of the State of São Paulo (Cetesb) and Brazilian Institute of Environment and Renewable Natural Resources (IBAMA). However, it is essential to note that additional organizations may also be involved beyond these initial partners as the project progresses. Additionally, an independent orderer is implemented to handle consensus mechanisms, transaction ordering, and distribution, ensuring a decentralized and impartial approach to transaction management.

Figure 9 shows Hyperledger Caliper, a benchmarking tool developed to measure the performance of blockchain platforms. It operates by configuring reference and network files to simulate specific workloads. Hyperledger Caliper uses workload modules to generate these loads, allowing for the assessment of performance metrics such as latency, throughput, and resource usage. For this study, Hyperledger Caliper will be configured to create Non-Fungible Tokens (NFTs) under DoS attack conditions, capturing critical metrics to evaluate the blockchain system's robustness and scalability in high-demand scenarios.

Figure 9 – Current architecture of Carbon 21.



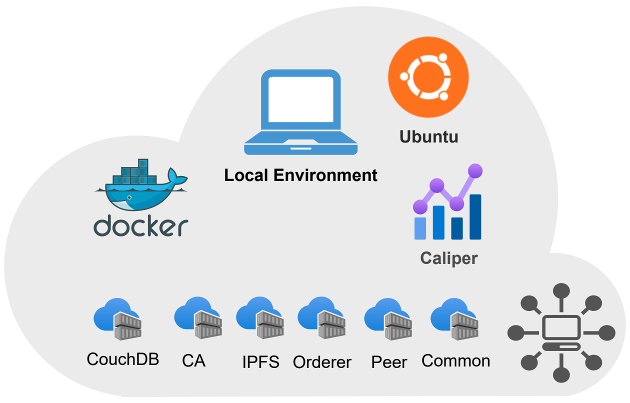
Source: The author.

The operational flow of Hyperledger Caliper involves several key steps. First, benchmark and network configuration files are set up to define the parameters and topology of the test. Next, workload modules generate transactions and interactions, simulating real-world usage. During the execution phase, these transactions are processed by the blockchain network. Finally, performance metrics such as latency, throughput, and resource usage are collected and analyzed to assess the blockchain system's performance under the given conditions.

To carry out the proposed analysis, three scenarios are developed, in which, in general, they differ only by the consensus mechanisms that are applied. This choice is because the literature shows the importance of a consensus mechanism on a blockchain network, which is a relevant factor for carrying out the experiments. Thus, the following scenarios are presented:

1. Scenario I - Local environment with 50 Transactions Per Second (TPS): Experiment executed considering 50 TPS in a local computer (desktop/laptop) in a controlled environment, illustrated by Figure 10
2. Scenario II - Local environment with 100 TPS: Experiment with the same flavor, environment, and local computer of Scenario I (Figure 10), but double the transaction rate to 100 TPS.

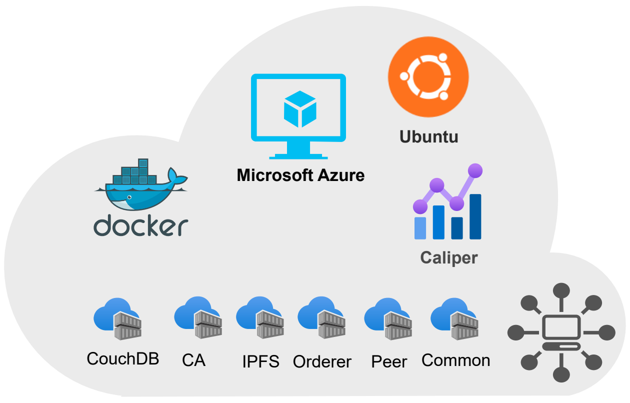
Figure 10 – Architecture Scenario I and II - Local Environment.



Source: The author.

3. Scenario III - Azure Cloud with 50 TPS: Experiment executed in a dedicated Virtual Machine (VM) hosted in the Microsoft Azure Cloud (Figure 11).
4. Scenario IV - Azure Cloud with 100 TPS: Experiment executed in a dedicated VM hosted in the Microsoft Azure Cloud (Figure 11).

Figure 11 – Architecture Scenario III and IV - Cloud Environment using Microsoft Azure.



Source: The author.

3.4 KEY COMPONENTS

The transparency and auditability mechanisms are essential features to be made available to end users so that they can continuously verify the correct operation of the system. For this, the system should consider two key components:

1. Hyperledger: specifically Hyperledger Fabric, plays a crucial role in recording and securing NFT and Fungible Token (FT) transactions. Its federated blockchain structure ensures that all actions performed within the NFT ecosystem are recorded. Furthermore, the Raft, a consensus mechanism that ensures efficient performance while reducing computational costs; and

2. IPFS: it is a decentralized, public, and open system for consolidating platform updates and providing additional transparency to NFT and FT operations;

The combination of Hyperledger Fabric and IPFS allows for efficient transparency mechanisms. Hyperledger records actions on a federated blockchain, providing a detailed audit trail, while IPFS consolidates updates into a public, decentralized open system. This consolidation minimizes computational costs while maintaining accessibility and transparency. On the other hand, the Raft consensus mechanism improves performance while ensuring the integrity of NFT transactions.

3.5 TEST PLAN

Hyperledger Caliper was used in this project to collect and analyze metrics. It is a tool that allows users to measure the performance of a blockchain implementation with a set of predefined use cases. Hyperledger Caliper generates practical reports packed with a variety of performance indicators that serve as a reliable reference for our project. The metrics available are:

1. Success Rate: Measure all successful, failed and unfinished transactions for a test cycle.

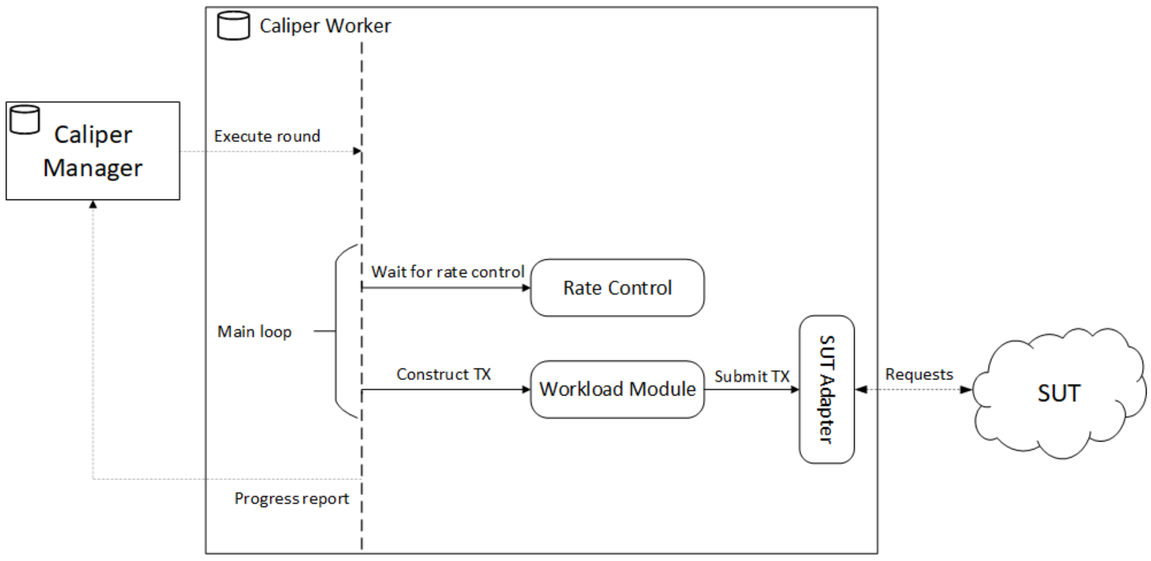
2. Transaction & Read Latency: Measure the time for an issued transaction to be completed and a response being available to the application that issued the transaction. The test cycle's maximum, minimum, and average latency is provided.

3. Transaction & Read Throughput: Measure the flow rate of all transactions through the system, in transactions per second, during a cycle.

4. Resource consumption: See the maximum and minimum memory and CPU resource consumption and Input/Output (IO) traffic during a cycle for each component process.

For the experiment implementation, we customized the Hyperledger Caliper source[1] code to provide these metrics on a timeline, offering minute-by-minute values instead of absolute values. This meticulous customization allows for the generation of detailed graphs, a powerful tool that shows metrics over time. These graphs significantly enhance the analysis of blockchain performance, particularly under stress conditions, as presented in Figure 12.

Figure 12 – Hyperledger Caliper Functional Flow.



Source: The author.

The Caliper Manager orchestrates the benchmarking process. It initiates rounds of tests by sending commands to the Caliper Worker and receives progress updates throughout the execution. The Caliper Worker executes the benchmarking tasks as directed by the Caliper Manager. It manages the main loop of the benchmarking process, generating and submitting transactions to the blockchain system. The rate control regulates the transaction generation rate to ensure it adheres to the specified benchmark conditions. It controls the timing for when the Workload Module can construct transactions. The Workload Module constructs the transactions based on the benchmark configuration. Once the Rate Control permits, it generates these transactions and sends them to the System Under Test (SUT) Adapter. Acting as an interface, the SUT Adapter converts the transactions from Caliper into a format that the SUT can process and then submits these transactions to the SUT. After this entire process, which Caliper manages, metrics are captured at a specific time parameterized by the developer. For this project, it was configured to capture the results every second and stored in files for later reading and analysis.

Two experiments were conducted to evaluate the performance of the blockchain system under different transaction loads. The first experiment involved executing 50

[1] Available on https://github.com/vitorebatista/caliper

TPS to mint NFTs. The second experiment increased the load to 100 TPS, both running in a local and cloud environment. These experiments aimed to analyze the system's robustness and scalability under varying stress conditions, providing insights into its ability to handle high-demand scenarios. Additionally, the experiments sought to statistically understand the correlation between the transaction load and the performance metrics, offering a comprehensive assessment of the system's behavior under different conditions.

In a real environment in which an unintentional DoS attack occurs, two situations typically happen:

1. It occurs from the accumulation of several transactions that the platform receives, which causes a high rate of network traffic and, as a result, latency occurs on the network. Based on the occurrence of latency in the blockchain network, consequently, delays in validation and insertion of new blocks occur; and
2. This maintains the same *modus operandi* of the first situation, but leads to service unavailability or dropped packets that are queued for network traffic may result from this latency and reduced network traffic.

The test rounds were performed on a blockchain network with nodes created using Docker in a local environment. The environment was hosted on a laptop with 8 vCPUs, 16Gb of memory, and 256Gb of storage space, ensuring a controlled and consistent setting for the experiments.

Although the use of a blockchain network is not commonly applied in a single computing cloud, the choice was made due to the practicality of executing the experiments and isolation from other factors (e.g., background traffic) that make the analysis more subjective and complex. The architecture was built from this question through three main components: network, router, and VMs. In terms of the applied network architecture, it is an overlay network that connects via virtual links and switches.

Furthermore, experiments were executed in a Microsoft Azure cloud environment, which was possible by encouraging credits to be a master's student at UDESC. The credit was US$190.00 to be used within one month, which made it possible to run in parallel the experiments with several VMs with the configurations defined in scenarios III and IV.

With the execution of the experiments it will be possible to have three main metrics, the first represents the number of transactions performed and lost by the system, the second represents the number of transactions performed per minute, and the third represents the machine statistics (processor, memory, network and storage I/O consumption of the system during the execution of the transaction flood. Thus, the instance resources will be monitored:

- Processing: Monitor the percentage of processor occupancy per minute during the queries period;
- Memory: Monitor the use of RAM occupied per minute during the queries period;
- Network traffic: Monitor the volume of traffic on the link during the queries;
- Number of transactions: Monitor the number of transactions sent per minute;
- Latency: monitor how long each transaction takes to get from source to destination; and
- Storage IO read/write: the amount of data read from and written to storage per minute during the period.

Based on the monitoring collection of this information, it is possible to extract metrics in which statistical analyses will help to understand the system's behavior in one or more determined test scenarios. In addition, one hypothesis is defined for the experiments: It is an intentional DoS attack, i.e., the operations to be performed have the objective of simulating actions expected by users within the system. It is essential to point out that none of the VMs have virtual memory resources swap enabled, they only use the main memory because of the objective of depleting their resources. In this sense, the following experiments are performed in each of the scenarios:

- **Experiment I**: Performs the execution considering 50 TPS through a user/client registered on the blockchain network to reduce network traffic or service unavailability, aiming at identifying, from the monitoring of resources and transactions, the stability of the network in the event of a large number of valid requests in a short time (i.e., a DoS due the VM flavor limitations); and
- **Experiment II**: Double the number of transactions to 100 TPS. This experiment aims to understand how quickly the system degrades compared to Experiment I, observing the rate at which system performance deteriorates under increased load.

In order to analyze the interplay between experiments and functional requirements within the context of blockchain and NFTs, the following list outlines the association between Experiments I and II conducted using the presented architecture, with reference to the functional and non-functional requirements.

- FR1: The environment must allow transactions to be carried out through Application Programming Interface (API) or automated mechanisms for the blockchain network;
  - The transactions will be fired using the API by Hyperledger Caliper
- FR2: The sending of transactions must be collected its quantity and its hash for eventual verifications;
  - The project has mechanisms to verify by IPFS and the CouchDB.
- FR3: Collect VM metrics such as processing, memory, networks (IO transmitted and received), storage (read and write) and transaction latency;
  - The Hyperledger Caliper has the ability to collect all the metrics with the customizations made[2].
- FR4: The operations called must be related to a NFT method, i.e., it must validate or create a NFT;
  - The test suite using Hyperledger Caliper was created to mint specifically NFT
- NFR1: The system must provide means of parameterizing the system, allowing it to be adapted to the characteristics of the experiment; and
  - The Hyperledger Caliper is able to have different parameters, one used was the number of transactions (50 and 100 TPS.
- NFR2: The techniques adopted to capture metrics and transactions should not affect performance.
  - The metrics collected consider the data from the Docker, so the process outside was not considered.

Both experiments contemplate the exact relationships between Non-Functional Requirements (NFR) and Functional Requirements (FR), which makes the results of the experiments easy to analyze, compare, and correlate between them.

The execution of experiments with Caliper customizations produces CSV files containing essential data for result analysis. Each CSV file from a Docker container represents metrics (memory, CPU, network, and storage). Additionally, another file is generated with data on accumulated transactions—successful, failed, pending, and

[2] Available on https://github.com/vitorebatista/caliper

latency. The Caliper records all this data with second-by-second granularity. After the experiment, Python scripts are used to normalize the data and generate the graphs and analyses described in Section 4.

3.6 TESTBED SETUP

The blockchain network from Carbon 21 was created using Docker containers, making it simple and quick to configure and launch in GNU/Linux environments. In this sense, the blockchain was configured in two different environments, local and in the cloud.

To effectively evaluate the performance and security of the Carbon 21 blockchain network under a DoS attack, a testbed setup was designed to comprise both local and cloud environments to ensure a robust analysis of the system's capabilities.

3.6.1 Local environment

The local environment was set up on a laptop with the following specifications:

- Processor: Intel® Core™ i7-10510U CPU @ 1.80GHz - 8 Cores.
- Memory: 16Gb RAM.
- Storage: 256Gb SSD.
- Operating System: GNU/Linux Ubuntu 20.04 LTS.

The blockchain network was created using Docker containers, which simplified the configuration and deployment process. The local setup included essential blockchain components such as Certification Authority (CA), peers, and databases (CouchDB), all connected via a virtual network, illustrated in Figure 5.

The experiments were executed sequentially to ensure the reliability and accuracy of the collected data. The process involved the following steps:

- Complete execution of a test scenario.
- Collection and saving of all relevant performance and security metrics.
- Full restart of the environment, including:
  - Uninstallation of all project dependencies.
  - Clearing of caches.
  - Removal of any residual data that could influence subsequent tests.

Each experiment began with a clean slate by performing these steps, thereby eliminating any potential residual effects from previous tests. This meticulous approach allowed for the collection of a robust and reliable dataset, which is critical for accurate analysis. So, this process facilitated controlled experiments with transaction rates of 50 TPS and 100 TPS to analyze the performance impact under varying loads.

3.6.2 Cloud environment

This test environment was built as a project in the Microsoft Azure Cloud; the approach chosen was the use of two flavors of VMs, considering the credits available to set up and run the Microsoft Azure services. The testbed was built on the Microsoft Azure platform for the cloud environment. This setup leveraged the flexibility and scalability of cloud services to simulate a more extensive and distributed network. Two VM configurations were utilized based on the available credits:

- VM Configuration for 50 TPS:
  - vCPU: 2.
  - Memory: 8Gb RAM.
  - Storage: 512Gb SSD.
  - Operating System: GNU/Linux Ubuntu 20.04 LTS.
- VM Configuration for 100 TPS:
  - vCPU: 8.
  - Memory: 32Gb RAM.
  - Storage: 512Gb SSD.
  - Operating System: GNU/Linux Ubuntu 20.04 LTS.

The process involved the following steps:

- Complete execution of a test scenario.
- Collection and saving of all relevant performance and security metrics.
- Full restart of the environment, including:
  - Uninstallation of all project dependencies.
  - Clearing of caches.
  - Removal of any residual data that could influence subsequent tests.

By performing these steps, we ensured that each experiment began with a clean slate, eliminating potential carry-over effects from previous tests.

3.7 CHAPTER CONSIDERATIONS

This chapter defines the requirements for developing an experimental environment to analyze the security and performance of private blockchain networks, specifically concerning DoS attacks in an Infrastructure as a Service (IaaS) cloud provider. Additionally, previous research work was identified that may differ in purpose but provide valuable references for the analysis.

Furthermore, a significant and trustworthy approach for this research was proposed, which involves utilizing the local and cloud environment with different scenarios and experiments. This chapter has provided a thorough and detailed foundation for analyzing the performance and security of the Carbon 21 blockchain network, ensuring confidence in the results.

4 IMPLEMENTATION AND RESULTS

During the entire experiment's execution process, the instance resources and the blockchain network were monitored using Hyperledger Caliper to identify possible instabilities or oscillations in the network, storage, memory, and processor. To ensure that the experiment results are reliable, six replications were performed and two rounds, the first one with 50 TPS and the second with 100 TPS, according to the environment to be executed (local and cloud). Since the measurements of the response variable are subject to random variations, replications of an experiment are used to determine the impact of measurement error on the response variable. According to Lilja 2000, two replications should be sufficient to indicate both the magnitude of the errors and the interaction.

As each experiment produces a CSV file for each Docker container, an additional file containing transaction data (success, failure, pending) and latency, Exploratory Data Analysis (EDA) techniques are essential to interpret these results. Developed primarily by John Wilder Tukey in the 1970s, EDA is akin to detective work, where numerical and graphical clues are analyzed to uncover patterns and insights (Box Plot 2008). A comprehensive statistical analysis was performed to understand the experiment results, generating various graphs. One of the key graphical techniques used was the Box Plot, a type of diagram used to compare several sets of observations (Hofmann e Kafadar 2017) illustrating:

- The measure of central tendency, primarily the median.
- The variability of the data.
- The symmetry of the data distribution.

Additionally, a regression model graph was used to analyze the data. The regression plot provides several benefits (Ansari e Nassif 2022), including:

- Identifying relationships between variables.
- Highlighting trends and patterns over time.
- Allowing for predictions and insights into future behavior based on historical data.

This approach helps to visually summarize and compare the performance metrics across different Docker containers, providing a clear understanding of the system's behavior under various conditions.

4.1 RESULTS ON LOCAL ENVIRONMENT

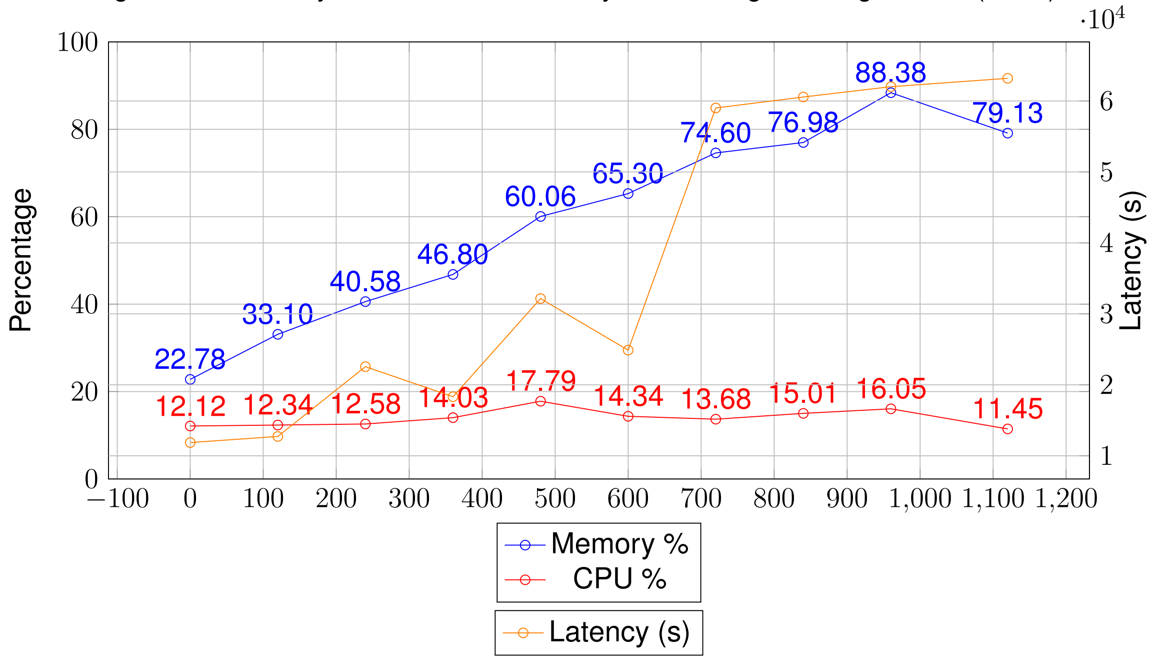
The experiments conducted in the local environment, as detailed in Subsection 3.6.1, were designed to evaluate the performance of the Carbon 21 blockchain network under controlled conditions. This section presents the results of these experiments, focusing on key metrics such as CPU usage, memory consumption, network IO, and transaction latency for 50 and 100 TPS.

4.1.1 Results generating 50 TPS on local environment

Regarding the consumption of system processing, it is possible to observe the efficiency of the blockchain, as it does not require much computational power since, during the entire execution of the attack, the processor remained stable with few changes in consumption during the whole period of attack of 1120 minutes, reaching almost 19 hours of execution until the service crashed. In Figure 13, it is possible to observe that the CPU remained stable throughout the experiment, not exceeding more than 18% of the VMs total capacity, even increasing the number of transactions.

In memory consumption, it is possible to notice that memory tends to accumulate during execution; this is one reason the service stops working at a certain point in the execution, thus reducing memory consumption at the last point since the container stopped responding. Table 3 highlights some measurement points of Figure 13, in addition to having the result of completed, unfinished, failed, and total transactions.

Figure 13 – Memory %, CPU % and Latency over Time generating 50 TPS (Local).



Source: The author.

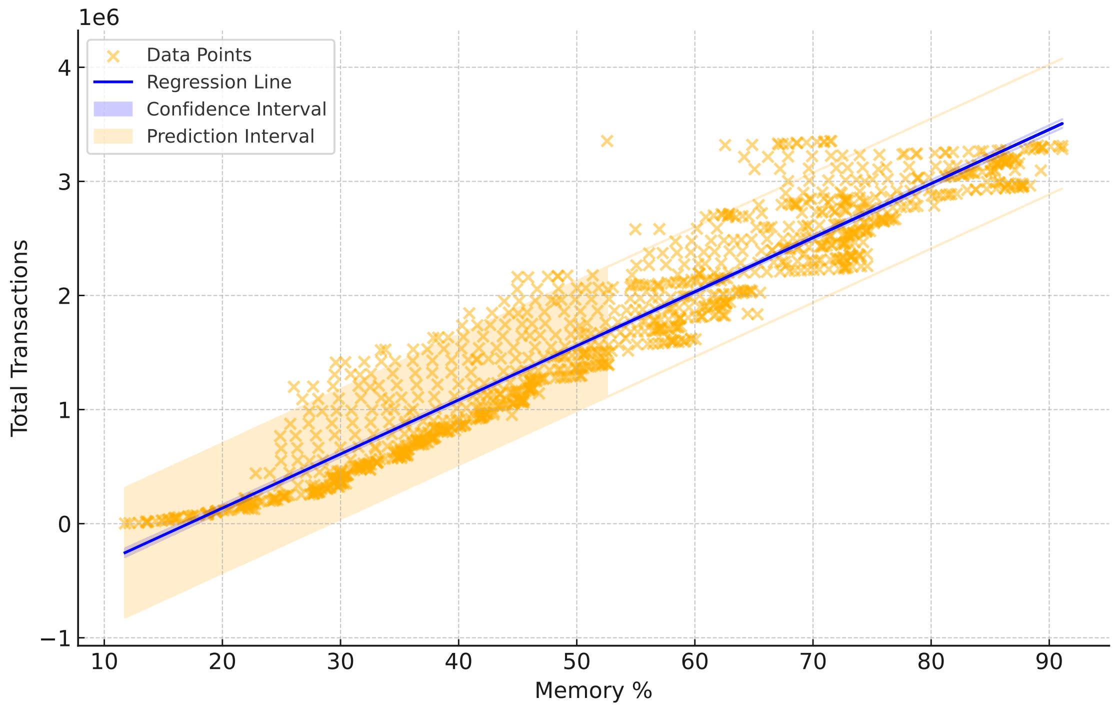
Table 3 – Latency, Memory, CPU, and Transactions by thousand in 50 TPS (Local).

Minute	0	120	240	360	480	600	720	840	960	1120
Latency (ms)	11866	12724	22557	18322	32151	24885	59012	60559	62013	63192
Memory %	22.78	33.10	40.58	46.80	60.06	65.30	74.60	76.98	88.38	79.13
CPU %	12.12	12.34	12.58	14.03	17.79	14.34	13.68	15.01	16.05	11.45
Total Failed Transactions	0.0	0.0	0.0	0.0	0.0	2.549	22.104	94.771	96.425	305.793
Total Unfinished Transactions	2.99	9.291	19.024	31.71	50.432	74.957	119.029	227.746	250.764	483.742
Total Successful Transactions	358.818	718.86	1078.832	1438.802	1798.868	2153.297	2496.733	2784.063	3142.397	3250.781
Total Transactions	358.823	718.868	1078.86	1438.89	1798.881	2158.898	2518.852	2878.848	3238.884	3559.71

Source: The author.

After developing a linear regression model, it is helpful to know how well the equation models the measured data. It was knowing how strong the linear correlation between the input and output is essential. The coefficient of determination (r^2), and its square root, called the correlation coefficient (*r*), are quantitative measures of this observed linearity (Lilja 2000). Figure 14 shows the regression model with $r = 0.954$ and $r^2 = 0.911$. The value of r indicates a strong positive correlation between memory usage percentage and total transactions. An r value close to 1 suggests that as memory usage increases, total transactions also increase in a linear relationship. Moreover, the r^2 value of approximately 0.911 means that about 91.1% of the variance in total transactions can be explained by the memory usage percentage, showing our regression model fits the data well (Lilja 2000).

Figure 14 – Regression model with the total transactions and memory (Local and 50 TPS).

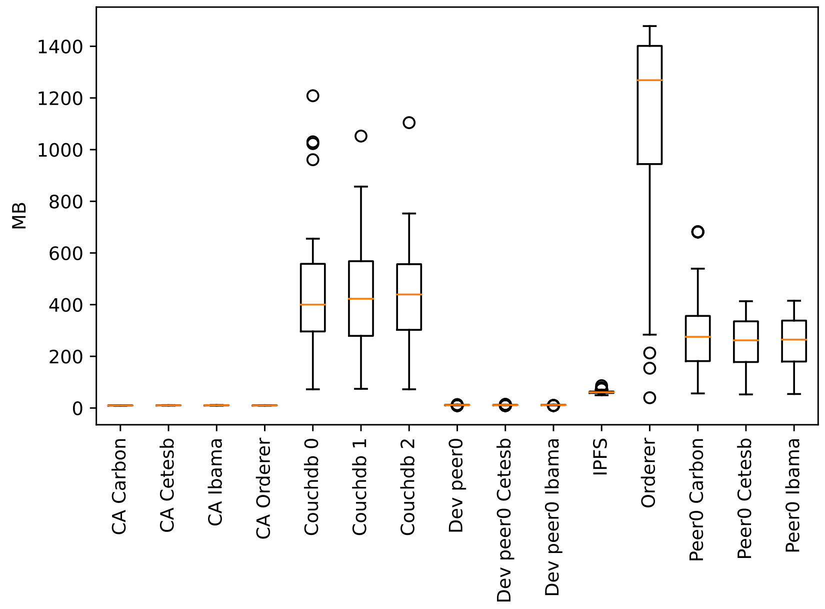


Source: The author.

During the 50 TPS experiment, the blockchain network maintained stable CPU usage throughout the 1120-minute execution, and CPU consumption never exceeded 18%. However, memory consumption gradually increased over time, eventually leading to service failure. The total number of successful transactions reached approximately 3,250,781 before the system crashed.

Figure 15 is the box plot that provides a visual summary of the distribution of memory usage for all containers. It is possible to see that all the CA, Dev Peer, and IPFS have relatively insignificant memory usage, indicating consistent usage with few outliers. However, all CouchDB show higher memory usage, especially the Orderer, with larger Interquartile Ranges (IQRs). There are noticeable outliers, indicating some instances where memory usage is significantly higher.

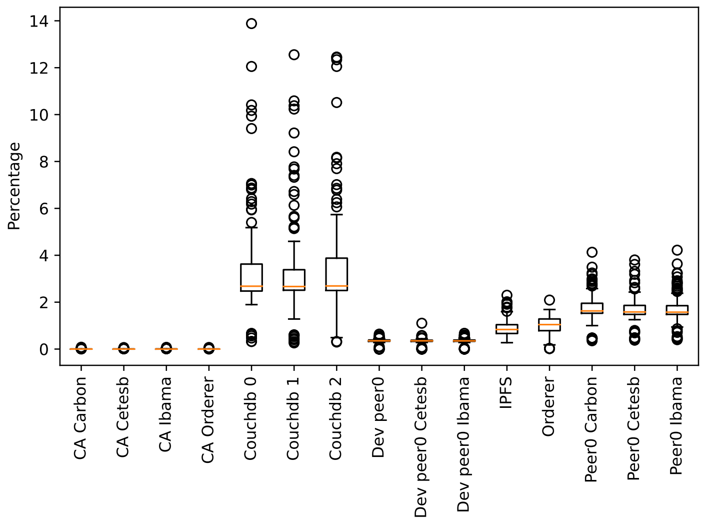
Figure 15 – Memory Usage in the local environment in 50 TPS.



Source: The author.

When analyzing CPU consumption, Figure 16 reveals higher usage and some variability and most significant outliers on all the CouchDB, IPFS and Peers, indicating occasional spikes in CPU usage. Conversely, CA and Dev Peers containers have stable CPU usage with few variations.

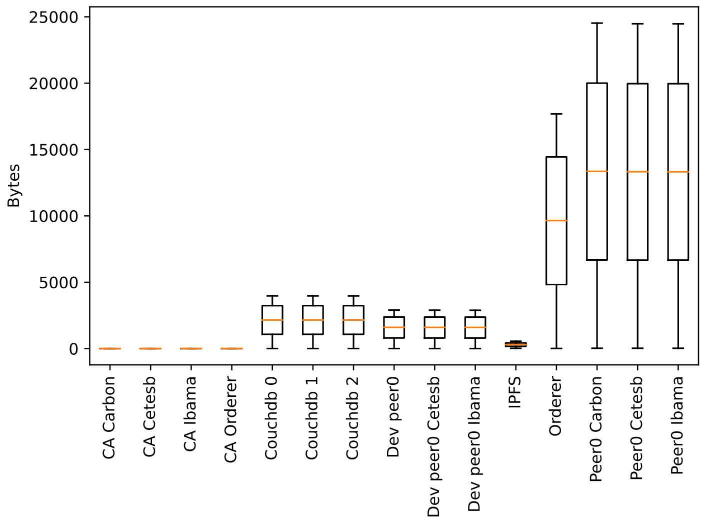
Figure 16 – CPU Usage in the local environment in 50 TPS.



Source: The author.

For the Network IO received usage, illustrated in Figure 17, it is possible to see the Orderer and Peers instances handle significantly more network traffic than the others, suggesting their critical role in the system. The higher variability and outliers indicate these containers might experience periodic surges in network activity, which could affect overall system performance. Then CouchDB and Dev Peers have relatively lower usage, with insignificant usage for all the CA.

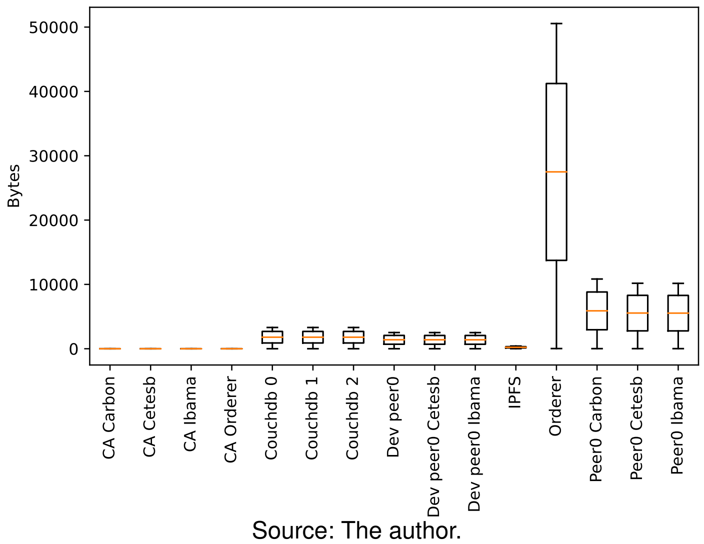
Figure 17 – Network IO received in the local environment in 50 TPS.



Source: The author.

Still analyzing the use of the network, but now for the transmitted one, Figure 18 presenting all the CA display insignificant and consistent network IO transmission with minimal variability and no significant outliers, similar with CouchDB. In contrast, Orderers have higher medians and more variability than Peers, indicating they handle more network traffic and have more fluctuating workloads.

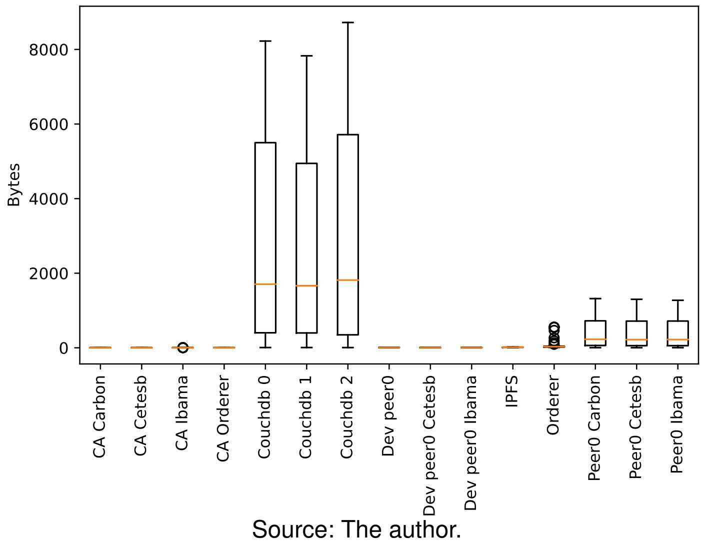
Figure 18 – Network IO transmitted in the local environment in 50 TPS.



Source: The author.

When analyzing the storage IO metrics, explicitly focusing on the bytes read over time (Figure 19), CouchDB instances handle significantly more block IO, suggesting their critical role in storage needs. The higher variability and outliers indicate these containers might experience periodic surges in block IO, which could affect overall system performance, like the increase of latency. The other instances, like the CA and Dev Peers, display relatively lower and consistent block IO with minimal variability and no significant outliers.

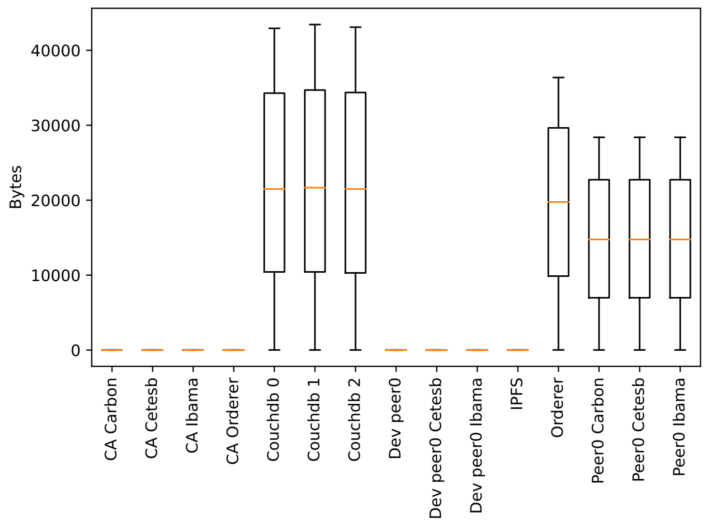
Figure 19 – Storage IO read in the local environment in 50 TPS.



Source: The author.

Finally, Figure 20 presents the write storage IO data. The result is very similar to Figure 19, with the CouchDB higher median values, but now with also Orderer and Peers too, indicating they consistently write more block IO bytes. Furthermore, the all the CA, IPFS, and Dev Peers display insignificant and consistent block IO writes with minimal variability and no significant outliers.

Figure 20 – Storage IO write in the local environment in 50 TPS.



Source: The author.

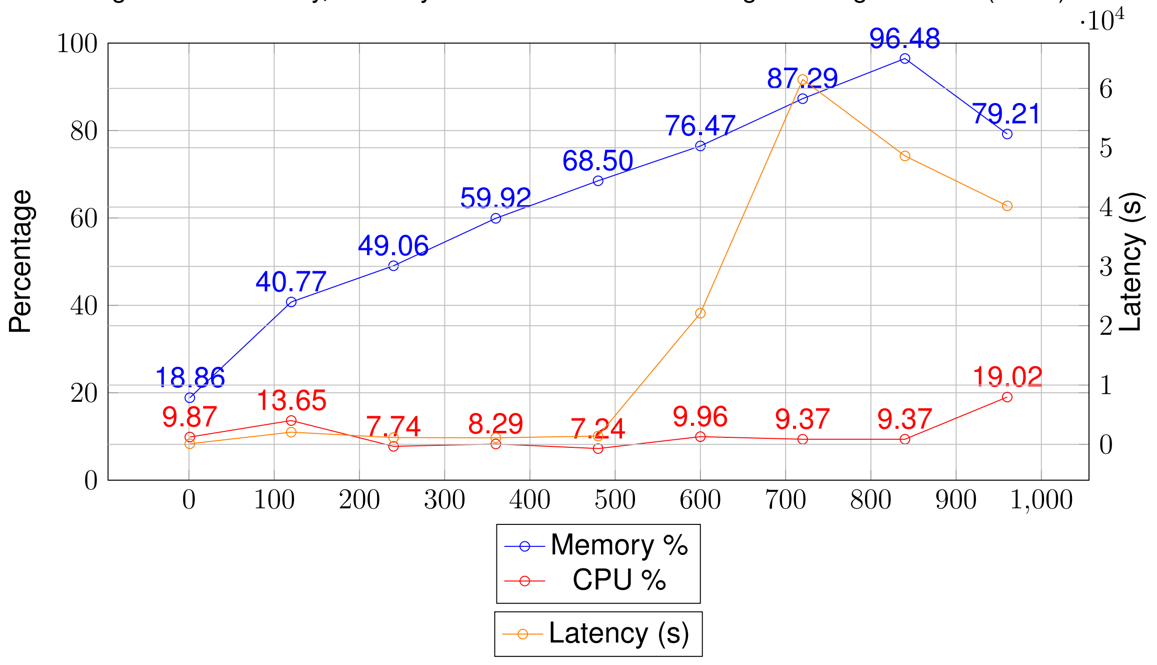
After being able to observe all of this information, we can conclude that in this experiment:

- Network IO - The Orderer and Peer containers handle significantly more network traffic, as indicated by higher median values for both bytes received and transmitted. These containers are critical in managing network communications within the system. The higher variability and frequent outliers observed in these containers suggest periodic network traffic surges.
- Storage IO - The CouchDB, Orderer, and Peer containers consistently show higher median values and more significant variability in both bytes received and written. This indicates these containers are heavily involved in storage IO operations.
- CPU and Memory - The CouchDB containers handle substantial storage IO and will likely significantly impact CPU and memory usage. The Orderer also requests higher usage of memory than others. The observed outliers and variability in CouchDB and Orderer metrics suggest potential performance bottlenecks that warrant further investigation and optimization efforts.

4.1.2 Results generating 100 TPS on local environment

The 100 TPS round execution ran for 960 minutes (16 hours) until the blockchain service crashed, making it no longer possible to process new transactions successfully. Figure 21 shows a similar result to the previous experiment, as as execution time passes there is an increase in memory usage and the consistency of CPU consumption.

Figure 21 – Latency, Memory % and CPU % over Time generating 100 TPS (Local).



Source: The author.

Table 4 evinces, in addition to the memory and CPU results, the transaction totals over time. Similar to Experiment I, this experiment shows a noticeable trend of memory consumption increasing steadily over time. This buildup in memory usage contributes to the service failure, as seen by a sudden drop in usage at the final stage when the container stops responding.

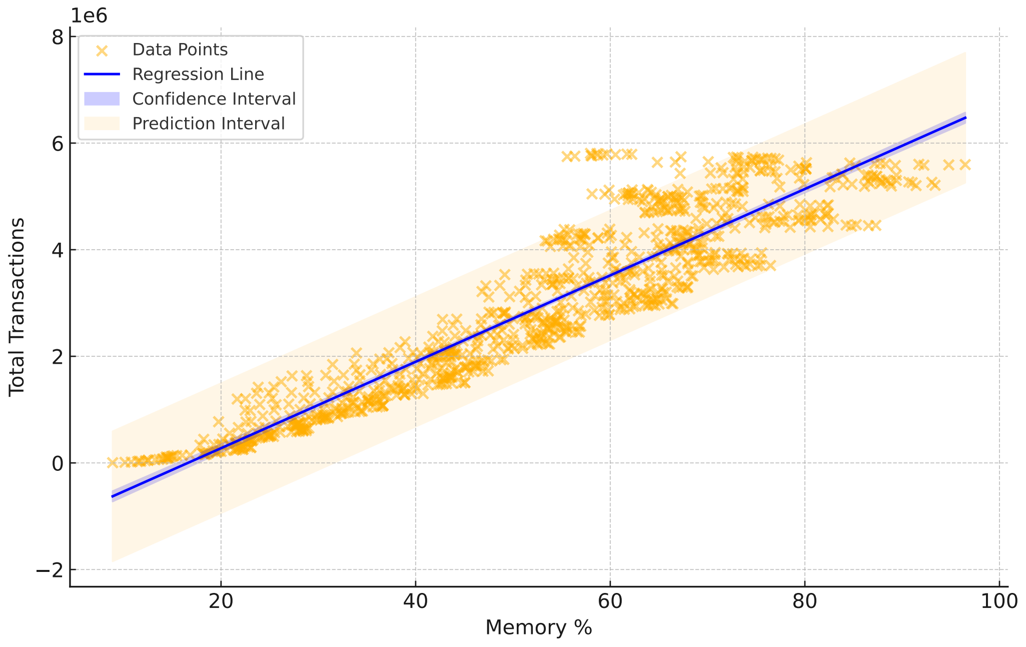
Table 4 – Latency, Memory, CPU, and Transactions by thousand in 100 TPS (Local).

Minute	0	120	240	360	480	600	720	840	960
Memory %	18.86	40.77	49.06	59.92	68.50	76.47	87.29	96.48	79.21
Latency (ms)	104	2,060	1,150	1,110	1,403	22,110	61,492	48,618	40,195
CPU %	9.87	23.65	7.74	8.29	7.24	9.96	9.37	9.37	19.02
Total Failed Transactions	0.00	8.15	116.58	151.24	157.24	187.63	265.60	321.18	381.16
Total Successful Transactions	718.32	1424.95	2013.17	2708.05	3419.22	4107.48	4747.86	5420.97	5420.97
Total Unfinished Transactions	67.36	177.84	366.73	585.43	780.43	980.14	1221.76	1591.97	1659.85
Total Transactions	718.45	1435.12	2134.74	2861.88	3581.16	4299.77	5019.15	5747.19	5806.96

Source: The author.

Figure 22 shows the regression model with $r = 0.929$ and $r^2 = 0.864$. It means that approximately 86.4% of the variance in total transactions can be explained by Memory %. Like the 50 TPS result, the 100 TPS round indicates a strong relationship between Memory % and total transactions. In the 100 TPS experiment, the system ran for 960 minutes before crashing, showing similar resource usage patterns as the 50 TPS experiment. While CPU consumption maintained a relatively consistent pace, memory usage exhibited a steady and concerning upward trend. The total number of successful transactions reached approximately 5,420,965 before the service disruption.

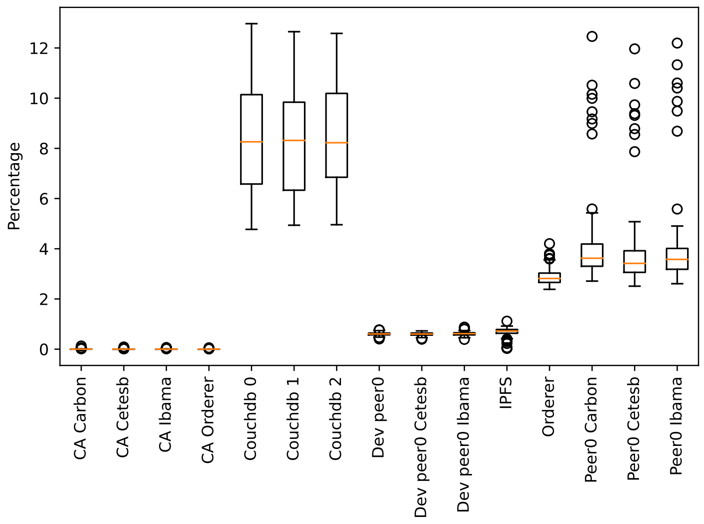
Figure 22 – Regression model with the total transactions and memory (Local and 100 TPS).



Source: The author.

When analyzing the container results in detail using the box plot, Figure 23 exposes most containers have stable CPU usage with few variations, except for the CouchDB and Peer containers, which show higher usage and some variability. The Peers container has the most significant outliers, indicating occasional spikes in CPU usage. All CA containers have shallow CPU usage, similar to the 50 TPS, close to 0%, with minimal variation and no significant outliers. The Dev peers and IPFS show moderate CPU usage if compared with CA, with a few outliers indicating occasional spikes in usage. The Orderer has consistent CPU usage with minimal variation and no significant outliers.

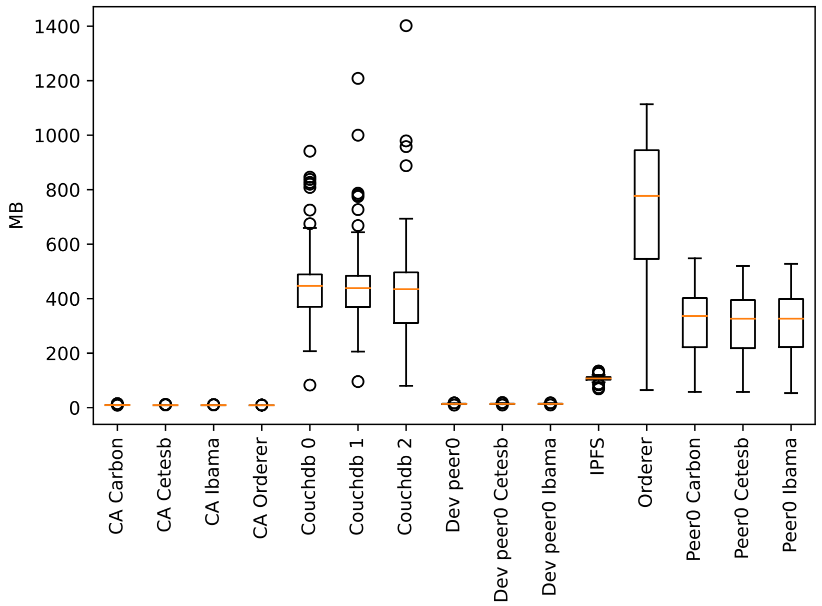
Figure 23 – CPU Usage in the local environment in 100 TPS.



Source: The author.

Now, in Figure 24, when analyzing the consumption in MB of memory, containers with smaller IQRs and fewer outliers (e.g., all the CA) indicate more consistent memory usage patterns. As Box Plot 2008, containers with larger IQRs and numerous outliers (e.g., CouchDB, Orderer, Peers) show higher variability, which could imply less predictable memory usage and potential for performance issues. All the CA and Dev Peers containers have relatively lower and more consistent memory usage, with less variation and fewer outliers than others. However, the CouchDB containers use more memory and show more variability, indicating sporadic spikes in memory usage. The IPFS container has higher memory usage with significant variability, as indicated by the larger IQR and the presence of outliers. The Orderer and Peers container stands out with an extensive range of memory usage, with no significant spikes.

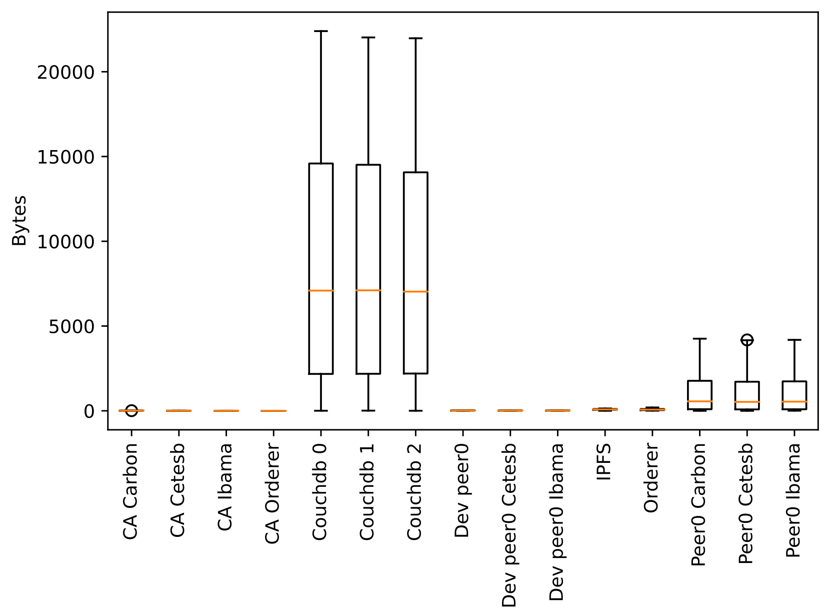
Figure 24 – Memory Usage in the local environment in 100 TPS.



Source: The author.

When analyzing the storage read data, Figure 25 shows the CouchDB containers display significant spikes in storage read activity, indicating periods of heavy and variable usage. These spikes suggest that CouchDB containers are involved in intensive data processing tasks. The Peer Containers show notable spikes, though not as pronounced as those of the CouchDB containers. This indicates substantial but less variable storage read activity compared to CouchDB. CA, IPFS, and Orderer containers maintain relatively stable and low storage read metrics throughout the observation period, indicating they are not heavily involved in storage-intensive operations. High Variation in CouchDB and Peer Containers are likely engaged in intensive data processing tasks, leading to high and variable storage read activity. The high standard deviation indicates inconsistent storage usage patterns, which may indicate workload spikes or inefficient storage access patterns. On the other hand, stability can be advantageous for tasks requiring consistent performance.

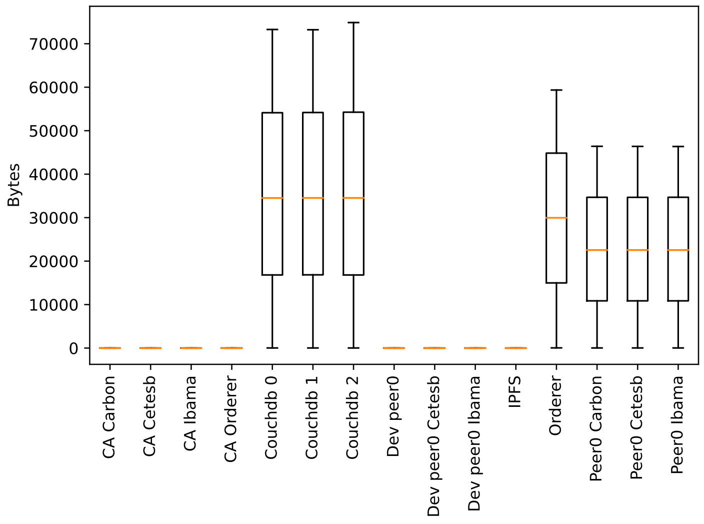
Figure 25 – Storage IO read in the local environment in 100 TPS.



Source: The author.

Figure 26 we collected and analyzed storage write metrics. The CouchDB containers exhibit the highest storage write activities among all the containers analyzed. The storage write operations for these containers consistently increase over time, indicating high data transactions and storage operations. This trend confirms that the CouchDB instances are heavily utilized for storing and retrieving blockchain data. The Orderer container also shows a significant increase in storage write operations over time, though at a slightly lower rate than the CouchDB containers. This indicates that the Orderer node is actively involved in processing and storing transaction data. Peer nodes indicate a steady increase in storage write operations, these nodes are crucial for maintaining the blockchain ledger and processing transactions, which is reflected in their storage write patterns. Containers associated with CA, IPFS and Dev Peers show minimal storage write activity, their storage write metrics remain almost constant throughout the observation period.

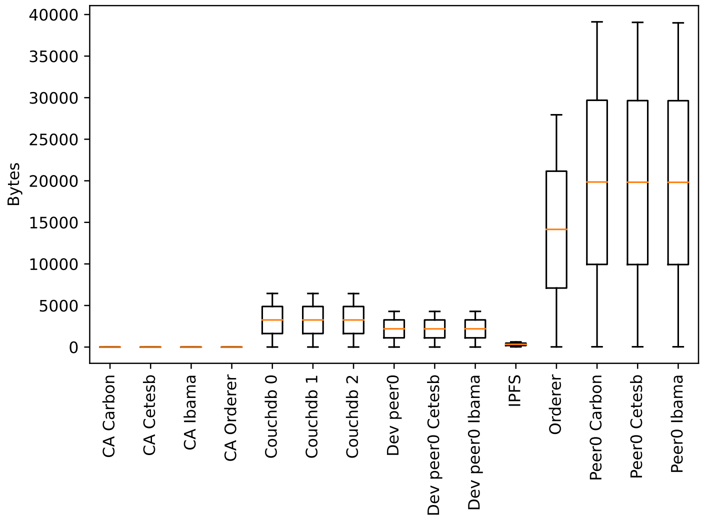
Figure 26 – Storage IO write in the local environment in 100 TPS.



Source: The author.

Figure 27 shows the Network IO received metrics provide a detailed look into the network activity of various Docker containers over time. The Peers and Orderer containers exhibit the highest activity among all containers, reflecting significant variability in their network read operations. The CA and IPFS show very low and consistent network read activity. The CouchDB and Dev Peers containers show moderate to high activity, but not compared with Orderer and Peers.

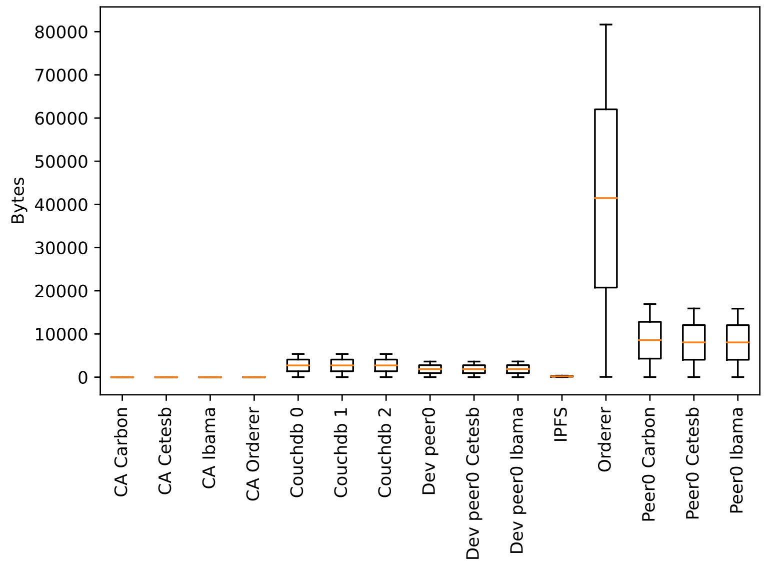
Figure 27 – Network IO received in the local environment in 100 TPS.



Source: The author.

Figure 28 reveals the Orderer container stands out with the highest network IO values and significant variability, including extreme outliers. The presence of outliers suggests that there are occasional spikes in activity, possibly due to certain operations or workload bursts. The Peers containers show a higher level of network IO compared to the CA containers, with moderate variability and some outliers, indicating they handle more substantial and varying data transmissions. The CA and IPFS containers have low and consistent network IO values with minimal variability, suggesting they perform stable, low-volume network operations. The CouchDB, Dev Peers containers show moderate to high network IO, with a notable spread indicating variability in data transmission.

Figure 28 – Network IO transmitted in the local environment in 100 TPS.



Source: The author.

After being able to observe all of this information, we can conclude that in this experiment:

- Network IO - The Orderer and Peer containers handle significantly more network traffic, as indicated by higher median values for both bytes received and transmitted. These containers are critical in managing network communications within the system. The higher variability and frequent outliers observed in these containers suggest periodic network traffic surges.

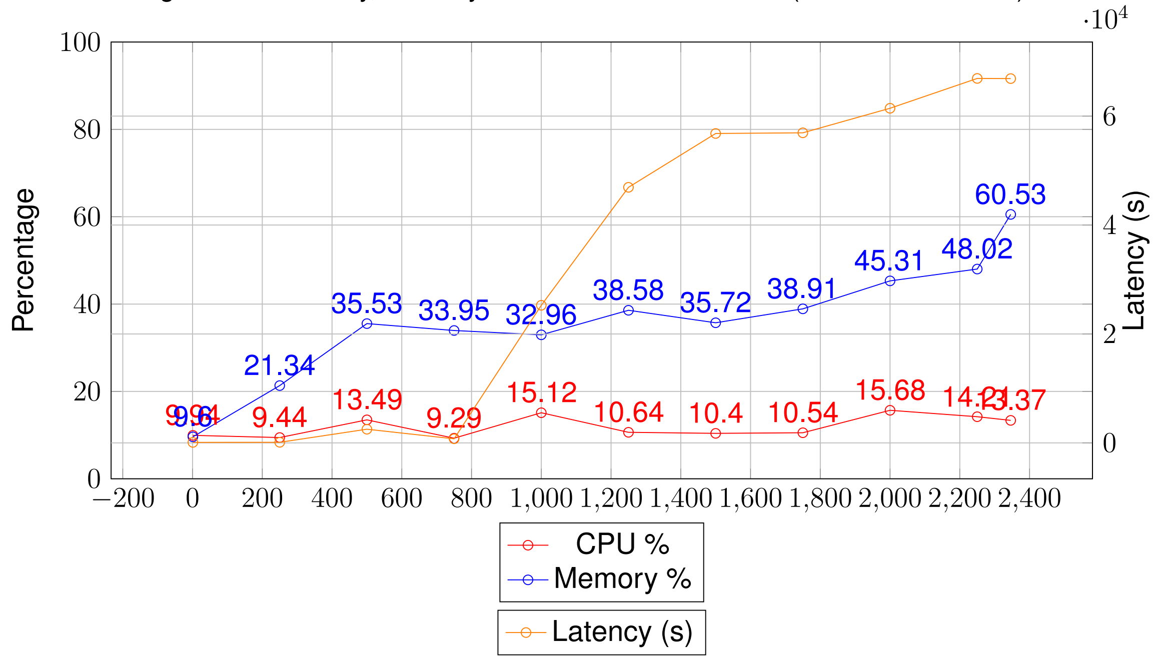
- Storage IO - The CouchDB, Orderer, and Peer containers consistently show higher median values and more significant variability in both bytes received and written. This indicates that these containers are heavily involved in storage IO operations.

- CPU and Memory - The CouchDB, Orderer, and Peers containers significantly handle substantial CPU and memory usage. The observed outliers and variability in CouchDB, Peers, and Orderer metrics suggest potential performance bottlenecks that warrant further investigation and optimization efforts.

4.2 RESULTS GENERATING 50 TPS ON AZURE

The 50 TPS round execution ran for 2347 minutes (39 hours and 7 minutes) until the blockchain service crashed, making it no longer possible to process new transactions successfully. The experiment results are illustrated in Figure 29, which depicts the percentage of memory and CPU usage over time, along with the transaction latency.

Figure 29 – Latency, Memory % and CPU % over Time (Cloud and 50 TPS).



Source: The author.

Table 5 allows us to identify the transaction totals over time in addition to the latency, memory, and CPU results. In this experiment, different from the 50 TPS in the local environment, described in the Subsection 4.1.1, there isn't a high CPU and memory usage over time.

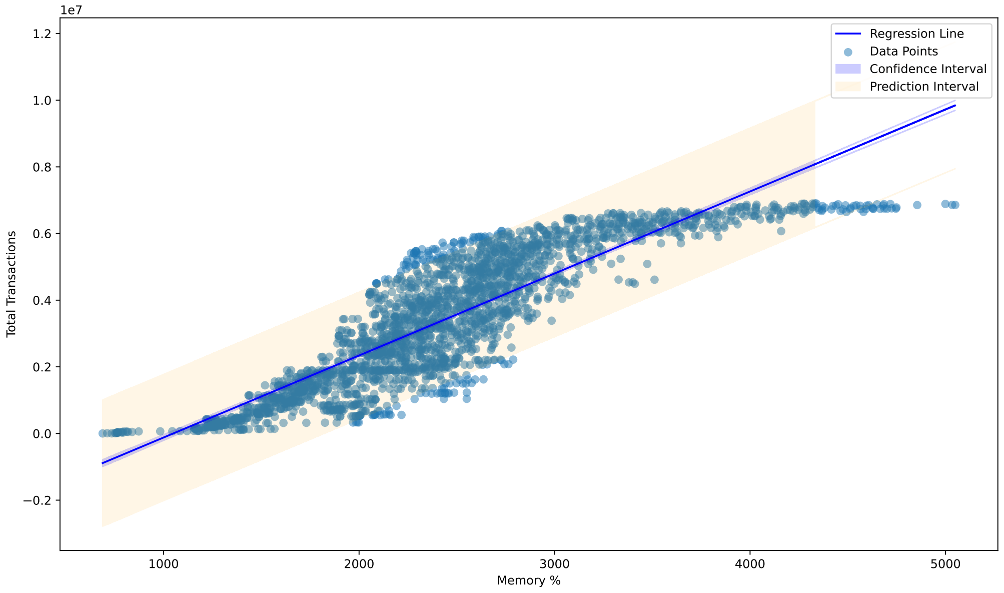
Table 5 – Latency Memory, CPU and Transactions by thousand (Cloud and 50 TPS).

Minute	250	500	750	1000	1250	1500	1750	2000	2250	2347
Latency (ms)	107	2513	750	5821	4692	5681	5693	6143	6691	6668
Memory %	21.34	35.53	33.95	32.96	38.58	35.72	38.91	45.31	48.02	60.42
CPU %	9.44	13.49	9.29	15.12	10.64	10.40	10.54	15.68	14.21	10.47
Total Failed Transactions	0.00	1.63	120.86	160.59	160.59	160.59	161.93	162.05	162.15	162.40
Total Successful Transactions	748.91	1497.08	2127.78	2865.46	3615.16	4364.81	5114.46	5864.11	6613.81	6902.91
Total Unfinished Transactions	18.99	56.16	213.17	350.98	367.24	411.51	649.95	1404.20	2157.95	2450.40
Total Transactions	748.91	1498.73	2248.66	3026.11	3775.77	4525.55	5279.46	6029.18	6778.98	7068.33

Source: The author.

The Figure 30 shows the regression model with $r = 0.873$ and $r^2 = 0.763$. It means that approximately 76.3% of the variance in total transactions can be explained by Memory %. Like the experiment in a local environment, the result indicates a strong relationship between the memory percentage and total transactions. The total number of successful transactions reached approximately 6,902,910 before the system crashed.

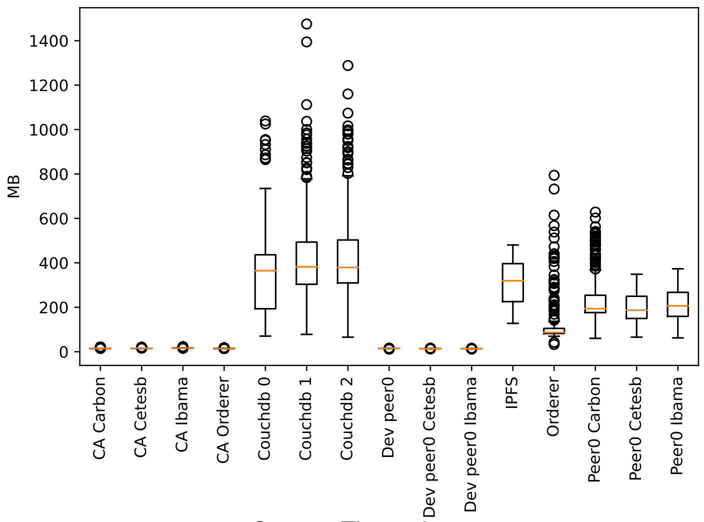
Figure 30 – Regression model with the total transactions and Memory % (Cloud and 50 TPS).



Source: The author.

Figure 31 is the box plot that provides a visual summary of the distribution of memory usage for all containers. It is possible to see that all the CA and Dev Peer have relatively low memory usage with small IQRs, indicating consistent usage with few outliers. However, all CouchDB show higher memory usage with larger IQRs. There are noticeable outliers, indicating some instances where memory usage is significantly higher. The IPFS and Peers have moderate to high memory usage with some variability. Finally, the Orderer uses moderate memory with high variability; the presence of outliers indicates occasional spikes in memory usage.

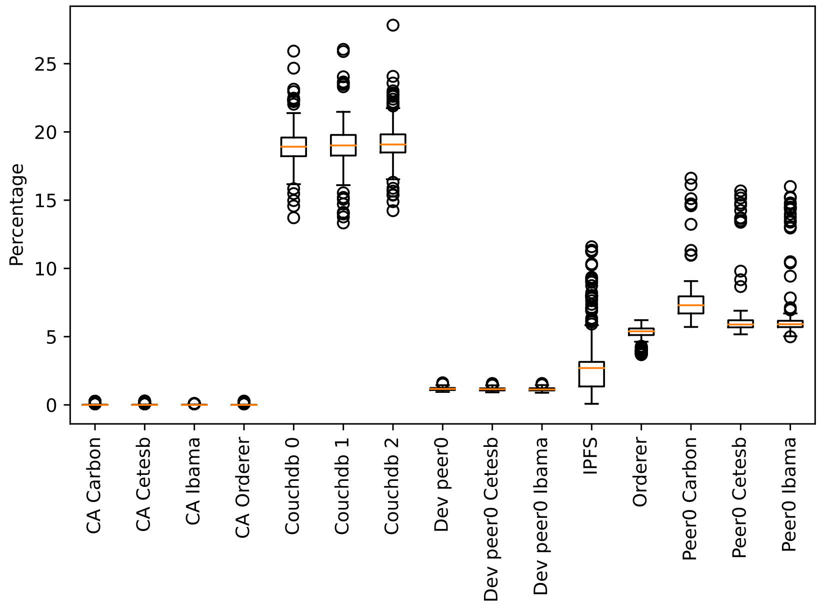
Figure 31 – Memory Usage in the cloud environment in 50 TPS.



Source: The author.

When analyzing CPU consumption, Figure 32 shows CouchDB, IPFS, and Peers exhibit higher and more variable CPU usage, with significant outliers indicating occasional spikes. This variability in CPU usage can be attributed to the nature of these components, as CouchDB handles a substantial amount of read and write operations to the database, IPFS manages decentralized storage and retrieval of files, and Peers are responsible for executing smart contracts and validating transactions. These operations can cause intermittent surges in CPU demand, reflecting the observed outliers. In contrast, all CA and Dev Peers display relatively lower and more consistent CPU usage. The CA primarily handles certificate issuance and management, which are less computationally intensive tasks compared to the continuous transaction processing handled by Peers. Similarly, Dev Peers are likely involved in development and testing activities, which do not require the same level of resource intensity as production Peers. Therefore, their CPU consumption remains more stable and predictable.

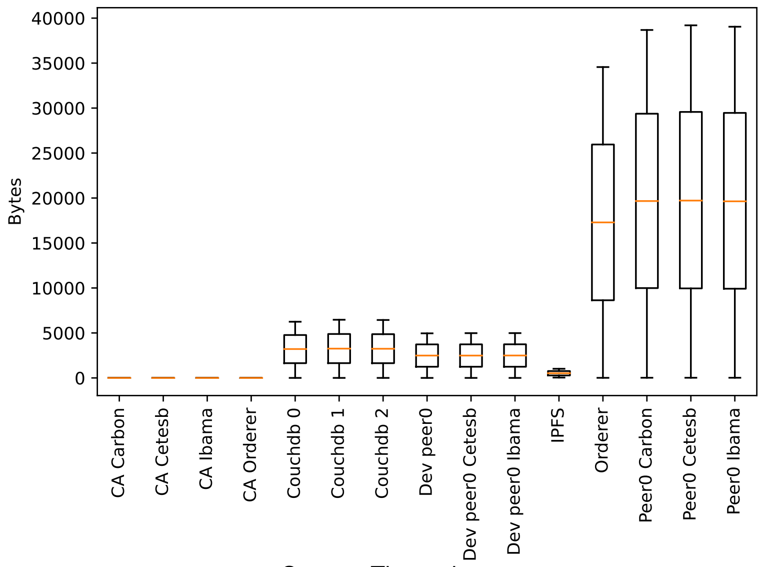
Figure 32 – CPU Usage in the cloud environment in 50 TPS.



Source: The author.

In Figure 33 shows the Network IO received metrics, it is possible to see the Orderer and Peers instances handle significantly more network traffic than the others, suggesting their critical role in the system. The higher variability and outliers indicate these containers might experience periodic surges in network activity, which could affect overall system performance. The CouchDB and Dev Peers containers show moderate to high activity, but not compared with Orderer and Peers. The CA and IPFS containers maintain relatively stable and low data received.

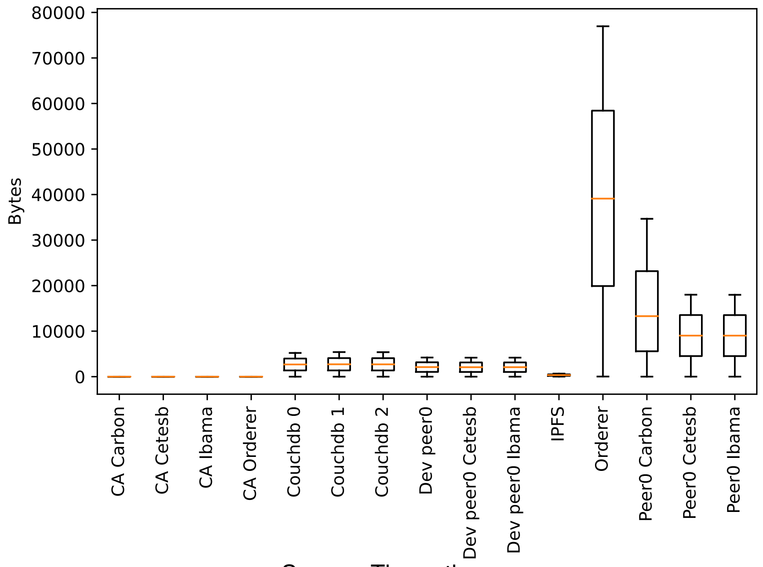
Figure 33 – Network IO received in the cloud environment in 50 TPS.



Source: The author.

In Figure 34, the network IO transmitted indicates the Orderer container stands out with the highest network IO values and significant variability, including extreme outliers. The presence of outliers suggests that there are occasional spikes in activity, possibly due to certain operations or workload bursts. The Peers containers show a higher level of network IO than the CA containers, with moderate variability and some outliers, indicating they handle more substantial and varying data transmissions. The CA and IPFS containers have low and consistent network IO values with minimal variability, suggesting they perform stable, low-volume network operations. The CouchDB, Dev Peers containers show moderate to high network IO, with a notable spread indicating variability in data transmission.

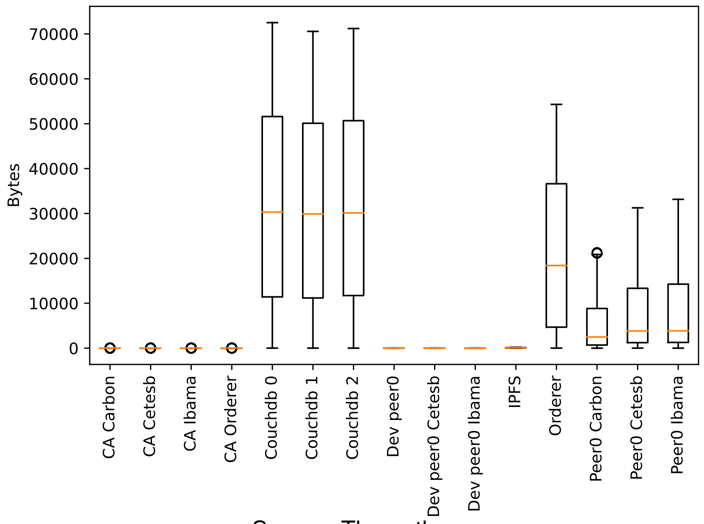
Figure 34 – Network IO transmitted in the cloud environment in 50 TPS.



Source: The author.

When analyzing the storage IO metrics, explicitly focusing on the bytes received over time, illustrated by Figure 35, CouchDB instances handle significantly more block IO, suggesting their critical role for storage needs. The higher variability and outliers indicate these containers might experience periodic surges in block IO, which could affect overall system performance, like the increase of latency. The other instances, like the CA and Dev Peers, display relatively lower and consistent block IO with minimal variability and no significant outliers.

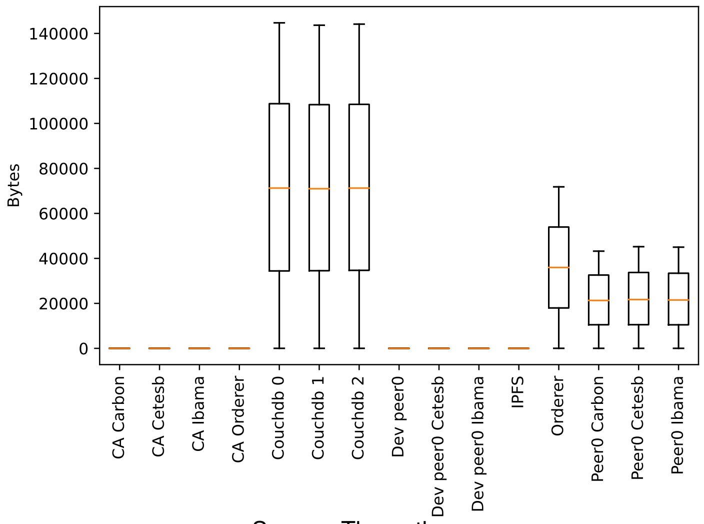
Figure 35 – Storage IO read in the cloud environment in 50 TPS.



Source: The author.

Finally, Figure 36 presents the write storage IO data. The result is very similar to Figure 35, with the CouchDB and Orderer showing higher median values compared to others, indicating they consistently write more block IO bytes. Furthermore, all the CA display relatively lower and consistent block IO writes with minimal variability and no significant outliers.

Figure 36 – Storage IO write in the cloud environment in 50 TPS.



Source: The author.

After observe all of this information, some considerations:

- Network IO - The Orderer and Peer containers handle significantly more network traffic, as indicated by higher median values for both bytes received and transmitted. These containers are critical in managing network communications within the system. The higher variability and frequent outliers observed in these containers suggest periodic network traffic surges.
- Storage IO - The CouchDB, Orderer, and Peer containers consistently show higher median values and more significant variability in both bytes received and written. This indicates that these containers are heavily involved in storage IO operations.
- CPU and Memory - The CouchDB containers not only handle substantial storage IO but are also likely to significantly impact CPU and memory usage. The observed outliers and variability in CouchDB metrics suggest potential performance bottlenecks that warrant further investigation and optimization efforts.

4.3 RESULTS GENERATING 100 TPS ON AZURE

The 100 TPS round execution ran for 3619 minutes (60 hours and 19 minutes) until the blockchain service crashed due to lack of disk space to store the data, making it no longer possible to process new transactions successfully. Figure 37 presents how stable memory and CPU are when executing 100 TPS, indicating that the environment configuration can handle the number of requests without harming the system's health.

Figure 37 – Memory % and CPU % over Time generating 100 TPS (Cloud).

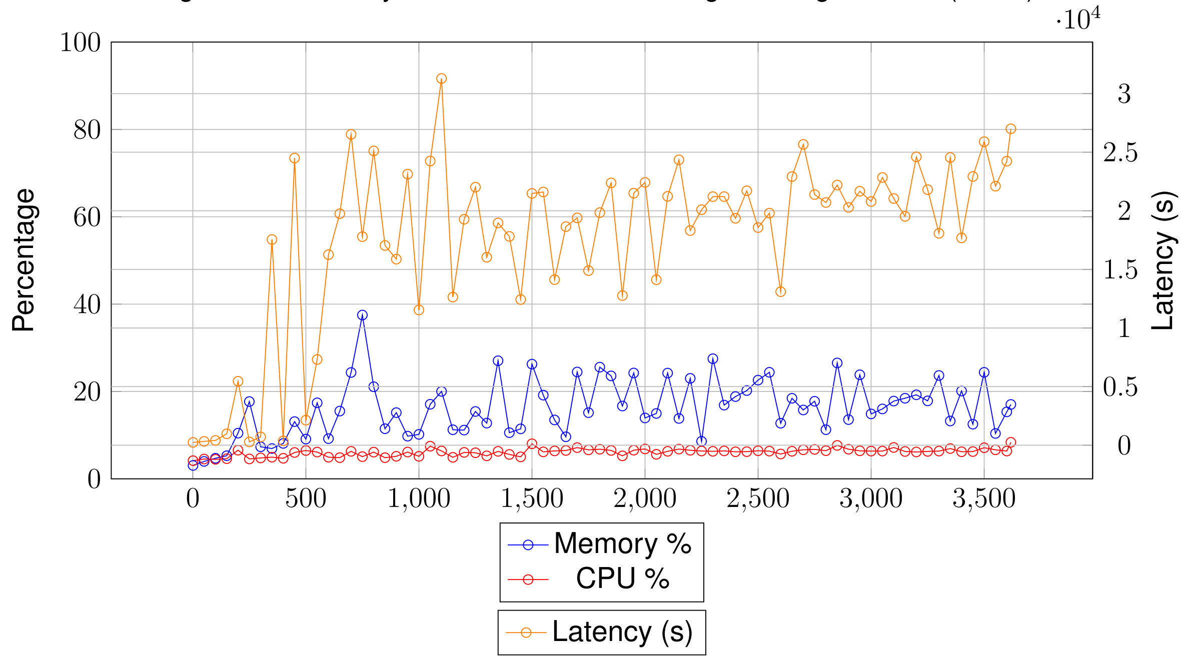


Table 6 exposes the transaction totals over time and the latency, memory, and CPU results. In this experiment, similar to the 50 TPS in the cloud, described in Section 4.2, there isn't a high CPU and memory usage over time. The total number of successful transactions reached approximately 21,023,340 before the system crashed.

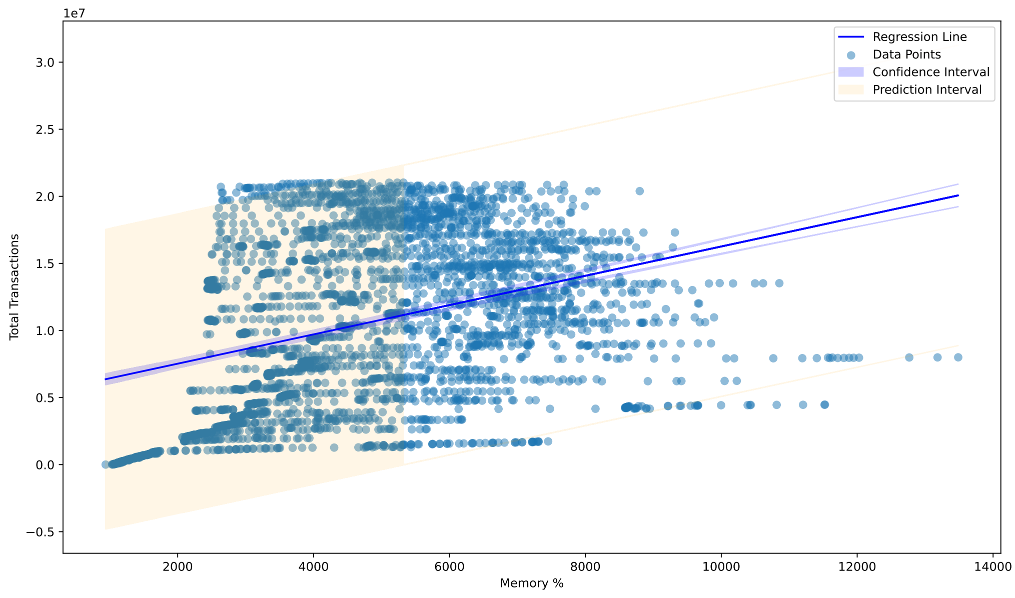
Table 6 – Memory, CPU, and Transactions by thousand in 100 TPS (Cloud).

Minute	250	500	750	1000	1250	1500	1750	2000	2250	2500	2750	3000	3250	3500	3619
Latency (ms)	278	2135	17794	11538	22017	21482	14896	22422	20094	18568	21385	20794	21805	25896	28525
Memory %	17.69	9.10	37.50	10.18	15.42	26.26	15.15	13.93	8.60	22.60	17.76	14.86	17.85	24.38	17.32
CPU %	4.55	6.48	5.08	5.15	5.97	8.02	6.63	6.84	6.37	6.43	6.74	6.35	6.28	7.12	5.86
Total Failed Transactions	0.00	0.00	13.88	13.88	13.88	13.88	13.88	13.88	13.88	13.88	13.88	13.88	13.88	13.88	27.62
Total Successful Transactions	1498.22	2998.01	4465.63	5945.12	7419.56	8887.49	10371.15	11820.57	13281.51	14712.31	16123.16	17554.19	18966.24	20365.75	21023.34
Total Unfinished Transactions	56.83	312.71	740.18	1209.38	1644.94	2091.07	2539.90	3008.15	3473.99	3960.01	4469.03	4946.58	5448.80	5966.83	6221.36
Total Transactions	1498.25	2998.23	4481.36	5960.29	7435.45	8903.39	10386.57	11836.54	13297.37	14727.98	16139.03	17570.03	18982.15	20381.88	21053.63

Source: The author.

Figure 38 shows the regression model with a correlation coefficient of $r = 0.3508$ and a coefficient of determination of $r^2 = 0.123$. This means that the percentage of memory usage can explain approximately 12.3% of the variance in total transactions. In this experiment, the number of transactions did not significantly impact memory consumption, indicating no strong relationship between the transaction rate of 100 TPS and memory usage in this scenario. This suggests that the system could handle the incoming requests efficiently without compromising the stability of the environment or causing any system failures. The system's resilience under this load implies that the memory resources were adequately provisioned to manage the given transaction rate.

Figure 38 – Regression model with the total transactions and memory (Cloud and 100 TPS).

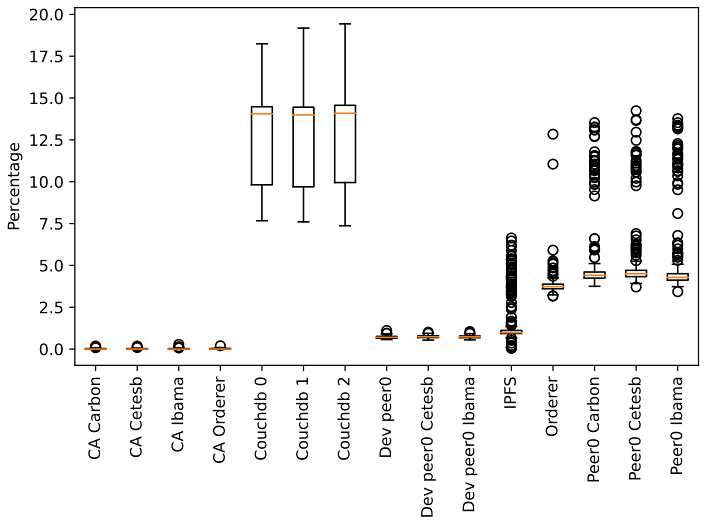


Source: The author.

It is essential to note that while the current transaction rate of 100 TPS did not significantly impact, increasing the number of requests substantially (doubling or more) could potentially create a positive relationship between transaction rate and memory consumption. Such an increase might strain the system's resources, leading to higher memory usage and possibly affecting overall performance and stability. Therefore, careful monitoring and scaling strategies would be essential to maintain system performance as the transaction load increases.

Figure 39 is the box plot that provides a visual summary of the distribution of CPU usage for all containers. Most containers have stable CPU usage with few variations, except for the CouchDB and Peer containers, which show higher usage and some variability. The Peer containers, responsible for validating transactions, executing smart contracts, and maintaining the blockchain state, also show higher CPU usage and variability. The most significant outlier indicates occasional spikes in CPU usage. All CA and Dev peers containers have shallow CPU usage with minimal variation and no significant outliers. The IPFS has relatively moderate CPU usage, noticeable variation, and outliers.

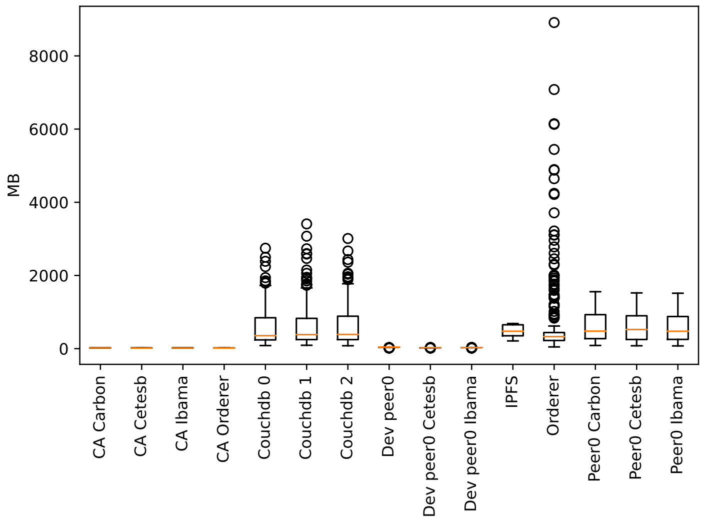
Figure 39 – CPU Usage in the cloud environment in 100 TPS.



Source: The author.

Now, in Figure 40, when analyzing the consumption in MB of memory, containers with smaller IQRs and fewer outliers (e.g., all the CA and Dev Peers) indicate more consistent memory usage patterns. All the CA and Dev Peers containers have relatively lower and more consistent memory usage, with less variation and fewer outliers than others. However, the CouchDB and Peers containers use more memory and show more variability, indicating sporadic spikes in memory usage. The Orderer container has higher memory usage with significant variability, as indicated by the larger IQR and the presence of outliers.

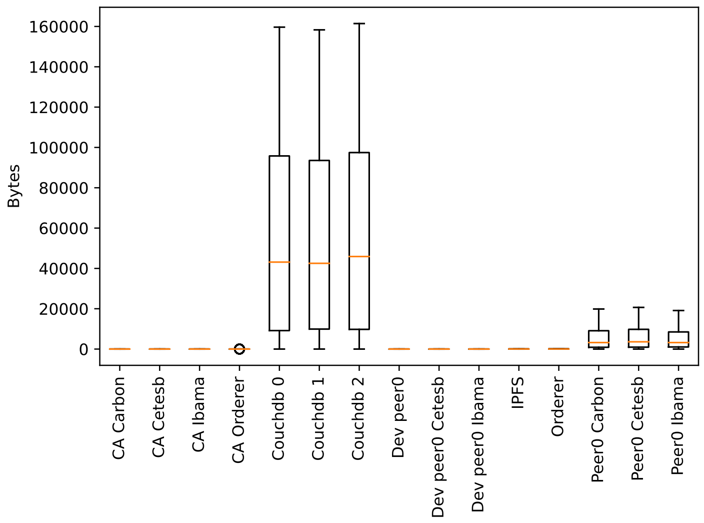
Figure 40 – Memory Usage in the cloud environment in 100 TPS.



Source: The author.

When analyzing the storage read data, Figure 41 reveals the CouchDB containers display significant spikes in storage read activity, indicating periods of heavy and variable usage. These spikes suggest that CouchDB containers are involved in intensive data processing tasks. The Peer Containers show notable spikes, though not as pronounced as those of the CouchDB containers. This indicates substantial but less variable storage read activity compared to CouchDB. CA, IPFS, and Orderer containers maintain relatively stable and low storage read metrics throughout the observation period, indicating they are not heavily involved in storage-intensive operations. High Variation in CouchDB and Peer Containers likely results from intensive data processing tasks, leading to high and variable storage read activity. The high standard deviation indicates inconsistent storage usage patterns, which may indicate workload spikes or inefficient storage access patterns. On the other hand, stability can be advantageous for tasks requiring consistent performance.

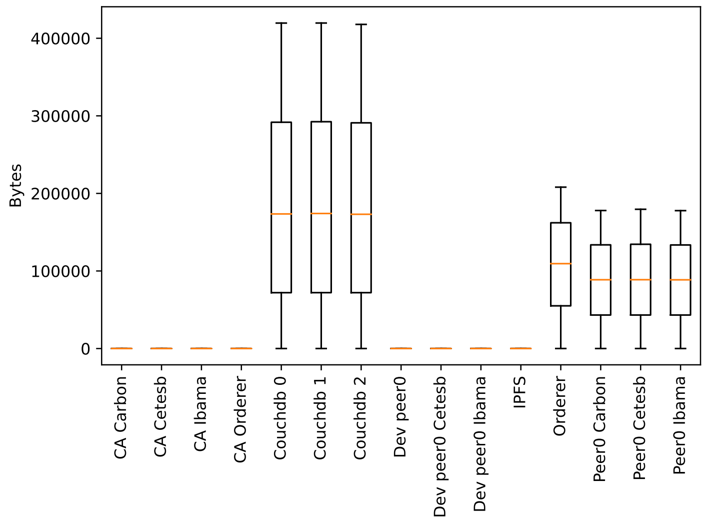
Figure 41 – Storage IO Read in the cloud environment in 100 TPS.



Source: The author.

Figure 42 exposes collected and analyzed storage write metrics. The CouchDB containers exhibit the highest storage write activities among all the containers analyzed. The storage write operations for these containers consistently increase over time, indicating high data transactions and storage operations. This trend confirms that the CouchDB instances are heavily utilized for storing and retrieving blockchain data. The Orderer container also shows a significant increase in storage write operations over time, though at a slightly lower rate than the CouchDB containers. This indicates that the Orderer node is actively involved in processing and storing transaction data. Peer nodes present a steady increase in storage write operations, these nodes are crucial for maintaining the blockchain ledger and processing transactions, which is reflected in their storage write patterns. Containers associated with CA, IPFS and Dev Peers show minimal storage write activity, their storage write metrics remain almost constant throughout the observation period.

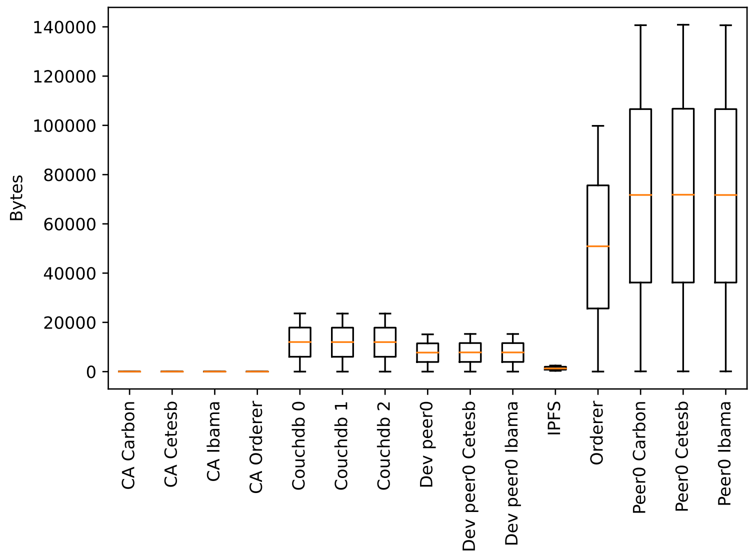
Figure 42 – Storage IO write in the cloud environment in 100 TPS.



Source: The author.

Figure 43 shows the Network IO received metrics provide a detailed look into the network activity of various Docker containers over time. The Peers and Orderer containers exhibit the highest activity among all containers, reflecting significant variability in their network read operations. The CA and IPFS show very low and consistent network read activity. The CouchDB and Dev Peers containers show moderate to high activity, but not compared with Orderer and Peers.

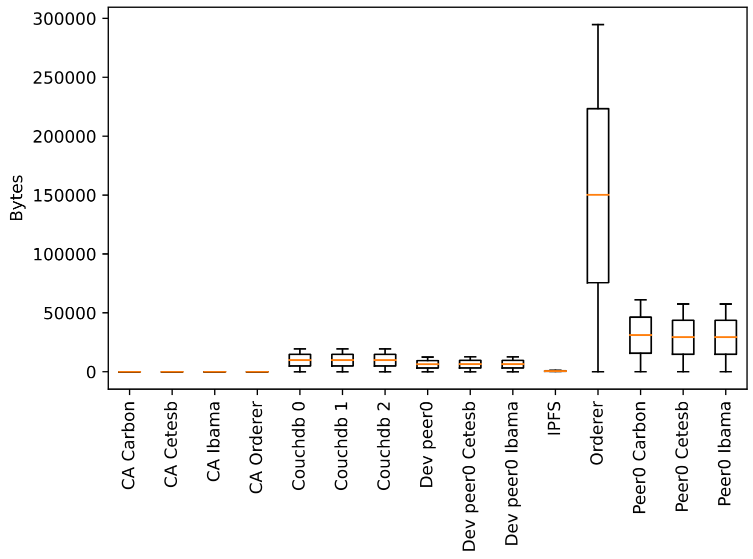
Figure 43 – Network IO received in the cloud environment in 100 TPS.



Source: The author.

Figure 44 reveals the Orderer container stands out with the highest network IO values and significant variability, including extreme outliers. The presence of outliers suggests that there are occasional spikes in activity, possibly due to certain operations or workload bursts. The Peers containers show a higher level of network IO than the CA containers, with moderate variability and some outliers, indicating they handle more substantial and varying data transmissions. The CA and IPFS containers have low and consistent network IO values with minimal variability, suggesting they perform stable, low-volume network operations. The CouchDB, Dev Peers containers show moderate to high network IO, with a notable spread indicating variability in data transmission.

Figure 44 – Network IO transmitted in the cloud environment in 100 TPS.



Source: The author.

After being able to observe all of this information, we can conclude that in this experiment:

- Network IO - The Orderer and Peer containers handle significantly more network traffic, as indicated by higher median values for both bytes received and transmitted. These containers are critical in managing network communications within the system. The higher variability and frequent outliers observed in these containers suggest periodic network traffic surges.
- Storage IO - The CouchDB, Orderer, and Peers containers consistently show higher median values and more significant variability in both bytes received and written, indicating these containers are heavily involved in storage IO operations.
- CPU and Memory - The CouchDB, Orderer, and Peers containers significantly handle substantial CPU and memory usage. The observed outliers and variability in CouchDB, Peers, and Orderer metrics suggest potential performance bottlenecks that warrant further investigation and optimization efforts.

4.4 RESULT ANALYSIS

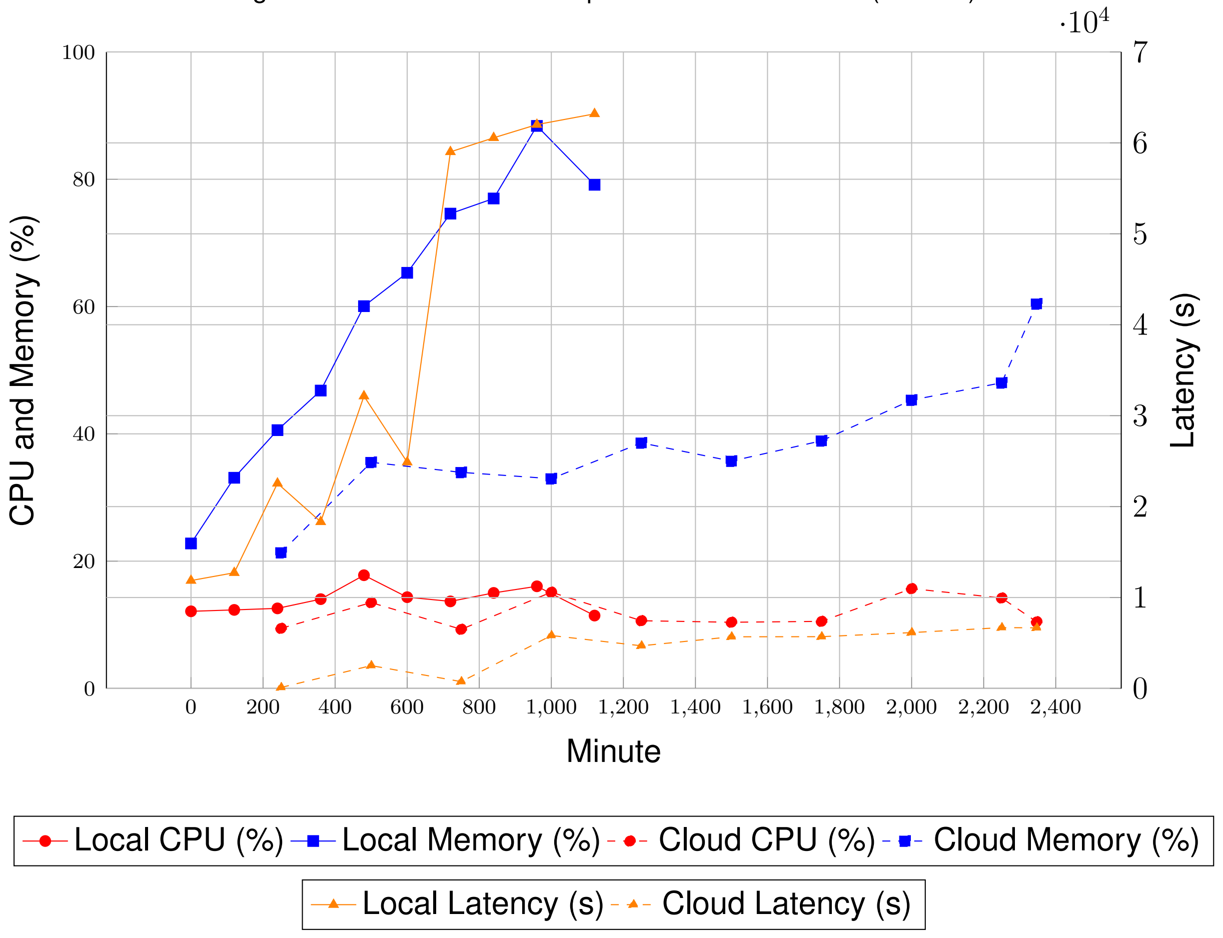
The methods employed in this project, including customizing Hyperledger Caliper, detailed data extraction and analysis, and the dual-environment testing approach (local and cloud), collectively contributed to a thorough and robust analysis of the system's performance. These strategies ensured that the insights gained were accurate and applicable to real-world blockchain deployments, enhancing the Carbono 21 project's reliability and efficiency.

4.4.1 Analysis of 50 TPS results between local and cloud

In comparing the execution results of 50 TPS between a local and a cloud environment (Figure 45), several differences and insights emerge. It is essential to highlight that the local environment had 16Gb RAM, and the cloud had 8Gb RAM. For the local environment, latency starts at 11866 ms and increases significantly over time, peaking at 60559 ms before ending at 63192 ms. This considerable increase indicates substantial delays and instability in processing transactions. Memory usage begins at 22.78%, steadily increases to a peak of 88.38%, and then slightly decreases to 79.13%, suggesting potential memory leaks or insufficient memory management. CPU usage remains relatively stable, starting at 12.12%, peaking at 17.79%, and ending at 11.45%. The total number of failed transactions is initially zero but increases sharply, peaking at 305,793. Similarly, unfinished transactions start at 2,990 and rise significantly to

483,742. Successful transactions increase steadily from 358,818 to 3,250,781, with the total number of transactions increasing from 358,823 to 3,559,710.

Figure 45 – Performance Comparison: Local vs Cloud (50 TPS)



Source: The author.

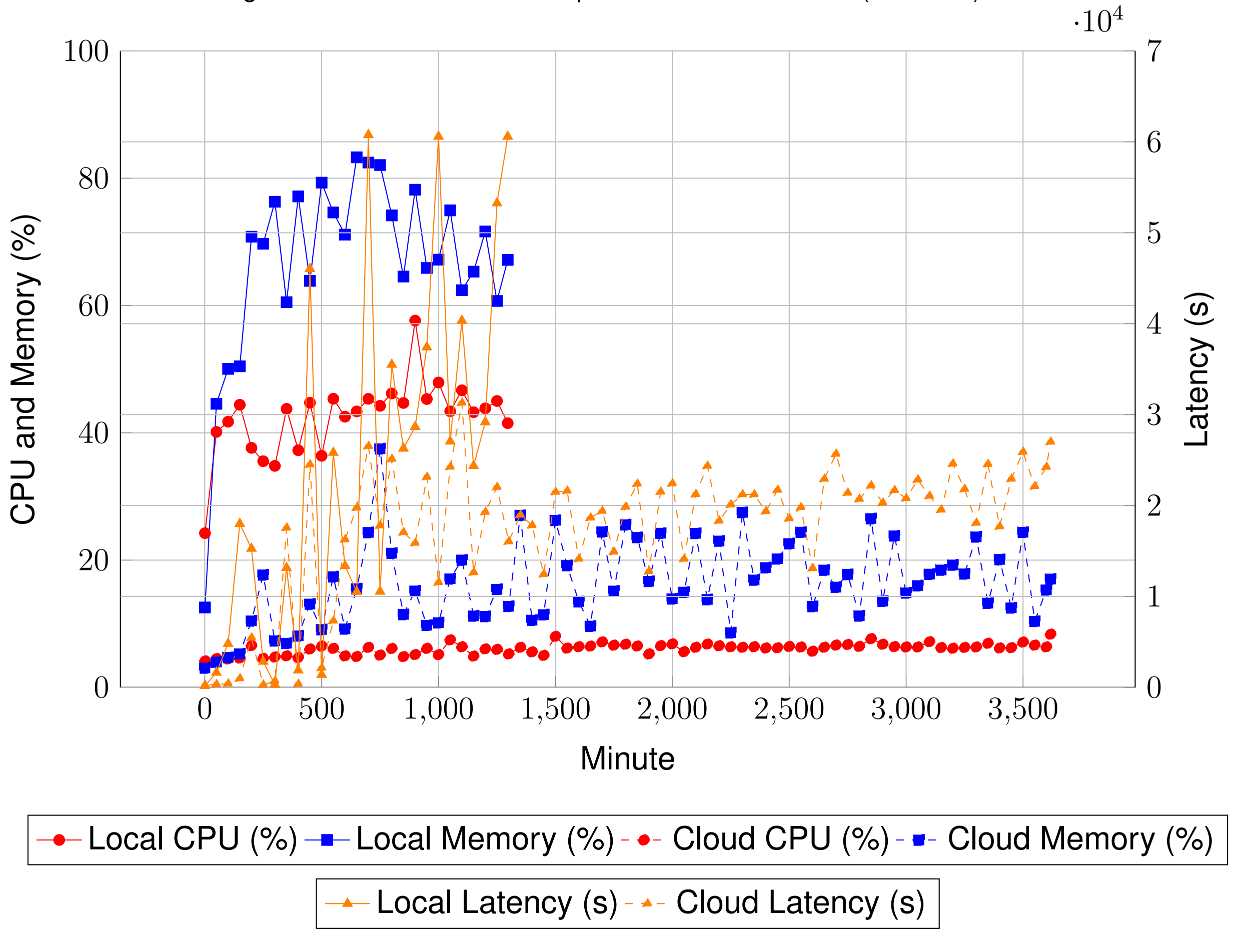
In contrast, the cloud environment shows more stable performance. Latency starts low at 107 ms and fluctuates but never reaches the extreme values seen in the local environment, peaking at 6691 ms. Memory usage starts at 21.34% and generally increases, peaking at 60.42%, showing more stability compared to the local environment. CPU usage starts at 9.44% and fluctuates, peaking at 15.68%, indicating efficient CPU utilization. The total number of failed transactions starts at none and increases steadily to 162,400, a lower rate than the local environment. Successful transactions increase steadily from 748,910 to 6,902,910, with unfinished transactions starting at 18,990 and rising significantly to 2,450,400. The total number of transactions increases from 748,910 to 7,068,330.

Overall, the cloud environment seems more stable performance with lower latency, efficient memory and CPU usage, fewer failed transactions, and more successful transactions than the local environment. The local environment struggles with high latency spikes, increasing memory usage, and more failed transactions, suggesting it may not be as suitable for high TPS execution without further optimization. These observations indicate that the cloud environment is better suited for maintaining performance under the given workload, highlighting the advantages of cloud resources in handling higher transaction processing scenarios.

4.4.2 Analysis of 100 TPS results between local and cloud

In comparing the execution results of 100 TPS between a local and a cloud environment (Figure 46) several differences and insights emerge. For the local environment, latency starts at 104 ms and increases significantly over time, peaking at 61,492 ms before ending at 40,195 ms. This considerable increase indicates substantial delays and instability in processing transactions. Memory usage begins at 18.86%, steadily increases to a peak of 96.48%, and then slightly decreases to 79.21%, suggesting potential memory leaks or insufficient memory management. CPU usage remains relatively stable, starting at 9.87%, peaking at 23.65%, and ending at 19.02%. The total number of failed transactions is initially zero but increases sharply, peaking at 381,160. Similarly, unfinished transactions start at 67,360 and rise significantly to 1,659,850. Successful transactions started with 718,320 and finished with 5,420,970, and the total number of transactions was 5,806,960.

Figure 46 – Performance Comparison: Local vs Cloud (100 TPS).



Source: The author.

In contrast, the cloud environment shows more stable performance. Latency starts low at 278 ms and fluctuates but never reaches the extreme values seen in the local environment, peaking at 28525 ms. Memory usage starts at 17.69% and generally increases, peaking at 37.5%, showing more stability compared to the local environment. CPU usage starts at 4.55% and fluctuates, peaking at 8.02%, indicating efficient CPU utilization. The total number of failed transactions starts at none and increases steadily to 27,620, a lower rate than the local environment. Successful transactions started with 1,498,220 and finished with 21,023,340, and the total number of transactions was 21,053,630.

Overall, the cloud environment proofs to have more stable performance with lower latency, efficient memory and CPU usage, fewer failed transactions, and more successful transactions than the local environment. The local environment struggles with high latency spikes, increasing memory usage, and more failed transactions, suggesting it may not be as suitable for high TPS execution without further optimization. These observations indicate that the cloud environment is better suited for maintaining performance under the given workload, highlighting the advantages of cloud resources in handling higher transaction processing scenarios.

4.4.3 Distribution of resources required by each container

The analysis of resource consumption by containers, detailed in Table 7, is a significant contribution of this project since it enables blockchain administrators to allocate VM resources more effectively by container, optimizing VM configurations based on actual needs. This targeted resource allocation helps reduce costs associated with over-provisioned VM configurations and ensures that resources are balanced according to the specific demands of each container.

Table 7 – Resource consumption by container

	Memory	**CPU**	**Network IO transmitted**	**Network IO Received**	**Storage Write**	**Storage Read**
CouchDB	High	Medium with outliers	Low	Low	High	High
Orderer	Medium with high outliers	Low with high outliers	High	High	Medium	Insignificant
Peers	Medium with high outliers	Medium with outliers	Medium	High	Medium	Low
Dev Peers	Insignificant	Insignificant	Low	Low	Insignificant	Insignificant
CA	Insignificant	Insignificant	Insignificant	Insignificant	Insignificant	Insignificant
IPFS	Low with high outliers	Low	Insignificant	Insignificant	Insignificant	Insignificant

Source: The author.

Table 7 provides a clear summary of resource consumption metrics for different containers, and the following points highlight the key findings from the analysis:

- Memory Usage: The Orderer node had higher memory usage outliers than CouchDB, indicating occasional spikes in memory demand. However, the Peer nodes also showed substantial memory consumption without significant outliers, suggesting consistent high memory usage.

- CPU Usage: CouchDB had the highest CPU consumption, with significant usage and outliers, followed by the Peer nodes. IPFS and the Orderer node also displayed high CPU usage with notable outliers, reflecting their computational demands.
- Network IO Transmitted: The Orderer node transmitted the highest number of bytes, followed by the Peer nodes. This indicates that the Orderer is responsible for significant data transmission within the network.
- Network IO Received: Peer nodes received more bytes than the Orderer node. This suggests that the Peer nodes are heavily involved in data reception and processing transactions from the network.
- Storage Write and Read: CouchDB exhibited the highest bytes written, surpassing both the Orderer and Peer nodes. This shows CouchDB's intensive storage I/O operations, which are crucial for maintaining the ledger and state databases.

The insights gained from this detailed resource consumption analysis enable blockchain administrators to allocate and scale strategically VM resources. For example, knowing that CouchDB requires high memory and storage IO resources, administrators can allocate VMs with higher memory and faster storage performance specifically for CouchDB containers. Similarly, understanding the high network IO requirements of Orderer and Peer nodes can lead to provisioning VMs with enhanced network capabilities for these containers.

Moreover, the ability to strategically scale VMs is enhanced by this research. Administrators can implement custom monitoring and auto-scaling systems tailored to the specific needs of each container type. For instance, containers that exhibit high variability in resource consumption, such as CouchDB and Peer nodes, can be equipped with auto-scaling rules to adjust resources dynamically during peak loads, ensuring optimal performance without manual intervention. This approach improves resource efficiency and the overall robustness and resilience of the blockchain network, particularly in scenarios involving high transaction loads or potential DoS attacks.

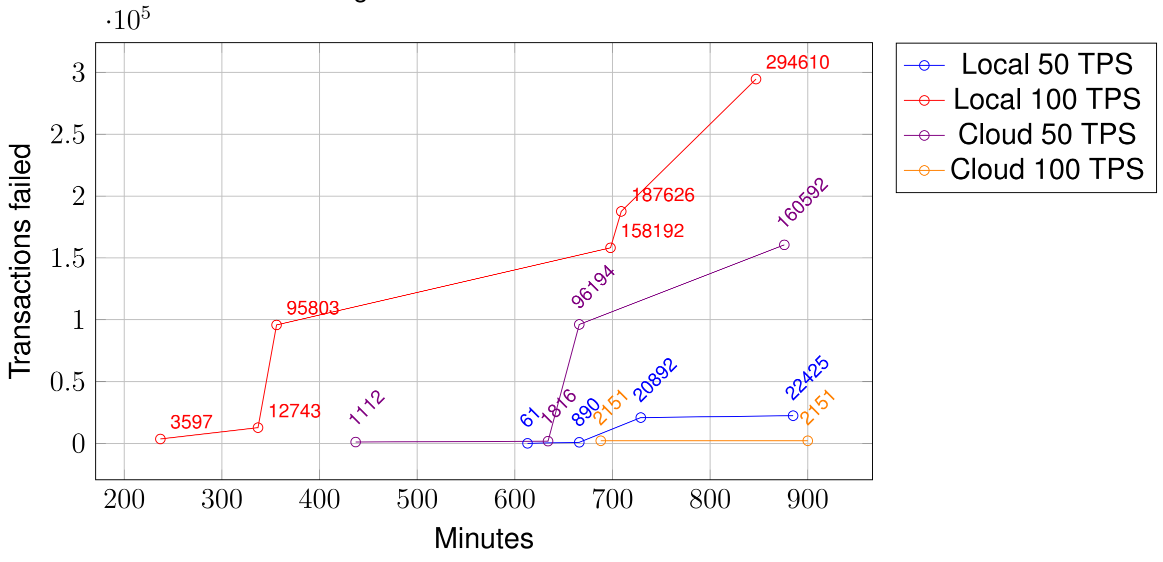
4.4.4 Time-based failure analysis

Analyzing the number of transaction failures during the experiments, as illustrated in Figure 47, reveals key differences in system performance under varying loads over the first 900 minutes (15 hours) of the experiments. At 50 TPS in local environment, the first failure occurred after 613 minutes (10 hours and 13 minutes), resulting in 61 failed transactions, however as the execution progresses the number of failures increases, reaching 22,425 failed transactions after 865 minutes. However, in the same environment at 100 TPS, failures were observed much earlier, after just 237 minutes (3 hours and 57 minutes), with a significant increase to 3,597 failed transactions, as time passes the number of transactions begins to fail in an increasingly larger volume, reaching 294,610 failed transactions after 847 minutes.

In the cloud environment, at 50 TPS, the failure starts at minute 437 (7 hours and 17 minutes) with 1112 failed transactions, 176 minutes (2 hours and 56 minutes) earlier than the local experiment of 50 TPS with a peak of 12095 failed transactions at minute 876 (14 hours and 36 minutes), which indicates that the environment configuration with limited resources caused the failures to start earlier.

Finally, in the cloud environment at 100 TPS, the first transactions began to fail at minute 688 (11 hours and 28 minutes) with 2151 failed transactions, which remained at this value until close to the end of the experiment of more than 60 hours, which indicates that the resources allocated to the VM helped manage the requests so as not to fail so quickly, in a large volume of failed transactions and with subsequent failures.

Figure 47 – Number of failed transactions.



This analysis highlights the critical time window within which the project team must respond before a DoS attack starts to significantly disrupt the system. The increase in transaction volume over time leads to system failures, indicating the point at which the system becomes unable to handle the load, necessitating timely intervention to maintain stability.

4.5 CHAPTER CONSIDERATIONS

In Chapter 4, the results of the experiments listed by Section 3.3 were described. The primary goal was to analyze the performance of the Carbon 21 blockchain network, especially under conditions of DoS attacks. The detailed examination of the resource consumption metrics, including memory, CPU, network IO, and storage IO, provided valuable insights into the system's behavior and potential bottlenecks.

Key observations included the high memory and storage IO requirements of CouchDB, the substantial network IO handled by Orderer and Peer nodes, and the variability in resource consumption across different containers. These findings are crucial for optimizing the allocation and scaling of VM resources in cloud environments. Finally, by understanding the specific needs of each container type, administrators can implement custom monitoring and auto-scaling systems to ensure optimal performance and resilience of the blockchain network.

5 CONSIDERATIONS & FUTURE WORK

Organizations, researchers and developers are constantly adopting the use of NFT technology in blockchain projects, as identified in Section 2.1. The Carbon 21 project, which is under development, has several private and public operations identified as the potential to be used in the performance analysis process of the experiments as per Section 3.1. In this context, there is concern about issues related to the security and performance of the application and its development environment, ensuring that in a possible malicious attack or not, criteria can be identified for analysis with well-defined characteristics of the instances and the environment.

In the development process of this work, there was a need to structure the definition of operations and the development of specific public and private methods, i.e., to develop the strategies to operationalize the Carbon 21 project. The project structure is now defined and developed and can be used to start the tests as identified in Section 3.5. The challenge is to run the experiments and have a good volume of data to perform performance analyses in different scenarios.

Through this research, we aim to understand security vulnerabilities and performance considerations in private blockchain networks, locally and specifically when deployed in cloud environments under DoS attacks. Therefore, for the Carbon 21 project, the result of this work is relevant to validate and understand the operational limits of the platform for certain transactions, as well as the behavior of the solution when in situations of DoS due to legitimate or malicious overload. In this way, by conducting experiments and analyzing the results, we can derive insights and recommendations to enhance the resilience and effectiveness of blockchain solutions in the face of potential DoS attacks.

The customization of Hyperledger Caliper played a pivotal role in the project by allowing the capture of performance metrics with high granularity. The main goal of these customizations was to gather detailed data at a per-second interval, which is crucial for in-depth performance analysis. By recording data every second, the experiments provided a detailed view of the system's performance over time, enabling the identification of fine-grained patterns and anomalies that might be missed with less frequent sampling. After that, the collected data were extracted into CSV files, normalized using Python scripts. The data were used to produce statistical plots, including Box Plots and regression models.

The experiments were conducted in local and cloud environments to validate the results and ensure the results were consistent and reliable, regardless of the deployment setting. Also, local environments are more straightforward to execute and do not incur the costs associated with hosting VMs in a cloud service. This makes local testing an economical option, particularly during the initial development and testing phases. Our analysis exhibited that the local environment can produce results similar to those obtained in the cloud, highlighting the feasibility of using local setups for preliminary testing. This can significantly reduce costs and streamline the development process before scaling to cloud environments for broader deployment and testing. Lastly, while local environments are helpful for initial testing, cloud environments are essential for simulating real-world conditions, such as distributed network performance and scalability. The cloud experiments provided insights into how the system would perform under actual deployment conditions, ensuring that the findings are applicable in practical scenarios.

The regression analysis further confirmed a strong correlation between memory usage and the total number of transactions when the VM is configured with low memory (less than 16GB), different when it has 32GB. This indicates that memory usage alone does not fully explain the variations in transaction success rates, highlighting the need for a more comprehensive analysis of other potential factors influencing performance. The results emphasize the necessity for improved memory management strategies to enhance the stability and performance of blockchain services under high transaction loads. The cloud environment demonstrated superior performance, maintaining stability and efficiency under 50 TPS and 100 TPS loads. In contrast, the local environment struggled with high latency spikes, increased memory usage, and a higher number of failed transactions, making it less suitable for high TPS execution without further optimization. Overall, these observations indicate that the cloud environment is better suited for maintaining performance under the given workload, highlighting the advantages of cloud resources in handling higher transaction processing scenarios.

A significant contribution of this project is the detailed analysis of resource consumption by containers, as presented in 4.4.3. This analysis enables blockchain administrators to allocate VM resources more effectively by container, optimizing VM configurations based on actual needs. For instance, CouchDB, Orderer, and Peer containers were identified as using substantial CPU and memory, indicating their critical roles and potential performance bottlenecks. Targeted resource allocation helps reduce costs associated with over-provisioned VM configurations and ensures that resources are balanced according to the specific demands of each container. Moreover, the ability of blockchain administrators to strategically scale VMs based on the research findings is a crucial advancement. Understanding each container's specific resource requirements, administrators can implement custom monitoring and auto-scaling systems tailored to these needs. This strategic approach allows for dynamic adjustment of resources, ensuring optimal performance and cost-efficiency. For example, containers exhibiting high variability in resource consumption can be set up with auto-scaling rules to handle peak loads without manual intervention, enhancing the blockchain network's overall robustness.

In conclusion, the methods employed, including customizing Hyperledger Caliper, detailed data extraction and analysis, and the dual-environment testing approach, collectively contributed to a thorough and robust analysis of the system's performance. These strategies ensured that the insights gained were accurate and applicable to real-world blockchain deployments, enhancing the Carbon 21 project's reliability and efficiency. Future work will involve further optimization efforts, addressing identified performance bottlenecks, and expanding the scope of experiments to include additional blockchain configurations and environments.

5.1 PUBLICATIONS

During the development of this work, the following conference article was published:

- Battisti, João; **Batista, Vitor**; Marques, Marco; Simplício Jr, Marcos; Koslowski, Guilherme; Pillon, Maurício; Kreutz, Diego and Miers, Charles. Performance analysis of the Raft consensus algorithm on Hyperledger Fabric and Ethereum. 2023. IEEE International Conference on Cloud Computing Technology and Science (CloudCom). DOI: 10.1109/CloudCom59040.2023.00035

5.2 FUTURE WORKS

This work made it possible to understand the memory issue with a high number of Transactions Per Second (TPS), so further investigation into memory management techniques and their impact on blockchain performance could be another contribution. For example, future work can explore garbage collection optimizations and memory allocation strategies to reduce latency and improve transaction throughput. Another round of experiments with higher transaction rates (beyond 100 TPS) can provide insights into the scalability limits of both local and cloud environments. It can help understand system performance's upper bounds and identify potential bottlenecks.

Lastly, analyzing network latency and throughput under different network conditions can be another significant contribution. For example, tests under varying bandwidth and latency scenarios can be created to understand the impact on blockchain performance and transaction processing.

BIBLIOGRAPHY

AL-ROOMI, M. et al. Cloud computing pricing models: a survey. **International Journal of Grid and Distributed Computing**, v. 6, n. 5, p. 93–106, 2013.

ANKENBRAND, T. et al. Proposal for a comprehensive (crypto) asset taxonomy. In: **2020 Crypto Valley Conference on Blockchain Technology (CVCBT)**. [S.l.: s.n.], 2020. p. 16–26.

ANSARI, S.; NASSIF, A. B. A comprehensive study of regression analysis and the existing techniques. In: **2022 Advances in Science and Engineering Technology International Conferences (ASET)**. [S.l.: s.n.], 2022. p. 1–10.

BATTISTI, J. F. et al. Performance analysis of the raft consensus algorithm on hyperledger fabric and ethereum on cloud. In: **2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)**. Los Alamitos, CA, USA: IEEE Computer Society, 2023. p. 155–160. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/CloudCom59040.2023.00035>.

BATTISTI, J. H. F. et al. Analysis of an ethereum private blockchain network hosted by virtual machines against internal dos attacks. In: BAROLLI, L.; HUSSAIN, F.; ENOKIDO, T. (Ed.). **Advanced Information Networking and Applications**. Cham: Springer International Publishing, 2022. p. 479–490. ISBN 978-3-030-99584-3.

BECK R. AGERSKOV, S. **Ethical guidelines for blockchain systems**. Copenhagen, Denmark: European Blockchain Center, 2024. OCLC: 1432701340. ISBN 978-87-7949-074-1. Disponível em: <https://blockchain-observatory.ec.europa.eu/news/ethical-guidelines-blockchain-systems-2024-05-15_en>.

BOX Plot. In: THE Concise Encyclopedia of Statistics. New York, NY: Springer New York, 2008. p. 55–57. ISBN 978-0-387-32833-1. Disponível em: <https://doi.org/10.1007/978-0-387-32833-1_43>.

CARBON, S. **Carbon credit**. 2023. Https://www.sustainablecarbon.com/en/.

CARBONO21. **Carbono 21 project**. 2023.

COMMITTEE, B. S. Ieee standard for blockchain based digital asset management. **IEEE Std 2418.10-2022**, p. 1–19, 2022.

ETHEREUM. **Non-fungible tokens (NFT)**. 2021. Disponível em: <https://ethereum.org/en/nft/>.

Environmental, social, and governance (esg). In: FILHO, W. L. et al. (Ed.). **Industry, Innovation and Infrastructure**. Cham: Springer International Publishing, 2021. p. 354–354. ISBN 978-3-319-95873-6. Disponível em: <https://doi.org/10.1007/978-3-319-95873-6_300064>.

GIRASA, R.; SCALABRINI, G. J. **Regulation of Innovative Technologies: Blockchain, Artificial Intelligence and Quantum Computing**. Springer International Publishing, 2022. ISBN 978-3-031-03868-6 978-3-031-03869-3. Disponível em: <https://link.springer.com/10.1007/978-3-031-03869-3>.

HOFMANN, H. W. H.; KAFADAR, K. Letter-value plots: Boxplots for large data. **Journal of Computational and Graphical Statistics**, Taylor & Francis, v. 26, n. 3, p. 469–477, 2017. Disponível em: <https://doi.org/10.1080/10618600.2017.1305277>.

HONG, G.; CHANG, H. A study on corporate information assets management system using nft. In: **2022 13th International Conference on Information and Communication Technology Convergence (ICTC)**. [S.l.: s.n.], 2022. p. 608–610.

HONG, S. et al. Fabasset: Unique digital asset management system for hyperledger fabric. In: **2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)**. [S.l.: s.n.], 2020. p. 1269–1274.

HYPERLEDGER. **A Blockchain Platform for the Enterprise**. 2020. Https://hyperledger-fabric.readthedocs.io/en/release-2.2/index.html.

HYPERLEDGER. **Hyperledger Blockchain Performance Metrics**. 2021. Https://www.hyperledger.org/wp-content/uploads/2018/10/HL_Whitepaper_Metrics_PDF_V1.01.pdf.

HYPERLEDGER. **Hyperledger Caliper**. 2023. Disponível em: <https://www.hyperledger.org/use/caliper>.

HYPERLEDGER. **Performance considerations**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/performance.html.

JIMÉNEZ, L. L. et al. Coma: Resource monitoring of docker containers. In: **CLOSER**. [S.l.: s.n.], 2015. p. 145–154.

KEELE, S. et al. **Guidelines for performing systematic literature reviews in software engineering**. [S.l.], 2007.

LILJA, D. J. **Measuring Computer Performance: A Practitioner's Guide**. [S.l.]: Cambridge University Press, 2000.

LINDEN, T. van der; SHIRAZI, T. Markets in crypto-assets regulation: Does it provide legal certainty and increase adoption of crypto-assets? **Financial Innovation**, v. 9, n. 1, p. 22, Jan 2023. ISSN 2199-4730. Disponível em: <https://doi.org/10.1186/s40854-022-00432-8>.

LIU, C.; WANG, H. Crypto tokens and token offerings: An introduction. In: ____. **Cryptofinance and Mechanisms of Exchange: The Making of Virtual Currency**. Cham: Springer International Publishing, 2019. p. 125–144. ISBN 978-3-030-30738-7. Disponível em: <https://doi.org/10.1007/978-3-030-30738-7_8>.

LODOLCE, M.; HOWLEY, C. **Gartner Forecasts Worldwide Public Cloud End-User Spending to Surpass $675 Billion in 2024**. 2024. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2024-05-20-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-surpass-\675-billion-in-2024>.

MINYOUN-A; LIMDONG-KYUN. Performance analysis of consensus algorithm considering nft transaction stability. **The Journal of The Institute of Internet, Broadcasting and Communication**, , v. 22, n. 2, p. 151–157, 04 2022.

MOSS. **The one-stop-shop for carbon solutions**. 2023. Https://moss.earth/en.

NAKAMOTO, S. **Bitcoin: A Peer-to-Peer Electronic Cash System**. 2008. Accessed: 2023-05-05. Disponível em: <https://bitcoin.org/bitcoin.pdf>.

NGUYEN, G.-T.; KIM, K. A survey about consensus algorithms used in blockchain. **Journal of Information processing systems**, v. 14, n. 1, 2018.

OLSSON, O. **A Taxonomy of Non-fungible Tokens: Overview, evaluation and explanation**. 2022. Disponível em: <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-477803>.

PAHL, C. et al. Cloud container technologies: a state-of-the-art review. **IEEE Transactions on Cloud Computing**, PP, p. 1–1, 05 2017.

P.H., M. et al. **Environmental reserve quotas in Brazil's new forest legislation: An ex ante appraisal**. Center for International Forestry Research (CIFOR), 2015. ISBN 978-602-387-004-2. Disponível em: <http://www.cifor.org/library/5609/environmental-reserve-quotas-in-brazils-new-forest-legislation-an-ex-ante-appraisal/>.

RASOLROVEICY, M.; FOKAEFS, M. Performance and cost evaluation of public blockchain: An nft marketplace case study. In: **2022 4th Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)**. [S.l.: s.n.], 2022. p. 79–86.

SAKIZ BURCU, G. A. H. Blockchain beyond cryptocurrency: Non-fungible tokens. **International Conference of Eurasian Economies**, p. 154–161, 2021.

TECH, G. **Blockchain and tokenization of real estate**. 2021. Https://growthtech.com.br/2021/06/02/a-blockchain-e-a-tokenizacao-de-bens-imoveis/.

TERRY, Q.; FORTNOW, M. **The NFT Handbook: How to Create, Sell and Buy Non-Fungible Tokens**. Wiley, 2021. ISBN 9781119838395. Disponível em: <https://books.google.com.br/books?id=zDVCEAAAQBAJ>.

Tianfield, H. Cloud computing architectures. In: **2011 IEEE International Conference on Systems, Man, and Cybernetics**. [S.l.: s.n.], 2011. p. 1394–1399. ISSN 1062-922X.

TINU, N. A survey on blockchain technology- taxonomy, consensus algorithms and applications. **International Journal of Computer Sciences and Engineering O**, v. 6, may 2018.

VAIRAGADE, R. et al. Proposal on nft minter for blockchain-based art-work trading system. In: **2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)**. [S.l.: s.n.], 2022. p. 571–576.

WANG, Q. et al. **Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges**. 2021.

YAGA, D. et al. **NISTIR 8202 - Blockchain Technology Overview**. [S.l.], 2018. Disponível em: <https://doi.org/10.6028/NIST.IR.8202>.

ZHU, Y. et al. Digital asset management with distributed permission over blockchain and attribute-based access control. In: **2018 IEEE International Conference on Services Computing (SCC)**. [S.l.: s.n.], 2018. p. 193–200.