Provendo Provas-de-Conexão para Auditabilidade de Acesso à Rede utilizando Blockchains

Luiz Felipe Machado [1], Leticia Cardoso Rodrigues [1], Luiz Bueloni [2], Roberta Lima Gomes [2], Everson S. Borges [1] e Magnos Martinello [1]

[1] Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari, 514, Goiabeiras – 29.075-910 – Vitória – ES – Brasil

{luiz.f.machado,leticia,everson}@edu.ufes.br; {magnos,roberta}@inf.ufes.br

[2]PoP-ES : Ponto de Presença da RNP no ES

bueloni@pop-es.rnp.br

Abstract. *The secure implementation of log management mechanisms in Internet access systems is crucial to ensuring the integrity, confidentiality, and availability of information. Access providers face challenges in meeting regulatory and compliance requirements, as logs can be altered, deleted, or disputed. These records play a fundamental role in audits, forensic investigations, and the establishment of security baselines. In this context, this paper presents the concept of proofs-of-connection for Network Access Auditing using blockchains. The performance evaluation examines log processing time for various batch sizes and analyzes the system's average throughput (logs per second), considering the time required for hash generation, blockchain registration, and indexing. To assess multiple configurations, a queue model is used to provide quantitative results for different batch size settings.*

Resumo. *A implementação segura de mecanismos para gestão de registros (logs) em sistemas de acesso à Internet é crucial para garantir a integridade, confidencialidade e disponibilidade das informações. Provedores de acesso enfrentam desafios para cumprir exigências regulatórias e de conformidade, pois os logs podem ser alterados, apagados ou contestados. Esses registros desempenham um papel fundamental em auditorias, investigações forenses e na definição de linhas de base de segurança. Nesse sentido, este artigo apresenta o conceito de provas-de-conexão para auditoria de acesso à rede utilizando blockchains. A avaliação de desempenho examina o tempo de processamento de logs para diferentes tamanhos de lote (batch) e analisa a taxa média de transferência do sistema (logs por segundo), considerando o tempo necessário para a geração de hashes, registro na blockchain e indexação. Para avaliar múltiplas configurações, um modelo de fila é utilizado, fornecendo resultados quantitativos para diferentes tamanhos de batch.*

1. Introdução

A implementação de mecanismos seguros para a gestão de registros (logs) em sistemas de acesso à Internet é essencial para garantir a integridade, confidencialidade e disponibilidade das informações. Provedores de acesso enfrentam desafios significativos no

cumprimento de exigências regulatórias e normas de conformidade, uma vez que os logs estão sujeitos a alterações, exclusões ou contestações, comprometendo sua confiabilidade e valor probatório.

Diante desse cenário, este artigo propõe o uso da tecnologia blockchain como uma camada de confiança para garantir a auditabilidade do acesso à rede. Essa abordagem impede a refutação posterior dessas informações, assegurando conformidade com as políticas de acesso e facilitando a detecção rápida de desvios ou violações. Além disso, ao adotar esta solução, o ISP poderá melhorar significativamente sua capacidade de responder a auditorias regulatórias e de segurança, garantindo que todos os requisitos legais sejam atendidos de maneira eficiente e confiável.

Para isso, contratos inteligentes são utilizados para armazenar de forma transparente e irreversível registros que atuam como *provas-de-conexão*. Uma prova de conexão é um registro criptográfico que atesta a associação entre um dispositivo e sua respectiva atividade na rede em um determinado instante de tempo t. Essa prova é gerada a partir de hashes criptográficos extraídos de logs de sistemas como Firewall, DHCP e Radius, vinculando endereços IP alocados via DHCP a dispositivos específicos, correlacionando-os com as traduções de NAT realizadas pelo Firewall no instante t e associando endereços físicos MAC aos usuários autenticados via Radius.

O desafio está na escalabilidade do número de conexões por segundo em relação à quantidade de hashes que podem ser armazenados na blockchain. Este trabalho busca equilibrar a granularidade da referência de logs por hash e a taxa de armazenamento de hashes por segundo. A perspectiva é encontrar um compromisso viável entre a indexação e a taxa de registros no e.g. Hyperledger Fabric (HLF) [Androulaki et al. 2018].

Para avaliar a escalabilidade do sistema, é conduzida uma análise de desempenho baseada em dados processados pela aplicação responsável por intermediar a geração de impressões digitais (hashes). Essa aplicação organiza os logs em batches, armazenando-os em um banco de dados (e.g., Elasticsearch) e, posteriormente, registra os hashes na blockchain, garantindo integridade e auditabilidade. O estudo foca na medição do throughput, da latência de processamento e da capacidade de gerenciamento de carga do sistema proposto. Para analisar diferentes configurações, propomos um modelo de fila analítico, permitindo a obtenção de resultados quantitativos para distintos tamanhos de batch, auxiliando na otimização do desempenho.

2. Estado-da-Arte

Embora a blockchain tenha um grande potencial para auditoria de acesso à rede, sua adoção prática ainda é limitada, especialmente em ambientes corporativos e entre provedores de serviços de Internet (ISPs). Um dos principais obstáculos para a adoção plena da tecnologia blockchain é a escalabilidade, uma vez que esses sistemas apresentam baixo throughput e alta latência em comparação com soluções tradicionais [Sanka and Cheung 2021].

Além disso, na literatura existem outros sistemas que combinam blockchain com tecnologias de privacidade, como zero-knowledge proofs (ZKP) e hashing criptográfico [Williams 2024], permitindo a verificação dos logs sem expor informações sensíveis. Para mitigar impactos no desempenho da rede, também têm sido pro-

postos modelos híbridos que integram blockchain com sistemas tradicionais de gerenciamento de logs, como os Security Information and Event Management (SIEMs) [González-Granadillo et al. 2021]. Entretanto, no contexto da auditabilidade de acesso à rede, a blockchain tem sido explorada essencialmente em pesquisas acadêmicas e projetos experimentais, mas sua implementação em larga escala ainda enfrenta desafios significativos [Macedo and Campista 2020].

No que diz respeito à escalabilidade de blockchains, particularmente considerando Hyperledger Fabric (HLF) [Androulaki et al. 2018], a maioria dos estudos investiga a relação entre parâmetros de configuração e métricas de desempenho, como latência e vazão, sendo que alguns também exploram ajustes dinâmicos desses parâmetros para otimizar o desempenho da rede. Por exemplo, em [Roy and Ghosh 2024] os autores propõem uma abordagem para gerenciar o crescimento do livro-razão no Hyperledger Fabric. A estratégia adaptativa ajusta dinamicamente o "tamanho do bloco"e o "tempo limite do bloco"para otimizar o crescimento do livro-razão.

O presente trabalho se diferencia ao introduzir o conceito de "provas-de-conexão", i.e. um registro criptográfico que será salvo na blockchain e que estabelece a relação entre um dispositivo e sua atividade na rede. Essa abordagem garante a integridade das informações, impedindo a refutação posterior e assegurando conformidade com as políticas de acesso. Além disso, o desempenho do sistema de processamento de logs é analisado considerando throughput (logs/s), latência de processamento e capacidade de gerenciamento de carga. Por fim, um modelo analítico simplificado de fila é proposto para avaliar diferentes configurações, fornecendo resultados quantitativos para distintos tamanhos de batch e auxiliando na otimização do desempenho da solução.

3. Auditabilidade de Acesso à Rede: Arquitetura e Implementação

A arquitetura do sistema de auditoria deve permitir o armazenamento, o processamento e a integridade dos logs provenientes de diversas fontes, tais como Firewall, DHCP e Radius. Esses logs desempenham papéis críticos: os logs de Firewall são essenciais para compreender a gestão das conexões e do NAT; os logs de DHCP possibilitam correlacionar endereços IP aos respectivos endereços MAC; e os logs do sistema Radius são fundamentais para a autenticação e autorização dos usuários na rede.

Para atender a esses requisitos, o sistema é organizado em múltiplas camadas, onde cada nível cumpre uma função específica – desde a coleta inicial dos dados, passando pelo processamento e armazenamento, até a recuperação e verificação final dos registros. Uma visão geral desse sistema é ilustrada na figura 1. Nesse caso, há um fluxo de logs coletados de diferentes servidores e pré-processados no logstash. Após essa filtragem inicial, os dados são enviados à aplicação que será responsável por intermediar a geração de impressão digital (*hash*) e associar esta a um lote de logs (**batch**) no elasticsearch e salvá-la na blockchain.

3.1. Coleta de Logs

Conforme ilustrado na Figura 1, os logs são capturados de diversas fontes e encaminhados via syslog para um contêiner do Logstash, onde são inicialmente armazenados em um buffer (por exemplo, no Redis) para absorver picos de requisições. Em seguida, outro contêiner com uma instância separada do Logstash consome esses logs, aplica filtros Grok

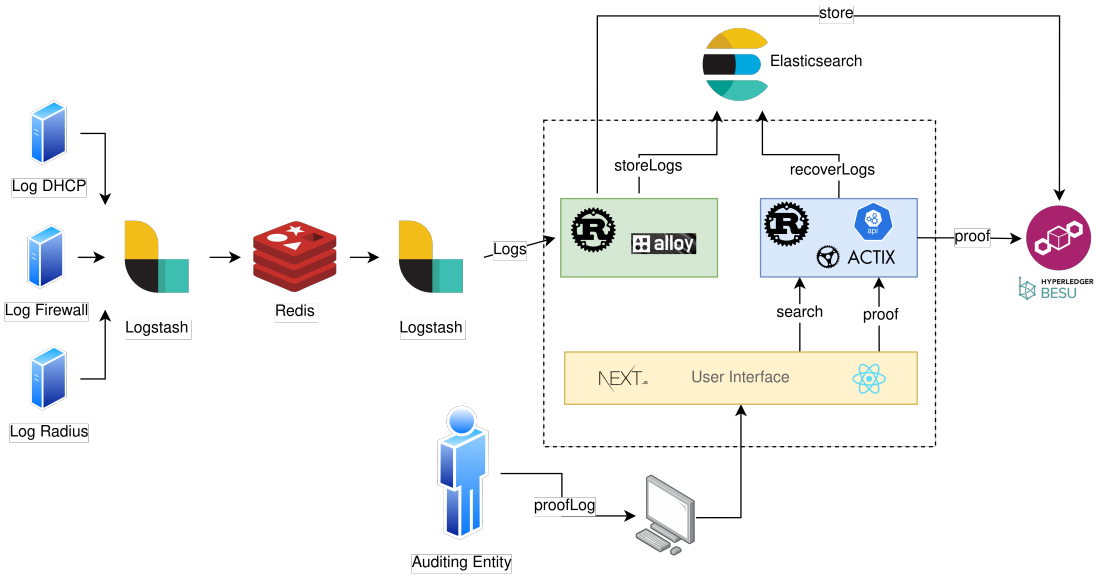


Figura 1. Arquitetura do Sistema de Auditabilidade de Acesso à Rede

para convertê-los em formato JSON e os encaminha para a aplicação responsável pela agregação e processamento dos dados.

Após essa etapa, um terceiro contêiner do Logstash consome os logs armazenados no buffer. Durante o processamento, filtros Grok são utilizados para estruturar os dados, facilitando sua consulta e análise. O documento processado é então enviado ao Dapp, onde os logs são manipulados e preparados para indexação. Simultaneamente, é gerado um hash SHA-256 para cada batch de logs, que é enviado à blockchain, garantindo a verificação e integridade dos dados.

3.2. Processamento e Envio para Blockchain e ElasticSearch

O processamento dos logs segue um fluxo bem definido, conforme ilustrado na Figura 2. Os logs são recebidos via HTTP e encaminhados a uma fila interna por meio de um despachante (dispatcher). Os workers consomem dessa fila, acumulando um batch de logs para gerar o hash desse conjunto de dados. Esse hash é calculado pela função `fingerprint`, que implementa o algoritmo SHA-256 de forma incremental, conforme mostrado no Código 1.

Após a geração do hash, os workers enviam essa informação, juntamente com o índice associado, para uma thread responsável por armazená-los na blockchain. Simultaneamente, o batch de logs e o índice são transmitidos para outra thread, encarregada de enviá-los ao ElasticSearch.

A função `fingerprint` aplica um método de hash incremental, processando individualmente cada documento JSON do batch. Para cada documento no vetor de entrada, a função inicializa um novo objeto hasher SHA-256, incorpora o hash acumulado até o momento e adiciona o conteúdo do documento atual convertido para string. O resultado final é uma string hexadecimal que representa o hash único do batch completo, garantindo que qualquer alteração em qualquer documento do batch resulte em um hash completamente diferente. Isso possibilita a verificação da integridade dos dados na blockchain.

```rust
pub fn fingerprint(content: &Vec<Json>) -> String {
    let mut hash = String::new();

    for doc in content.iter() {
        let mut hasher = Sha256::new();
        hasher.update(hash.as_bytes());
        hasher.update(doc.to_string());
        hash = format!("{:x}", hasher.finalize());
    }

    hash
}
```

Listing 1. Código Rust para geração de hash incremental de um batch de logs.

O parâmetro *Batch Size* pode ser configurado conforme as necessidades da aplicação e influencia diretamente seu desempenho. Esse parâmetro determina a quantidade de documentos JSON processados em conjunto antes da geração do hash e do envio para a blockchain e o ElasticSearch. Um *Batch Size* maior reduz a sobrecarga de operações de rede e processamento, diminuindo o número total de transações na blockchain e operações de indexação no ElasticSearch. Por outro lado, um *Batch Size* menor proporciona uma atualização mais frequente dos dados, mas com maior custo computacional e de rede. A escolha do valor ideal depende de fatores como a taxa de geração de logs, capacidade computacional disponível e requisitos de latência. Uma análise mais detalhada será apresentada na próxima seção.

Além do *Batch Size*, a quantidade de threads alocadas para *Dispatchers* e *Workers* também impacta significativamente o desempenho do sistema. A configuração ideal depende de fatores como arquitetura de hardware, volume de logs processados e características da rede. Um balanceamento adequado entre threads de *Dispatchers* (responsáveis por receber e enfileirar os logs) e threads de *Workers* (encarregadas do processamento, geração de hash e envio dos dados) pode otimizar o *throughput* e reduzir a latência. No entanto, a determinação precisa desses parâmetros exige análises de escalabilidade e testes de carga, que fogem do escopo deste trabalho e serão abordados em pesquisas futuras dedicadas à otimização de desempenho da infraestrutura.

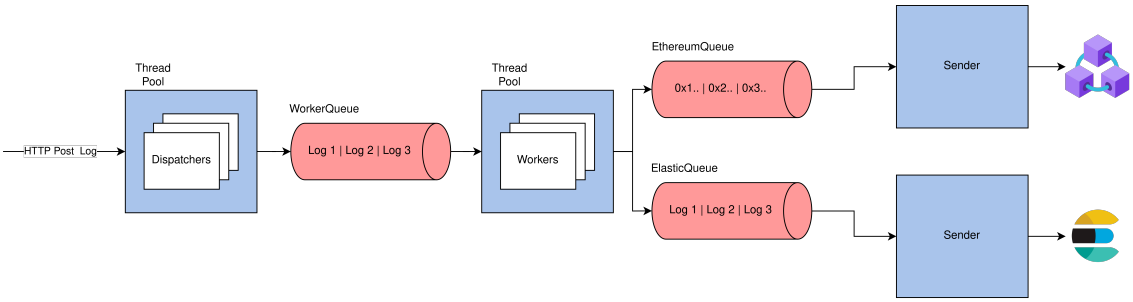


Figura 2. Arquitetura de Processamento da Dapp

3.3. Contratos Inteligentes e Blockchain

Para garantir a imutabilidade e a rastreabilidade dos registros, o sistema utiliza contratos inteligentes implementados na plataforma Hyperledger Besu, com os contratos escritos

em Solidity. O contrato expõe uma API que possibilita armazenar um hash associado a um índice do Elasticsearch. Esse hash, gerado a partir dos logs processados conforme a função `fingerprint` apresentada anteriormente, é registrado de forma permanente na blockchain, funcionando como uma "impressão digital"criptográfica dos dados originais.

A implementação do contrato inteligente, demonstrada no Código 2, estabelece um mecanismo para verificação de integridade através de duas funções principais: `store` para registrar novos hashes na blockchain e `proof` para validar a integridade dos dados armazenados no Elasticsearch. Esta abordagem permite que qualquer parte interessada possa verificar se os registros consultados mantêm-se inalterados desde seu armazenamento inicial, pois qualquer modificação não autorizada nos dados resultaria em um hash diferente, sendo imediatamente detectada durante o processo de verificação.

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.27;
contract Auditability {
    address public owner;
    struct IndexData {
        bytes32 hash;
        bool exists;
    }
    mapping(string => IndexData) private indices;
    event IndexStored(string indexed index, bytes32 hash);
    constructor() {
        owner = msg.sender;
    }
    function store(string memory index, bytes32 hash) public onlyOwner
    {
        require(!indices[index].exists, "Index already added.");
        indices[index] = IndexData({hash: hash, exists: true});
        emit IndexStored(index, hash);
    }
    function proof(string memory index, bytes32 hash) public view
   returns (bool) {
        require(indices[index].exists, "Index not found.");
        return indices[index].hash == hash;
    }
    function exists(string memory index) public view returns (bool) {
        return indices[index].exists;
    }
    modifier onlyOwner() {
        require(msg.sender == owner, "Only owner.");
        _;
    }
}
```

Listing 2. Contrato Inteligente para armazenamento e verificação de hashes de logs.

O contrato `Auditability` implementa um sistema de registro baseado em um mapeamento chave-valor, onde cada chave representa um índice do Elasticsearch e o valor corresponde ao hash SHA-256 dos logs associados. O contrato emite eventos para cada operação de armazenamento, criando um registro imutável e auditável de todas as interações. A função `proof` permite comparar o hash original de um determinado ín-

dice, fornecendo transparência adicional ao processo de verificação. Esta integração entre blockchain e Elasticsearch proporciona um mecanismo de auditoria confiável e resistente a adulterações, essencial para sistemas de log que exigem alta confiabilidade e conformidade com requisitos regulatórios.

A escolha da plataforma Hyperledger Besu para a implementação apresenta vantagens significativas para este contexto, principalmente por ser uma blockchain que oferece controle de permissões e privacidade, características essenciais para sistemas de auditoria. O modificador `onlyOwner` no contrato implementa um mecanismo básico de controle de acesso, garantindo que apenas entidades autorizadas possam registrar novos hashes. Este aspecto é particularmente relevante em ambientes corporativos ou governamentais, onde a autoridade para validar registros deve ser restrita a entidades específicas. Adicionalmente, o uso de eventos (`event IndexStored`) não apenas facilita a rastreabilidade das operações, mas também possibilita a implementação de sistemas de notificação para alertar sobre novas inclusões de logs, ampliando as capacidades de monitoramento do sistema.

3.4. Considerações de segurança e modelo de confiança

Embora a tecnologia blockchain forneça garantias robustas de imutabilidade e auditabilidade após o registro dos hashes, é essencial analisar a segurança dos componentes que precedem este registro. O sistema proposto opera sob um modelo de confiança híbrido, onde a infraestrutura blockchain oferece propriedades de não-repúdio para os registros confirmados, enquanto os componentes de coleta e processamento inicial dos logs dependem de medidas de segurança convencionais.

4. Avaliação de Desempenho

A análise de desempenho desta implementação é essencial, considerando o volume massivo de dados gerado por logs de acesso à rede. O sistema deve processar, armazenar e verificar grandes quantidades de registros mantendo eficiência operacional e escalabilidade da infraestrutura. Este estudo avalia especificamente o throughput, a latência de processamento e a capacidade de gerenciamento de carga do sistema proposto.

Considerando que cada registro de log passa por um processo de hash criptográfico e posterior armazenamento em contratos inteligentes, a sobrecarga computacional destas operações requer análise minuciosa. Um aspecto crítico é a taxa de transações na blockchain, que deve ser adequadamente dimensionada para garantir auditabilidade contínua sem introduzir gargalos no processamento do fluxo de dados. Os experimentos realizados avaliam diferentes configurações, incluindo uma configuração básica com apenas um nó no Hyperledger Besu e variações na frequência de armazenamento de registros na blockchain.

4.1. Configuração do Ambiente Experimental

O ambiente experimental utilizado neste estudo foi configurado em um computador pessoal com processador Intel i7-1250U (12 núcleos, 4,7 GHz), 32 GB de memória RAM e disco SSD NVMe de 512 GB. Todo o sistema foi executado na mesma máquina, com apenas um nó Hyperledger Besu.

A implementação do sistema de auditabilidade contempla múltiplos componentes, todos rodando localmente:

- Infraestrutura Blockchain: rede Hyperledger Besu v25.2.2 composta por um único nó validador em configuração de consenso IBFT 2.0, executado na máquina descrita acima.
- Processamento de Logs: aplicação desenvolvida em Rust v1.86.0, executada no mesmo host.
- Indexação e Armazenamento: cluster Elasticsearch v8.15.2, simulado com apenas uma instâncias local na mesma máquina.
- Testes de Carga: ferramenta Grafana k6 v1.0.0, também executada localmente para geração de tráfego simulado.

A interligação dos componentes foi realizada via interface de loopback local, eliminando praticamente qualquer latência de rede nos resultados experimentais.

Geração de carga: Para avaliar o desempenho do sistema proposto, utilizou-se a ferramenta **k6 grafana**[1], que permite simular o uso real do sistema em um ambiente controlado. O teste consistiu em simular $n = 1000$ usuários virtuais enviando requisições a cada 1 segundo. A simulação envia logs sintéticos durante 1 minuto. O intervalo de um segundo entre requisições sucessivas foi estabelecido para manter uma carga constante e mensurável, facilitando a análise comparativa entre diferentes configurações do sistema.

4.2. Tempo de Processamento e Throughput Médio

O sistema foi submetido a uma bateria de testes com diferentes configurações de *Batch Size* para avaliar o desempenho das etapas críticas do processamento de logs. Os cenários de teste abrangeram um espectro amplo, com variações no número de logs processados por lote de 1.000 a 500.000 registros. As avaliações concentraram-se em três métricas fundamentais:

- **Latência para geração de hash**: Tempo médio necessário para calcular o hash criptográfico SHA-256 de um lote completo de logs;
- **Latência para registro na blockchain**: Tempo médio para transmitir o hash à rede blockchain e obter confirmação da transação;
- **Latência para indexação**: Tempo médio para armazenar e indexar o conjunto completo de logs no Elasticsearch.

A análise dos resultados apresentados na Figura 3 como boxplot, revela comportamentos distintos para cada componente do sistema. Para lotes pequenos (1.000 logs) Fig.3a, observa-se latência reduzida nas operações de geração de hash e indexação, enquanto o tempo de registro na blockchain permanece elevado devido ao custo fixo das transações e do algoritmo de consenso na blockchain.

Em configurações intermediárias (10.000 Fig.3b a 100.000 logs Fig.3c), identifica-se um aumento proporcional na latência de geração de hash e indexação, com comportamento aproximadamente linear em relação ao tamanho do lote. No entanto, o tempo de registro na blockchain mantém-se relativamente constante, o que é esperado, considerando que apenas uma única transação é necessária independentemente do volume de logs no lote, já que apenas o hash e o identificador do índice são registrados.

[1] https://k6.io

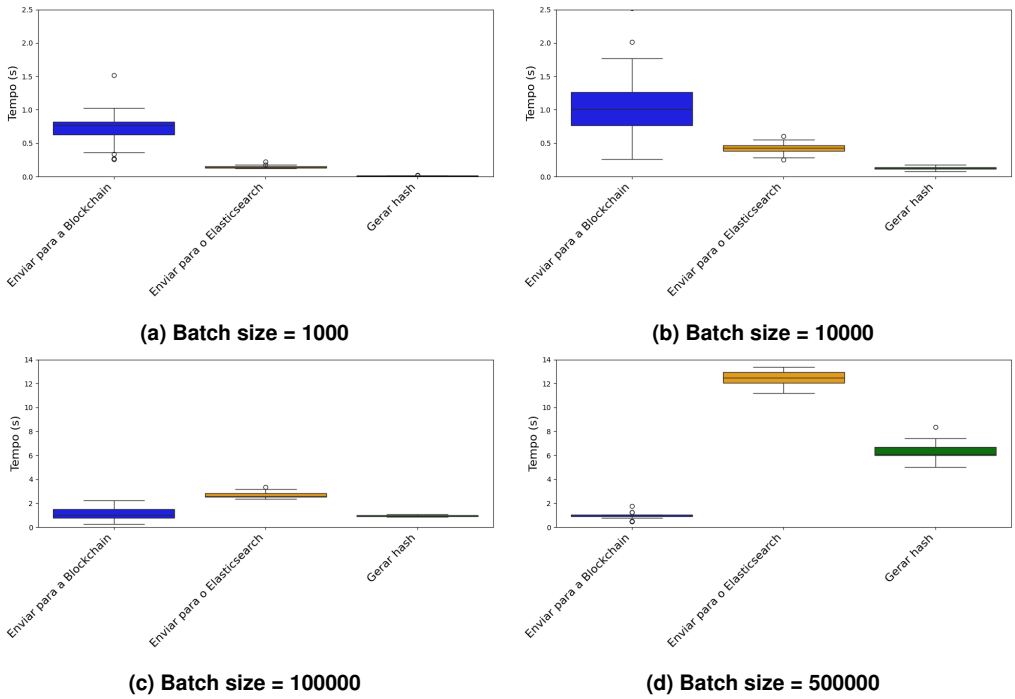


Figura 3. Boxplot do tempo de processamento para cada etapa do sistema em função do tamanho do lote.

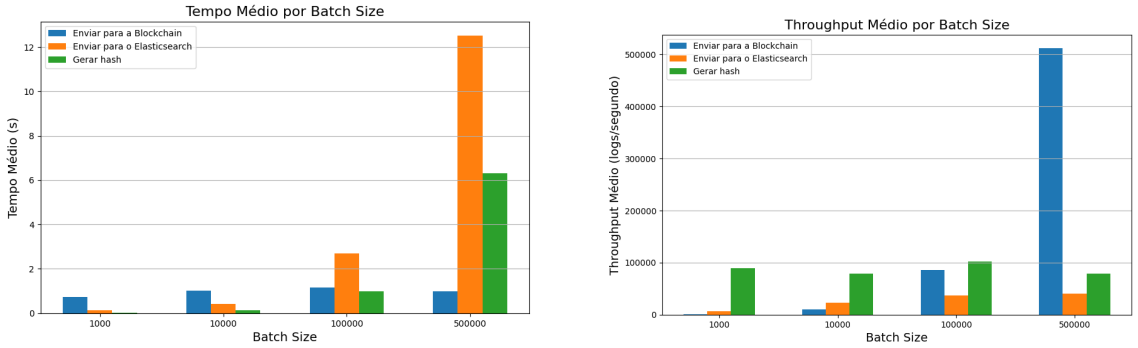
Para lotes de grande dimensão (500.000 logs) Fig.3d, constata-se um crescimento mais acentuado nas latências de geração de hash e indexação. Este comportamento não-linear sugere que, a partir de determinado volume, fatores como limitações de memória e concorrência por recursos computacionais começam a influenciar o desempenho do sistema. Esse resultado é confirmado pelo tempo médio de processamento para cada componente na Figura 4(A).

Embora as métricas de tempo médio forneçam insights valiosos sobre o comportamento individual de cada componente, o *throughput* médio oferece uma perspectiva mais abrangente sobre a eficiência global do sistema. Esta métrica, definida como a quantidade de logs processados por unidade de tempo (logs/segundo), é essencial para avaliar a capacidade real de processamento sob diferentes cargas de trabalho.

A Figura 4b demonstra o comportamento do *throughput* médio em função do tamanho do lote. Para configurações com lotes pequenos, observa-se um *throughput* reduzido, consequência direta da sobrecarga operacional fixa associada às transações blockchain e operações de indexação, que não são amortizadas eficientemente quando distribuídas por um número limitado de registros.

À medida que o tamanho do lote aumenta, verifica-se um crescimento significativo no *throughput*, corroborando a hipótese de que o tempo constante necessário para registro na blockchain se torna proporcionalmente menos impactante quando distribuído por um volume maior de logs. Este comportamento confirma a eficácia da estratégia de processamento em lotes para sistemas que integram tecnologias blockchain e mecanismos de indexação.

Como evidenciado na Figura 4(A), o tempo médio para registro na blockchain



(a) Tempo médio de processamento **(b) Throughput médio (logs/segundo)**

Figura 4. Tempo médio de processamento e Throughput médio para cada etapa em função do tamanho do lote.

permanece praticamente invariável independentemente do tamanho do lote. Consequentemente, o *throughput* médio apresenta uma correlação positiva com o *Batch Size*, confirmando o ganho de eficiência proporcionado pelo processamento em lotes maiores, conforme ilustrado na Figura 4b. É importante ressaltar que, embora os logs completos não sejam armazenados diretamente na blockchain, o hash registrado funciona como uma impressão digital criptográfica que garante a integridade e autenticidade de todo o conjunto de dados.

4.3. Análise de Latência e Identificação de Gargalos

A análise de latência do sistema foi conduzida através de testes de carga sistemáticos, utilizando a ferramenta k6 grafana, com um fluxo constante de 1.000 logs por segundo. O principal objetivo desta investigação foi estabelecer uma correlação entre o tamanho do lote de processamento (Batch Size) e métricas críticas de desempenho, especificamente a latência de resposta e a capacidade de processamento efetiva (throughput).

Os resultados experimentais, apresentados na Figura 5, evidenciam uma relação inversamente proporcional entre o tamanho do lote e a latência do sistema. Para configurações com lotes de dimensões reduzidas, observou-se um aumento substancial na latência média de resposta. Este fenômeno pode ser atribuído à formação de um gargalo operacional no pipeline de processamento, originado pela capacidade limitada de processamento e transmissão de hashes para a infraestrutura blockchain.

A investigação detalhada da arquitetura revelou um padrão de bloqueio em cascata que compromete o desempenho global do sistema: inicialmente, os componentes Worker são bloqueados aguardando que o módulo Sender conclua a transmissão do hash para a blockchain; subsequentemente, o componente Dispatcher também entra em estado de espera devido à indisponibilidade dos Workers. Este efeito cumulativo manifesta-se como um aumento na latência percebida pelos usuários finais do sistema.

Em contrapartida, configurações utilizando lotes de maior dimensão demonstraram desempenho substancialmente superior, com redução significativa da latência. Como pode ser observado na Figura 6, o sistema mantém capacidade consistente de processamento para a totalidade das 1.000 requisições por segundo quando operando com tamanhos de lote otimizados. Esta melhoria é consequência direta da amortização do custo

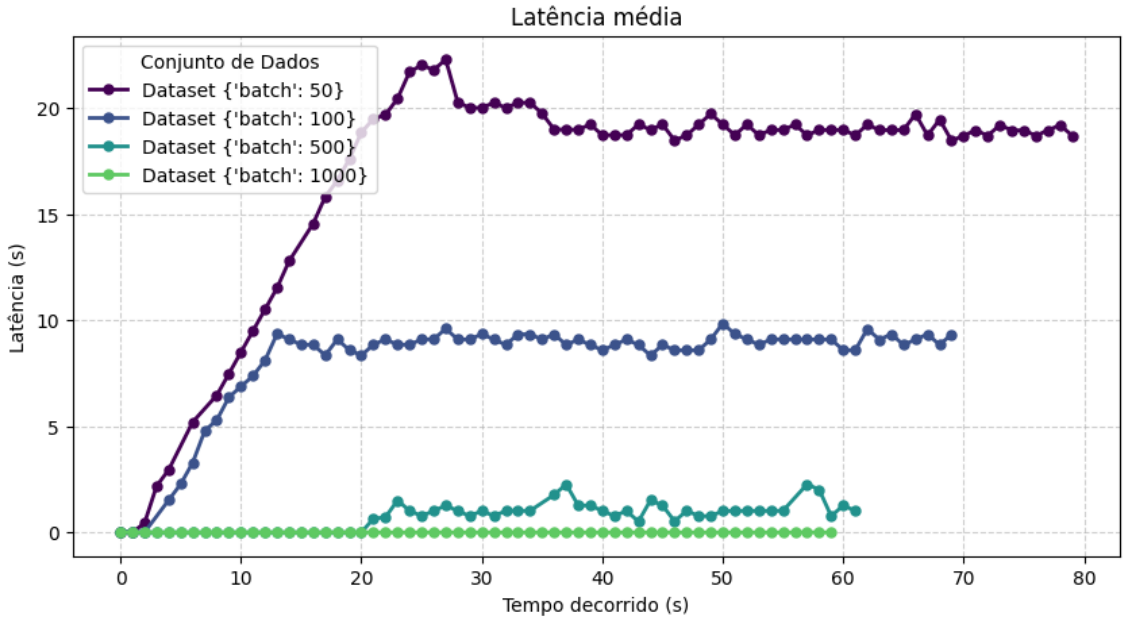


Figura 5. Latência média de resposta em função do tamanho do lote (Batch Size). A análise demonstra que configurações com lotes menores apresentam degradação significativa de desempenho devido à formação de gargalos nas estruturas de fila do sistema.

fixo associado às operações de blockchain e do aumento na eficiência de utilização dos recursos computacionais disponíveis.

A principal otimização observada com lotes maiores deriva da maior eficiência do módulo Sender ao processar conjuntos mais volumosos de dados, reduzindo a frequência de interações com a infraestrutura blockchain e, consequentemente, minimizando a contenção nos componentes anteriores do pipeline. Esta redução na pressão sobre os módulos Worker e Dispatcher permite que o sistema mantenha fluxo contínuo de processamento, mesmo sob carga elevada e constante.

4.4. Relação entre Batch Size e Comportamento da Fila

Este estudo concentra-se na análise da fila de envio para blockchain, identificada como o principal gargalo do sistema. Embora o processo de envio de batches para o Elasticsearch também exija recursos computacionais significativos, este componente pode ser otimizado através da paralelização com múltiplas threads. Em contraste, as operações de escrita na blockchain demandam maior cautela na implementação de concorrência, devido à necessidade de garantir integridade transacional.

4.4.1. Modelo Analítico de Fila

Para analisar quantitativamente o comportamento do sistema, adotamos o modelo de fila M/M/1, caracterizado pelos seguintes parâmetros e pressupostos:

- Processo de chegada seguindo distribuição de Poisson com taxa λ
- Tempo de serviço com distribuição exponencial e taxa μ

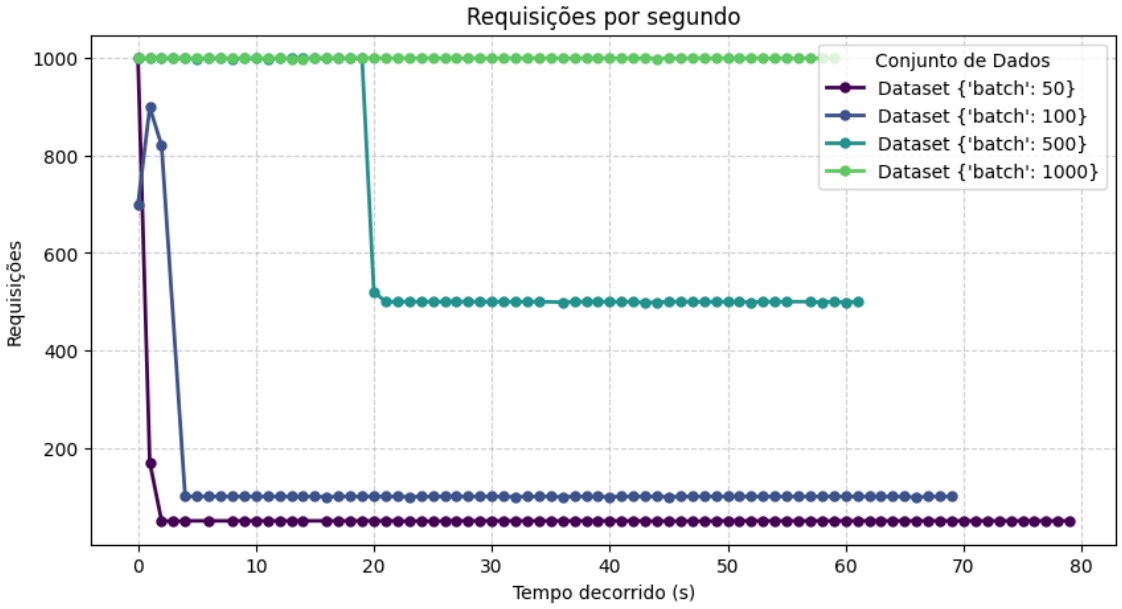


Figura 6. Capacidade de processamento (throughput) do sistema em função do tamanho do lote. Configurações com lotes de maior dimensão permitem ao sistema processar consistentemente 1.000 requisições por segundo sem degradação de desempenho.

- Capacidade infinita de armazenamento na fila
- Disciplina de atendimento FIFO (First-In, First-Out)

Esta modelagem é particularmente apropriada para o cenário em questão, onde temos um único servidor (uma thread dedicada para escrita na blockchain) processando requisições que chegam de forma estocástica. O modelo M/M/1 permite derivar métricas de desempenho através de relações matemáticas bem estabelecidas na teoria de filas.

Um aspecto fundamental na análise é a relação entre a taxa de entrada de logs brutos (λ') e a taxa efetiva de entrada de batches (λ) no sistema, dada pela equação:

$$\lambda = \frac{\lambda'}{BS} \tag{1}$$

onde BS representa o Batch Size, ou seja, o número de logs agrupados em cada unidade de processamento enviada à blockchain.

A Tabela 1 apresenta as principais relações matemáticas derivadas do modelo M/M/1, essenciais para a análise quantitativa do sistema:

4.4.2. Análise Comparativa baseada em modelo analítico de filas

Para avaliar o impacto do Batch Size no desempenho do sistema, conduzimos uma análise comparativa considerando os seguintes parâmetros operacionais:

- Taxa de entrada de logs: $\lambda' = 1000\ \text{logs/s}$
- Capacidade de processamento: $\mu = 500\ \text{hashes/s}$

Métrica	**Fórmula**	**Equação**
Taxa de utilização do sistema	$\rho = \frac{\lambda}{\mu}$	(2)
Probabilidade de n hashes no sistema	$P(n) = \rho^n(1-\rho)$	(3)
Probabilidade de mais de k hashes	$P(n > k) = \rho^{k+1}$	(4)
Tempo médio de espera na fila	$W_q = \frac{\rho}{\mu(1-\rho)}$	(5)
Tamanho médio da fila	$L_q = \frac{\rho^2}{1-\rho}$	(6)

Tabela 1. Principais fórmulas do modelo de fila M/M/1

A Tabela 2 apresenta os resultados quantitativos para diferentes configurações de Batch Size:

Batch Size	λ (hashes/s)	ρ	W_q (s)	L_q (hashes)	$P(n > 5)$
1 (Processamento serial)	1000	2.0	—	—	—
10 (Configuração mínima)	100	0.2	0.0005	0.05	0.00032
50 (Configuração otimizada)	20	0.04	0.000083	0.0017	1.0×10^{-7}
100 (Superdimensionado)	10	0.02	0.000041	0.00041	3.2×10^{-9}

Tabela 2. Comportamento do sistema com diferentes configurações de Batch Size

Os resultados demonstram relações não-lineares entre o Batch Size e as métricas de desempenho do sistema. No cenário de processamento serial (BS = 1), observa-se uma condição crítica onde $\rho = 2.0 > 1$, indicando instabilidade sistêmica e crescimento ilimitado da fila, o que inviabiliza a operação prática nesta configuração.

À medida que aumentamos o Batch Size, verificamos um impacto desproporcional nas métricas de desempenho. A transição de BS = 10 para BS = 50 reduz a taxa de utilização do sistema em 80% (de 0.2 para 0.04), enquanto o tamanho médio da fila (L_q) diminui em aproximadamente 97% (de 0.05 para 0.0017). Mais significativamente, a probabilidade de acúmulo de requisições na fila ($P(n > 5)$) decresce em três ordens de magnitude, passando de 3.2×10^{-4} para 1.0×10^{-7}.

A configuração otimizada (BS = 50) apresenta um equilíbrio entre eficiência operacional e responsividade do sistema, com um tempo médio de espera na fila (W_q) de apenas 83 microssegundos e uma taxa de utilização (ρ) de 0.04, que proporciona uma margem significativa para absorver picos momentâneos de tráfego.

É importante ressaltar que este modelo não incorpora a latência adicional introduzida pelo processo de agrupamento dos logs, que aumenta linearmente com o Batch Size. Portanto, embora configurações com valores extremamente altos (BS = 100) apresentem métricas superiores, podem comprometer a responsividade geral do sistema devido ao tempo necessário para a formação dos batches. Esta observação sugere a existência de um ponto ótimo específico para cada cenário operacional, que equilibra os benefícios do processamento em lote com as restrições de latência aceitáveis pela aplicação.

5. Conclusão

Os resultados demonstram que o uso de blockchain para auditoria de acesso à rede, por meio de provas-de-conexão, proporciona maior integridade e transparência na gestão de logs. A implementação de contratos inteligentes garante que os registros sejam armazenados de forma imutável e verificável, reduzindo o risco de manipulação ou perda de informações. Além disso, a abordagem proposta facilita auditorias e investigações forenses, permitindo que provedores de serviços de Internet (ISPs) e organizações corporativas assegurem conformidade com regulamentações de segurança e privacidade. A análise de desempenho evidenciou que a otimização da estrutura de armazenamento, combinada com a indexação eficiente dos registros, melhora significativamente a escalabilidade do sistema.

Como trabalhos futuros planeja-se investigar os custos computacionais associados à execução dos *smart contracts* (consumo de gás) em diferentes cenários de operação. Além disso, validar o modelo em ambientes de produção, especialmente no PoP-ES (Ponto de Presença da RNP no Espírito Santo), para avaliar a viabilidade da proposta.

6. Agradecimentos

Agradecemos ao projeto Ilíada, da RNP, pelo financiamento do GT-Audita [2] . Nosso reconhecimento especial para Reinaldo Gomes, pelo acompanhamento técnico e pelas valiosas orientações ao longo do desenvolvimento deste trabalho. Pelo apoio da FAPES (06/2022, 1026/2022, 941/2022, 20/2022). Prof. Magnos Martinello é apoiado pelo CNPq, 312058/2023-3.

Referências

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S. W., and Yellick, J. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA. Association for Computing Machinery.

González-Granadillo, G., González-Zarzosa, S., and Diaz, R. (2021). Security information and event management (siem): Analysis, trends, and usage in critical infrastructures. *Sensors*, 21(14).

Macedo, L. O. and Campista, M. E. (2020). Tecnologia blockchain para auditoria em redes móveis. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 798–811, Porto Alegre, RS, Brasil. SBC.

Roy, U. and Ghosh, N. (2024). Fabman: A framework for ledger storage and size management for hyperledger fabric-based iot applications. *IEEE Transactions on Network and Service Management*, 21(3):3140–3151.

Sanka, A. I. and Cheung, R. C. (2021). A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *J. Netw. Comput. Appl.*, 195(C).

Williams, A. (2024). Zero-knowledge proofs and their role within the blockchain. *Commun. ACM*, 67(7):6–7.

[2]https://github.com/nerds-ufes/GT-Audita