Ericksulino Manoel de Araújo Moura
Orientador: Glauber Dias Gonçalves

Comparação e Análise de Custo e Desempenho entre Nós de Redes Blockchain Permissionadas e Públicas

Picos - PI
22 de janeiro de 2024

Ericksulino Manoel de Araújo Moura
Orientador: Glauber Dias Gonçalves

Comparação e Análise de Custo e Desempenho entre Nós de Redes Blockchain Permissionadas e Públicas

Monografia submetida ao Curso de Bacharelado em Sistemas de Informação como requisito parcial para obtenção de grau de Bacharel em Sistemas de Informação.

Universidade Federal do Piauí
Campus Senador Heuvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Picos - PI
22 de janeiro de 2024

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

M929c Moura, Ericksulino Manoel de Araújo.
Comparação e análise de custo e desempenho entre Nós de Redes Blockchain permissionadas e públicas./ Ericksulino Manoel de Araújo Moura. – 2022.
46 f.

1 Arquivo em PDF
Indexado no catálogo *online* da biblioteca José Albano de Macêdo-CSHNB
Aberto a pesquisadores, com restrições da Biblioteca

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Piauí, Curso de Bacharelado em Sistemas de Informação, Picos, 2022.
"Orientadora: Prof. Glauber Dias Gonçalves"

1. Blockchain-tecnologia. 2. Custo-desempenho. 3. Informação-sistemas.
I. Moura, Ericksulino Manoel de Araújo. II. Gonçalves, Glauber Dias.
II. III. Título.

CDD 658.403

Elaborado por Sérvulo Fernandes da Silva Neto CRB 15/603

COMPARAÇÃO E ANÁLISE DE CUSTO E DESEMPENHO ENTRE NÓS DE REDES BLOCKCHAIN PERMISSIONADAS E PÚBLICAS

ERICKSULINO MANOEL DE ARAÚJO MOURA

Monografia aprovada como exigência parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 1 de fevereiro de 2024

Documento assinado digitalmente
gov.br
GLAUBER DIAS GONCALVES
Data: 01/02/2024 15:29:52-0300
Verifique em https://validar.iti.gov.br

Prof. Glauber Dias Gonçalves

Documento assinado digitalmente
gov.br
FRANCISCO AIRTON PEREIRA DA SILVA
Data: 05/02/2024 14:46:54-0300
Verifique em https://validar.iti.gov.br

Prof. Francisco Airton Pereira da Silva

Documento assinado digitalmente
gov.br
CARLOS ALEXANDRE SILVA DE MELO
Data: 02/02/2024 11:23:10-0300
Verifique em https://validar.iti.gov.br

Prof. Carlos Alexandre Silva de Melo

Agradecimentos

Primeiramente, expresso minha profunda gratidão a Deus, pois a ele devemos toda a nossa gratidão. Em particular, quero agradecer por me conceder a força e a perseverança necessárias para enfrentar os desafios da vida acadêmica.

Gostaria de deixar meus agradecimentos a toda a minha família, em especial a minha mãe Maria das Dores de Araújo, primeiramente pelo dom da vida, por ter cuidado de muito com todo amor e cuidado mesmo apesar das difiuculdades e da jornada dupla de ser mãe e pai ao mesmo tempo, sempre me deu bons ensinamentos e incentivos que moldarem minha índole e caráter, o que me ajudou a ser quem sou hoje. Ao memória do meu Pai Edvaldo José de Moura, que mesmo não estando presente em todos os momentos, também deixou a sua parcela de contribuição na formação do meu carater. Também gostaria de agradecer a memoria da minha amada avó Maria Isabel de Araújo, que nós deixou recentemete e a ela tenho muito a agradeçer, pois ela ajudou de forma ativa na minha criação junto com meus tios Antônio Joquim de Araújo e Maria dos Remedios de Araújo, agradeçer a eles por todo o apoio e confiança que depositaram em mim, assim como meus outros tios e tias.

Ao professor Dr. Glauber Dias Gonçalves, por me acolher no seu grupo de pesquisa e como o seu orientando, com ele pude aprender tudo que sei sobre pesquisa científica, proporcionando novos aprendizados pessoais e profissionais, além de ser um exemplo de profissional. Agradeço a todos os professores que formam o corpo docente do curso de Sistemas de Informação, eles tiveram um papél fundamental na minha jornada nesse curso, com os seus ensinamentos valiosos e a metodologia incomparável em repassar seu conhecimento sobre os mais diversos conteúdos.

A todos os que integram o Laboratório de Pesquisas Aplicadas a Análises de Dados PAAD pelos conselhos, dicas e auxílio na minha caminhada acadêmica, assim como também no aprendizado de diversos outros assuntos relacionados à pesquisa e para vida, Em especial meus colegas e amigos, Humberto José da Silva Junior, Vitor José Ferreira dos Santos de Santana, Emanuel Aurélio Ferreira de Miranda, Eva Luana Almeida da Silva e Wendel dos Santos Nunes. Aos meus amigos e colegas de turma que me acompanharam nessa jornada de estudos e aprendizados, em especial ao meu colega e amigo José Miqueias de Araújo Pereira que vem me acompanhando e auxiliando desde o inico deste curso.

Por fim, agradeço a todos que me acolheram e contribuíram para minha jornada acadêmica de forma direta ou indireta, deixo aqui meus sinceros agradecimentos.

Inovação é a capacidade de ver a mudança como uma oportunidade não uma ameaça
Steve Jobs

Resumo

Blockchain é uma tecnologia que amplia a segurança nas relações entre organizações via o registro auditável e descentralizado de transações. Há uma crescente atenção por aplicações que utilizam essa tecnologia. Entretanto, a eficiência e custo de tais aplicações pode ser influenciada pela rede blockchain utilizada. De fato, a escolha da rede impacta nos requisitos não funcionais das aplicações, em especial desempenho (e.g., em relação a taxa de transações efetivadas) e custo. Este trabalho investiga o impacto no desempenho e custo da infraestrutura de rede blockchain para lidar com uma determinada carga de trabalho. Primeiramente, este trabalho de conclusão propõe um modelo de arquitetura de rede comum entre a rede pública Ethereum e permissionada Hyperledger Fabric com base em recursos por nó da rede blockchain. A seguir, avalia-se o custo por transação para aplicações nessa arquitetura considerando latências e custos mínimos para os pares da rede, em função da carga de trabalho. Os experimentos realizados nas plataformas, Ethereum e Hyperledger Fabric, mostram os limites de escalabilidade dessas plataformas e os seus compromissos entre custo e desempenho no projeto de aplicações baseadas em blockchain.

Palavras-chaves: Blockchain, Desempenho, Custo, Ethereum, Hyperledger.

Abstract

Blockchain is a technology that increases security in relationships between organizations via the auditable and decentralized record of transactions. There is growing attention for applications that use this technology. However, the efficiency and cost of such applications can be influenced by the blockchain network used. In fact, the choice of network impacts the non-functional requirements of applications, in particular performance (e.g., in relation to the rate of completed transactions) and cost. This work investigates the impact on performance and cost of blockchain network infrastructure to handle a given workload. Firstly, this conclusion work proposes a common network architecture model between the public Ethereum network and the permissioned Hyperledger Fabric based on resources per node of the blockchain network. Next, the cost per transaction for applications in this architecture is evaluated considering latencies and minimum costs for network peers, depending on the workload. The experiments carried out on the Ethereum and Hyperledger Fabric platforms show the scalability limits of these platforms and their compromises between cost and performance in the design of blockchain-based applications.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

API	*Application Programming Interface*
APM	*Active Passive Measurement*
AWS	*Amazon Web Services*
CDF	*Cumulative Distribution Function*
CLI	*Command Line Interface*
CSV	*Comma Separated Values*
DApps	*Decentralized Applications*
EC2	*Amazon Elastic Cloud Computing*
gRPC	*Google Remote Procedure Call*
KS	*Kolmogorov-Smirnov*
MSP	*Membership Service Provider*
MLE	*Maximum Likelihood Estimation*
P2P	*Peer to Peer*
PBFT	*Practical Byzantine Fault Tolerance*
PoS	*Proof of Stake*
PoW	*Proof of Work*
SWF	*Amazons Simple Workflow Service*
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
TPS	*Transactions per second*
UDP	*User Datagram Protocol*
VM	*Virtual machine*

Sumário

1 Introdução

Blockchain é uma tecnologia disruptiva com impactos nas relações entre pessoas, consumo e produção de bens e serviços (XU; WEBER; STAPLES, 2019). Essa tecnologia possibilita o registro seguro e descentralizado de dados ou transações entre entidades (pessoas e/ou organizações) que podem não se conhecer, e assim não terem confiança mútua. Logo, os dados e transações entre essas entidades são registrados de forma imutável, com acesso público ou privado para fins de verificação de autenticidade e derivação de novas transações. Isso se tornou possível a partir da evolução e unificação de outras tecnologias como criptografia assimétrica e protocolos de consenso distribuído via comunicação par a par, que são a essência de blockchains (GREVE et al., 2018).

Existe um crescente interesse por novas aplicações derivadas dessa tecnologia no meio corporativo e nos serviços públicos, além das já conhecidas aplicações para cripto ativos Bitcoin e Ethereum (NAKAMOTO, 2008; WOOD, 2014). Os recursos da tecnologia blockchain como os *contratos inteligentes* estendem o seu uso em diferentes domínios de aplicação corporativas (XU; WEBER; STAPLES, 2019). Os Contratos Inteligentes nada mais são que os contratos codificados e colocados em uma base de dados de execução automática e autônoma (ÁVILA, 2019). Contudo, essa tecnologia encontra-se ainda em fase de amadurecimento e necessita de ferramentas para gerenciamento de custos e recursos computacionais (i.e., infraestrutura) que permitirão a sua adoção por organizações em setores como indústria, serviços e governos. Atualmente os modelos de infraestrutura mais adotados para a tecnologia blockchain são *redes públicas* e *permissionadas* cada uma com características de desempenho e custos específicas.

As redes blockchain públicas foram as primeiras a serem desenvolvidas e são ainda as mais utilizadas. Plataformas populares como Ethereum permitem o desenvolvimento e execução de contratos inteligentes, sem restrição ao acesso ou uso desses recursos e constituem um intricado ecossistema de aplicações descentralizadas (DApps). Contudo, transações nessas redes podem levar minutos para serem confirmadas dado o grande número de usuários que as submetem e o consenso distribuído realizado pelos nós mantenedores da rede para validar transações[1]. Esses nós têm direito de gerar novos ativos (ou moeda) e adquiri-los (mineração), assim como cobrar tarifa aos usuários por transação confirmada.

Uma aplicação em rede pública requer um nó provedor de acesso para encaminhar transações requisitadas pelos seus usuários aos nós mantenedores. Existem provedores de acesso tais como a AWS[2], Alchemy[3], QuickNode[4] e Infura[5] que oferecem recursos de

[1] Desempenho da rede Ethereum em tempo real: https://etherscan.io
[2] https://docs.aws.amazon.com/blockchain-templates/
[3] https://www.alchemy.com/
[4] https://www.quicknode.com/
[5] https://infura.io

infraestrutura para esse nó como um serviço para facilitar o desenvolvimento de aplicações. Logo, o custo da aplicação consiste primordialmente no recurso computacional do nó provedor, ao passo que o usuário geralmente arca com a tarifa da transação.

Por sua vez, uma rede blockchain permissionada (ANDROULAKI; et al., 2018) é uma alternativa atrativa para organizações que possuem infraestrutura e corpo técnico próprios, visando escapar de questões de custos (tarifação) e desempenho instáveis das redes blockchains públicas como Ethereum e Bitcoin. Hyperledger Fabric é uma das plataformas para blockchains permissionadas mais populares atualmente[6] com recursos para a implantação de uma infraestrutura de rede privada entre organizações e desenvolvimento de aplicações no topo dessa rede. Nesse caso, os participantes da rede formam um consórcio e arcam com o custo da infraestrutura, buscando ganhos no desempenho em relação às redes blockchain públicas.

Nesse contexto, entender as características das infraestruturas computacionais para implantação e o funcionamento de uma rede blockchain é essencial para orientar o corpo técnico e executivo das organizações a planejarem uma possível adoção dessa tecnologia. Esses atores necessitam avaliar os modelos de rede pública ou permissionada e o problema em questão é entender requisitos não funcionais essenciais de cada modelo, em especial aspectos de custo e desempenho, assim como o compromisso entre ambos para planejar adequadamente as aplicações que funcionarão no topo da rede blockchain.

A maioria das propostas da literatura que lidam com essa questão focam em aplicações específicas para rede pública (LEAL; CHIS; GONZÁLEZ–VÉLEZ, 2020; ROUHANI; DETERS, 2017; ZHANG et al., 2020) ou rede permissionada (BALIGA et al., 2018; THAKKAR; NATHAN; VISWANATHAN, 2018; WANG; CHU, 2020; XU et al., 2021). Alguns trabalhos ainda focam na análise de uma aplicação típica para ambas as redes (MONRAT; SCHELEN; ANDERSSON, 2020; MALIK et al., 2019). Contudo, nenhuma dessas propostas busca identificar a infraestrutura, especificamente a arquitetura do nó da rede, considerando ao mesmo tempo os fatores desempenho e custo para uma organização participar de uma rede blockchain pública ou permissionada, que são representadas neste trabalho pelas plataformas Ethereum e Hyperledger Fabric.

Os experimentos conduzidos neste trabalho de conclusão mostram que um nó Ethereum com capacidade computacional básica (i.e., baixo custo) pode executar cargas intensas (200 transações por segundo – TPS) com alto nível de desempenho (latência média de 9 e 11 segundos) para redes blockchains públicas, apesar do alto consumo de processamento. Nesse caso, foi salientado que a arquitetura proposta leva ao melhor compromisso entre custo e desempenho, pois aumentos do poder computacional não levam à redução de latência. Por sua vez, na plataforma Hyperledger Fabric, o poder computacional de um nó é um fator importante dado que se espera alto desempenho (baixa latência de transação) em redes permissionadas. Nesse modelo de rede, dois tipos de nós são recomendados, con-

[6] https://www.ibm.com/topics/hyperledger

siderando o melhor compromisso entre custo e desempenho: um nó básico (baixo custo) para cargas menores que 140 TPS, e a partir dessa marca, um nó intermediário alcança latência média abaixo de 0,2 segundos.

1.1 Objetivos

Propor uma abordagem para analisar o custo e o benefício da infraestrutura computacional para aplicações em redes blockchain públicas e permissionadas, através da realização de análises comparativas entre diferentes contextos de implementação, visando contribuir para o avanço do conhecimento e auxiliar na tomada de decisões na adoção de tecnologias blockchain. Além disso, como contribuição adicional, o projeto visa desenvolver uma aplicação cliente para viabilizar as análises acima propostas.

1.1.1 Objetivos Específicos

- Analisar redes blockchain públicas e permissionadas conjuntamente;
- Considerar custo (monetário) e benefício (desempenho) simultaneamente para diferentes capacidades de infraestrutura;
- Desenvolver uma aplicação cliente para avaliar o desempenho e custo de redes blockchain.

2 Referencial Teórico

Para uma compreensão mais aprofundada deste trabalho, esta seção apresentará conceitos fundamentais que desempenham um papel crucial. Serão explicados os seguintes tópicos: Blockchain, Ethereum, Hyperledger Fabric e Análise de estatística, estabelecendo uma base sólida para a análise e implementação propostas.

2.1 Blockchain

Blockchains ganharam forte impulso na última década, após o lançamento inicial da cripto moeda Bitcoin (NAKAMOTO, 2008). Blockchain é uma tecnologia emergente que oferece suporte distribuído confiável e seguro para realização de transações entre participantes que não necessariamente têm confiança entre si e que estão dispersos em larga escala numa rede P2P (QUEIROZ, 2018). Uma blockchain pode ser definida como um livro-razão imutável para gravação de transações, mantidas dentro de uma rede distribuída de participantes desconfiados.

Essa tecnologia é o resultado de um conjunto de técnicas de computação distribuída, criptografia e até teoria dos jogos. A blockchain implementa uma máquina de estados replicada para a manutenção consistente de um estado global compartilhado por um conjunto de pares distribuídos numa rede P2P (GREVE et al., 2018). Blockchain é uma nova tecnologia que muda a forma como armazenamos e registramos dados e transações. É semelhante a um banco de dados tradicional, mas a ideia por trás de um blockchain é que podemos remover o participante central do sistema ou entes intermediários (NOFER et al., 2017).

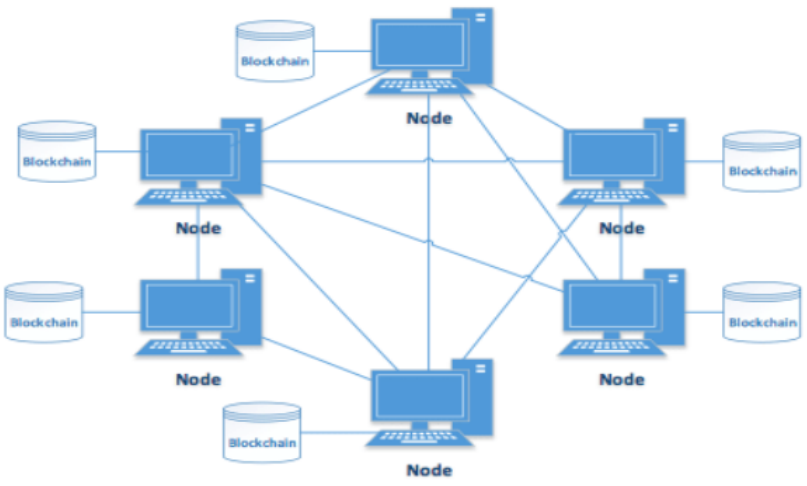


Figura 1 – Conexão entre os nós em uma rede Blockchain.

Conforme evidenciado na Figura 1, é possível observar que cada nó está interligado a uma cópia do blockchai, simbolizado por cilindros azuis. As linhas azuis escuras traçam as conexões entre os nós e entre estes e as cópias, proporcionando uma representação visual da natureza descentralizada da tecnologia blockchain. Nesse contexto, cada nó na rede desempenha o papel de manter uma cópia do blockchain, contribuindo para a distribuição e redundância característica dessa tecnológia.

Na origem da blockchain, está o protocolo do Bitcoin, proposto por Satoshi Nakamoto (NAKAMOTO, 2008), é de se esperar que essa tecnologia seria amplamente utilizada para transações entre criptomoedas e essa é uma das suas principais utilizações. O Bitcoin teve uma crescente adesão de usuários por ser o primeiro ativo digital totalmente descentralizado, ou seja, sem uma terceira parte de confiança, que é necessária para as transações financeiras.

Uso de blockchain em escopos diferentes de criptomoedas já e uma realidade (LAURENCE, 2019). Utilizada em diversas áreas da industria, a tecnologia Blockchain foi além do modelo de blockchain pública, aberta, como as criptomoedas para o modelo de blockchain privada ou federada. Nesse modelo os participantes tem acesso controlado à blockchain. Esse controle veio para atender melhor essas necessidades corporativas, que exigiam uma forma de autenticação dos seus vários membros que passam a ser participantes usuários da rede blockchain.

Em suma a tecnologia blockchain vem sendo utilizada de forma inovadora no desenvolvimento de aplicações e sistemas. Essa tecnologia tem como principais características ou propriedades: descentralização, disponibilidade, integridade, transparência, auditabilidade, imutabilidade e irrefutabilidade, anonimidade, no caso de redes públicas.

A blockchain é uma cadeia de blocos, onde cada bloco contém um conjunto de transações. Cada bloco é identificado por um hash único, que é como uma impressão digital. Além disso, cada bloco contém o hash do bloco anterior, criando uma ligação entre eles. Isso forma uma cadeia de blocos, daí o nome “blockchain”.

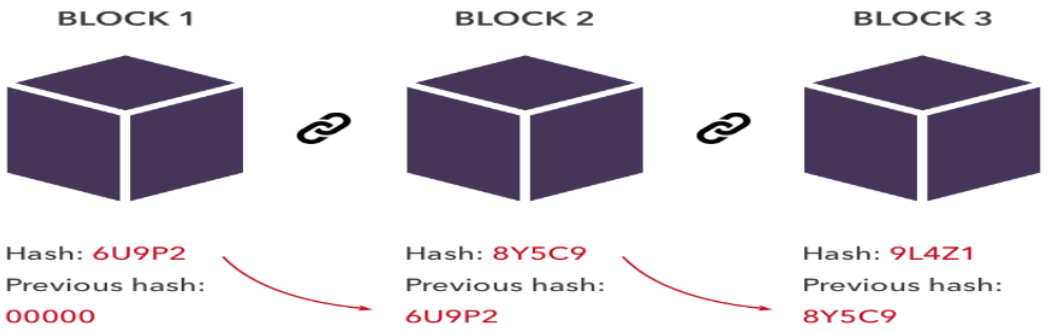


Figura 2 – Ligação segura entre blocos em uma Blockchain via hashes únicos.

Como podemos observar na Figura 2, quando um novo bloco é adicionado à blockchain, ele contém o hash do bloco anterior. Isso significa que se alguém tentasse alterar as

informações em um bloco, o hash desse bloco mudaria. Como cada bloco subsequente contém o hash do bloco anterior, a alteração de um bloco invalidaria toda a cadeia que vem depois dele. Isso torna a blockchain extremamente segura e resistente a fraudes ou alterações.

2.2 Ethereum

Ethereum (WOOD, 2014) toma emprestado muito do protocolo Bitcoin e de seu design Blockchain, mas o ajusta para oferecer suporte a aplicações além do dinheiro, nas quais o Ethereum melhora o conceito de scripts e metaprotocolos online (ZHANG et al., 2020). A arquitetura básica do Ethereum é composta por nós que executam softwares para verificar e manter as transações organizadas em blocos. Esses nós são computadores que executam os clientes Ethereum e que permitem que eles se conectem uns aos outros, contudo nesse projeto foi proposto o conceito de contratos inteligentes. Os contratos inteligentes expressam uma lógica de transações mais sofisticada, possibilitando a implementação de aplicações descentralizadas e autônomas, em diversos níveis e escopos (GREVE et al., 2018).

Especificamente, contratos inteligentes são código de aplicações (*bytecode*) com seus respectivos métodos e atributos podem ser registradas na blockchain. Transações invocando métodos públicos de um contrato podem modificar ou obter o estado dos seus atributos. A modificação desses atributos são registrados de forma imutável na blockchain ao longo do tempo para fins de auditoria. Mais importante, o último estado dos atributos de um contrato constitui o seu estado global, que pode ser armazenado em uma estrutura de dados otimizada à parte da blockchain para acesso rápido.

2.3 Hyperledger Fabric

Hyperledger Fabric é uma plataforma para soluções de livro razão (*ledger*) distribuído, sustentada por uma arquitetura modular que oferece altos graus de confidencialidade, resiliência, flexibilidade e escalabilidade (FABRIC, 2023). Ela foi projetada em 2015, pela Linux Foundation com o intuito de avançar as tecnologias de blockchain entre industrias. Contudo teve o apoio financeiro da IBM e atualmente a plataforma é mantida pela Fundação Hyperledger Foundation (FOUNDATION, 2023).

Hyperledger Fabric foi projetada para suportar implementações modulares de diferentes componentes, e acomodar complexidades que existem em todo o ecossistema econômico (FABRIC, 2023). Utiliza um sistema de contratos inteligentes por onde os participantes gerenciam as suas transações. Além disso cada um dos participantes de uma rede Hyperledger Fabric possui uma cópia do livro razão. O mecanismo usado para validar as

transações e criar blocos no Hyperledger Fabric é o *Practical Byzantine Fault Tolerance* (PBFT) (CASTRO; LISKOV et al., 1999).

Os contratos inteligentes na Hyperledger Fabric são denominados *chaincode* e são invocados por um aplicativo externo à blockchain quando esse aplicativo precisa interagir com o livro-razão (Hyperledger Fabric Project, 2023). Chaincodes podem ser implementados em *Go*, *Node.js* e Java. Essas três linguagens também são utilizadas para desenvolver as aplicações que invocam métodos do chaincode.

O Hyperledger Fabric não é uma rede blockchain aberta como a rede do Bitcoin, e sim uma rede privada ou permissionada. Para acessar uma rede Hyperledger é necessário que a maioria dos seus participantes aprove. O Hyperledger Fabric oferece suporte a sub-redes dentro da rede principal com o recurso denominado *canais*. Um canal Hyperledger Fabric pode estar associado a um ou mais chaincodes assim como uma ou mais organizações. Em geral, todos os membros (usuários) da organização têm acesso ao canal ao qual a organização está associada (Hyperledger Fabric Project, 2023).

O Fabric não estipula uma forma de consenso padrão, e são muitas as formas de compensar essa "falha" por assim dizer, é deixado a escolha livre para o tipo de consenso a ser usado. O Hyperledger Fabric foi projetado para permitir que os iniciantes da rede escolham um mecanismo de consenso que melhor represente os relacionamentos existentes entre os participantes (Hyperledger Fabric Project, 2023).

2.4 Análise de estatística

Nesta seção, abordaremos conceitos fundamentais das distribuições de probabilidades, Maximum Likelihood Estimation (MLE) e o teste de Kolmogorov-Smirnov. As distribuições de probabilidades são essenciais para descrever a incerteza em dados observados, enquanto a MLE é uma técnica-chave para estimar os parâmetros dessas distribuições. O teste de Kolmogorov-Smirnov, por sua vez, é uma ferramenta importante para validar a aderência entre os dados observados e as distribuições teóricas propostas. Esses conceitos são fundamentais para uma análise estatística precisa e robusta.

2.4.1 Distribuição de probabilidades

A compreensão das distribuições de probabilidade é importante tanto na mecânica estatística quanto na mecânica quântica, pois ajuda a descrever a incerteza em dados observados. Isso significa que as distribuições de probabilidade nos permitem entender a probabilidade de diferentes resultados em experimentos aleatórios. No século XIX, cientistas como Boltzmann, Maxwell e Gibbs introduziram o conceito de variáveis aleatórias na física. Essas variáveis são fundamentais na teoria da probabilidade e estatística, pois atribuem valores numéricos aos resultados de experimentos aleatórios. Em outras pala-

vras, elas nos ajudam a quantificar os resultados possíveis de um experimento, tornando mais fácil entender e analisar esses dados (NOVAES, 2022).

Quando a imagem (variação) de uma variável aleatória é finita ou infinita contável, a variável aleatória é chamada de variável aleatória discreta e sua distribuição pode ser descrita por uma função massa de probabilidade que atribui uma probabilidade a cada valor na imagem da variável (MAGALHÃES, 2006). Esse tipo de variável aleatória é caracterizado pela capacidade de assumir valores específicos e distintos.

Variáveis aleatórias contínuas, por outro lado, diferenciam-se das variáveis discretas ao abrangerem um conjunto infinito e não enumerável de valores em um intervalo contínuo de números reais. Isso significa que, ao contrário das variáveis aleatórias discretas, que podem assumir apenas valores distintos, as variáveis aleatórias contínuas podem assumir qualquer valor dentro de um intervalo específico. Uma variável aleatória contínua é aquela cujos possíveis resultados abrangem um intervalo contínuo de números reais. Em outras palavras, a variável pode assumir qualquer valor dentro de um intervalo especificado (MAGALHÃES, 2006).

2.4.2 Maximum Likelihood Estimation MLE

A Estimação de Máxima Verossimilhança (MLE) é um método amplamente utilizado para a estimação de parâmetros em modelos estatísticos (CHAN; WEN, 2015). Este método fornece uma abordagem sistemática para ajustar modelos estatísticos aos dados, sendo particularmente útil quando se lida com amostras grandes e complexas. A ideia principal por trás do MLE é escolher os parâmetros do modelo que maximizam a probabilidade (ou verossimilhança) dos dados observados (YAN, 2020).

Em outras palavras, o MLE escolhe os parâmetros que tornam os dados observados mais prováveis. Isso é feito maximizando uma função de verossimilhança, que é uma função dos parâmetros do modelo e dos dados observados (YAN, 2020). A eficácia do MLE reside na busca pelos valores dos parâmetros que tornam os dados observados mais prováveis sob o modelo proposto. A maximização da verossimilhança, assim, oferece uma estimativa robusta e eficiente dos parâmetros do modelo, facilitando a interpretação estatística e a tomada de decisões informadas.

A eficácia do MLE reside na busca pelos valores dos parâmetros que tornam os dados observados mais prováveis sob o modelo proposto. A maximização da verossimilhança, assim, oferece uma estimativa robusta e eficiente dos parâmetros do modelo, facilitando a interpretação estatística e a tomada de decisões informadas (BABU, 2022).

No entanto, é importante notar que a existência do MLE global nem sempre é garantida, mesmo em situações simples onde os dados vêm de misturas de Gaussianas (BABU, 2022). É essencial destacar que a busca pelo MLE pode ser sensível às características do modelo estatístico e à complexidade dos dados. A presença de múltiplos máximos locais ou a não convexidade da função de verossimilhança podem desafiar a obtenção do

MLE global. Portanto, é crucial verificar a validade da existência de tal estimador ao usar software ou algoritmos padrão para obter o MLE (BABU, 2022).

2.4.3 Kolmogorov-Smirnov

O teste de Kolmogorov-Smirnov (KS-Test) é uma técnica estatística utilizada para verificar se uma amostra de dados segue uma distribuição específica. Ele compara a distribuição acumulada empírica dos dados com a distribuição acumulada teórica esperada. O teste é baseado na diferença máxima entre uma distribuição cumulativa empírica e uma distribuição cumulativa hipotética (JR, 1951).

O artigo (CONG et al., 2020) aborda o teste Kolmogorov-Smirnov (KS) e sua aplicação em várias áreas, como detecção de anomalias, astronomia, segurança de banco de dados e sistemas de Inteligência Artificial (IA). No contexto da detecção de anomalias, o KS-test é aplicado para identificar desvios significativos na distribuição de dados, destacando observações que podem ser consideradas atípicas. Na astronomia, por exemplo, o teste é empregado para comparar distribuições de características estelares observadas com as expectativas teóricas.

A estatística Kolmogorov-Smirnov quantifica uma distância entre a função de distribuição empírica da amostra e a função de distribuição cumulativa da distribuição de referência, ou entre as funções de distribuição empíricas de duas amostras (ARTAYA, 2019). Essa métrica é expressa pelo valor da estatística de teste KS (D), que representa o maior desvio vertical absoluto entre as distribuições acumuladas. Essa medida fornece uma maneira eficaz de avaliar a aderência dos dados a uma distribuição teórica, sendo particularmente útil na identificação de desvios significativos em diferentes partes das distribuições.

Apesar de numerosos métodos sugeridos, o teste KS é, de longe, o teste de ajuste de bondade (GOF) mais popular usado na prática (EIGER; NADLER; SPIEGELMAN, 2013). Sua popularidade advém da sua simplicidade de aplicação e interpretação, bem como da capacidade de lidar com uma ampla variedade de distribuições. O teste KS não requer a especificação prévia de parâmetros da distribuição, tornando-o flexível e fácil de implementar em diferentes contextos.

3 Trabalhos Relacionados

Neste Capítulo, exploraremos estudos que avaliam custos e desempenho em plataformas Blockchain, buscando preencher uma lacuna na literatura ao oferecer uma síntese de trabalhos significativos sobre o assunto, proporcionando uma visão atualizada.

Existem disponíveis na literatura trabalhos que abordam a avaliação de custos e desempenho de plataformas Blockchains, porém não realizando comparação com infraestruturas e plataformas diferentes. Grande parte desses estudos apresentam avaliações específicas para uma plataforma. Dessa maneira, os parâmetros e requisitos considerados impossibilitam a escolha da melhor plataforma e infraestrutura necessárias para as quais as aplicações serão projetadas. Portanto, esta seção sintetiza alguns trabalhos relacionados à avaliação de custos e desempenho de plataformas Blockchains.

O trabalho desenvolvido por Rimba et al. (2020) investigou a questão do custo monetário de utilizar uma plataforma blockchain em comparação com uma infraestrutura de armazenamento em nuvem. Por meio de modelos de custo para processos de negócios, os autores compararam os custos na plataforma Ethereum e Amazons Simple Workflow Service (SWF). Os resultados apontaram uma grande variação de custo entre as duas soluções. O custo do blockchain Ethereum é, pelo menos, o dobro dos serviços tradicionais de nuvem fornecidos pelo Amazon SWF. Nosso trabalho se diferencia ao apresentar um modelo de custo para comparação da infraestrutura necessária para manter o provimento da plataforma blockchain, sendo ela pública ou permissionada.

Baliga et al. (2018), Thakkar, Nathan e Viswanathan (2018), Wang e Chu (2020) analisaram o desempenho da plataforma Hyperledger Fabric. A abordagem de Baliga et al. (2018) utilizou a ferramenta Hyperledger Caliper sob diferentes configurações para avaliar a latência e a taxa de transferência do Hyperledger Fabric. Avaliaram também o desempenho variando o número de *chaincodes*, *channels* e *peers*. Concluíram que a taxa de transferência é sensível às configurações e que a latência é significativamente afetada pelo tamanho da carga utilizada. Thakkar, Nathan e Viswanathan (2018) testou duas abordagens para avaliação de desempenho, otimização de cache e configuração de políticas de endosso. Como contribuição, os autores descreveram orientações sobre a configuração de parâmetros da rede e também os principais gargalos de desempenho. Em Wang e Chu (2020), os autores caracterizaram o desempenho das três fases de uma transação (endosso, execução e validação), sendo que a fase de execução mostrou boa escalabilidade de desempenho, ao passo que a validação obteve desempenho pior devido a carga computacional mais intensa sob o nó nessa fase. Contudo, os autores concluíram que variações na política de endosso, i.e., número mínimo de pares para aprovar uma transação, foi o principal fator na variação do desempenho. Seguindo recomendações desses trabalhos, utilizamos a política de endosso padrão da plataforma: 50% mais um pares.

Xu et al. (2021) propôs um modelo inovador para calcular com precisão a latência em diversas configurações de rede, identificando parâmetros cruciais por meio de validação experimental. O estudo concentra-se na análise de desempenho da latência no Hyperledger Fabric, uma plataforma de blockchain empresarial com permissão. Utilizando um modelo analítico, o artigo avalia a latência, destacando o tempo necessário para enviar propostas de transação e confirmar transações no ledger. Além disso, a pesquisa aborda a importância da análise teórica da latência em comparação com estudos empíricos, ressaltando a necessidade de compreender as nuances teóricas junto com avaliações práticas em diversas configurações de sistema e plataformas de hardware.

Os artigos Leal, Chis e González–Vélez (2020), Rouhani e Deters (2017), Zhang et al. (2020) fornecem avaliação de desempenho de redes blockchain privadas baseadas na plataforma blockchain Ethereum de código aberto. Leal, Chis e González–Vélez (2020) avaliam o desempenho da rede utilizando um conjunto de dados para definir a configuração ideal. Eles utilizaram diferentes custos, algoritmos de consenso, e número de nós de rede para determinar a configuração. Como contribuição é fornecida uma forma para encontrar a configuração ideal para um determinado número de transações exigidas por determinado caso de uso. O trabalho de Rouhani e Deters (2017) mostrou que o desempenho da rede Ethereum depende, além da configuração da rede, da implementação do cliente utilizada. O estudo mostra que o cliente Parity obteve desempenho significativamente melhor do que o cliente Geth. Em Choi e Hong (2021), os autores utilizaram o Hyperledger Caliper para avaliar a rede Ethereum. Os resultados mostram que o desempenho das transações pode diferir de acordo com seu conteúdo e configuração da rede.

Existem estudos de análise de desempenho Blockchain que avaliam e comparam as plataformas Hyperledger Fabric e Ethereum. E.g., em Monrat, Schelen e Andersson (2020) é realizada uma análise de desempenho e escalabilidade, variando as cargas de trabalho, das plataformas Ethereum, Quorum, Corda e Hyperledger Fabric. A conclusão geral do trabalho é que o Hyperledger Fabric tem um desempenho superior às demais plataformas porque atinge o consenso de forma mais eficiente. Em Malik et al. (2019) é realizada uma comparação do desempenho das plataformas Ethereum e Hyperledger Fabric utilizando uma aplicação de comércio de energia e Hyperledger Caliper. A conclusão é que o Ethereum fornece a melhor solução para a aplicação em pequena escala, mas, o Hyperledger Fabric pode ser mais adequado para aplicações de grande escala. Contudo, ambas as propostas não tratam dos aspectos arquiteturais do nó de uma rede blockchain publica ou permissionada que impactam no seu custo e desempenho, e portanto são aspectos relevantes para uma organização participar de um desses modelos de rede.

O artigo Melo et al. (2021) avalia modelos para determinar a capacidade de infraestruturas de computação em nuvem para aplicativos baseados em blockchain Ethereum, incluindo os custos em ambientes públicos e privados. Utiliza a ferramenta Mercury para avaliação e destaca a importância de interpretação visual e textual dos dados. Compara

os custos entre nuvens públicas e privadas para serviços de blockchain, visando fornecer informações sobre a viabilidade e custo-benefício dessa tecnologia na nuvem.

Tabela 1 – Trabalhos Relacionados.

Autor(es)	**Foco**	**Hyperledger Fabric**	**Ethereum**	**Custo**	**Desempenho**
(RIMBA et al., 2020)	Custo vs. nuvem em plataformas blockchain.	Não	Sim	Sim	Não
(BALIGA et al., 2018)	Desempenho do Hyperledger Fabric.	Sim	Não	Não	Sim
(THAKKAR; NATHAN; VISWANATHAN, 2018)	Otimização do Hyperledger Fabric.	Sim	Não	Não	Sim
(WANG; CHU, 2020)	Fases da transação no Hyperledger Fabric.	Sim	Não	Não	Sim
(XU et al., 2021)	Análise de desempenho da latência no Hyperledger Fabric.	Sim	Não	Não	Sim
(LEAL; CHIS; GONZÁLEZ–VÉLEZ, 2020)	Desempenho da rede Ethereum.	Não	Sim	Sim	Sim
(ROUHANI; DETERS, 2017)	Desempenho da rede Ethereum (Parity).	Não	Sim	Não	Sim
(MONRAT; SCHELEN; ANDERSSON, 2020)	Desempenho de Ethereum, Quorum, Corda e HL Fabric.	Sim	Sim	Não	Sim
(MALIK et al., 2019)	Comparação Ethereum e Hyperledger Fabric.	Sim	Sim	Não	Sim
(MELO et al., 2021)	Provisionamento de aplicativos Ethereum em blockchain privado: disponibilidade e custos.	Não	Sim	Sim	Não
Este Trabalho	Avaliação de custo e desempenho em Ethereum e Hyperledger Fabric.	Sim	Sim	Sim	Sim

A Tabela 1 sintetiza uma revisão abrangente de trabalhos relacionados à avaliação de custos e desempenho em plataformas blockchain, com foco específico nas populares Ethereum e Hyperledger Fabric. Os estudos abordam temas diversos, desde a comparação do custo de utilização de blockchains em relação a infraestruturas em nuvem até a análise detalhada do desempenho de diferentes componentes em redes permissionadas. Além disso, as investigações se estendem ao desempenho da rede Ethereum, considerando variações em configurações e clientes utilizados. As análises de desempenho do Hyperledger Fabric também são abordadas, com enfoque em otimizações, políticas de endosso e escalabilidade. Importante ressaltar que, apesar da riqueza desses estudos, uma lacuna persiste na literatura, especificamente na análise da infraestrutura do nó da rede, considerando simultaneamente os fatores de desempenho e custo para organizações participarem de redes blockchain públicas ou permissionadas. Este trabalho busca preencher essa lacuna ao apresentar uma arquitetura de nó e uma avaliação experimental detalhada, permitindo

uma análise holística das plataformas Ethereum e Hyperledger Fabric em termos de custo e desempenho.

4 Avaliação de Custo e Desempenho

Este capítulo oferece uma análise sobre a arquitetura proposta para avaliação e conduz uma avaliação experimental detalhada do custo e desempenho dos nós envolvidos. Dividido em duas seções principais, o capítulo começa por apresentar a arquitetura geral concebida para a avaliação em pares de redes blockchain, destacando sua adaptação e aplicabilidade em plataformas específicas, como Ethereum e Hyperledger Fabric. Em seguida, adentra-se na avaliação experimental, onde são minuciosamente analisados os custos e desempenhos dos nós, fornecendo informações para desenvolvedores e organizações que buscam compreender o equilíbrio entre esses fatores para a participação efetiva nessas redes blockcains.

4.1 Arquitetura Proposta para Avaliação

Esta seção apresenta a arquitetura geral considerada para avaliação em um par de redes blockchain e a sua utilização para plataformas específicas. Dessa forma é possível equiparar os recursos utilizados de diferentes plataformas, afim de comparar os custos essenciais para sua operação. Inicialmente é introduzida a descrição da arquitetura geral e, a seguir, é demonstrado como ela se aplica às plataformas Ethereum e Hyperledger Fabric. O objetivo dessa proposta é unificar diferentes arquiteturas de rede blockchain, e.g., redes públicas e permissionadas, para analisar requisitos de custo e desempenho de se participar dessas redes em termos de infraestrutura básica (i.e., um nó da rede). Já com uma arquitetura unificada, o trabalho focou-se nos aspectos essenciais de blockchain, buscando reduzir a complexidade dessas análises e, ao mesmo tempo, mantê-las realistas em termos de custo e benefícios.

A arquitetura geral para um participante da rede blockchain pública ou permissionada é mostrada na Figura 3[1]. Nessa figura, os componentes verticais em linhas contínuas representam nós da rede mantidos por uma entidade (pessoa ou organização) que participa da rede blockchain. O nó é um computador físico ou uma máquina virtual em serviços de computação em nuvem administrado pela entidade. Por sua vez, os componentes horizontais em linhas tracejadas representam os protocolos de *consenso* e par-a-par (*P2P*), elementos básicos para o funcionamento de uma rede blockchain que devem estar contidos em cada nó da rede. Nas principais implementações de blockchains atuais, esses

[1] Produzida sob a motivação em diferentes conteúdos técnicos como: https://ethereum.org/en/developers/docs/nodes-and-clients e https://kctheservant.medium.com/multi-host-deployment-for-first-network-hyperledger-fabric-v2-273b794ff3d

dois elementos podem ser modularizados como dois processos diferentes, aproveitando as tecnologias de isolamento de recursos de computação leve como contêineres.

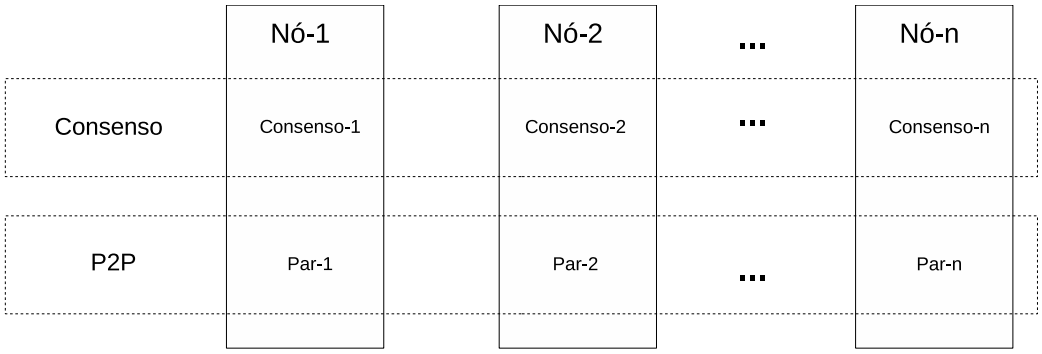


Figura 3 – A arquitetura geral para um nó da rede blockchain pública ou permissionada.

O protocolo de consenso define as regras para a escolha do nó líder da vez, i.e., periodicamente escolhido, responsável pela construção do próximo bloco de transações a ser replicado para os demais nós da rede. Existem diferentes protocolos de consenso como, por exemplo, Prova de Trabalho e Prova de Participação. Esses protocolos de consenso são adotados em redes blockchain públicas como *Bitcoin e Ethereum*. *Raft* e *Kafka* são serviços de ordenação de transações que funcionam como protocolos de consenso em redes blockchain permissionadas como Hyperledger Fabric (GREVE et al., 2018).

O protocolo *P2P* é responsável pela comunicação entre os nós, estendendo-se também às etapas de processamento de transações pelos nós da rede. A comunicação geralmente segue protocolos *Gossip*, onde nós obtém uma lista limitada de parceiros e estabelecem conexões entre eles formando uma rede sobreposta para a difusão de blocos de transações. O processamento desses blocos varia de acordo a plataforma sendo que Bitcoin e Ethereum adotam a estratégia ordenar-executar blocos, ao passo que Hyperledger Fabric adota a estratégia executar-ordenar-validar blocos (ANDROULAKI; et al., 2018).

4.1.1 Ethereum

Ethereum é atualmente a segunda maior rede pública de blockchain do mundo em arrecadação de fundos, portanto uma representante importante desse modelo de rede[2]. A arquitetura básica do Ethereum é composta por nós que executam softwares para verificar e manter as transações organizadas em blocos. Esses nós são computadores que executam os clientes Ethereum e que permitem que eles se conectem uns aos outros. Os clientes Ethereum são responsáveis por verificar se os dados inseridos ou solicitados por meio de transações cumprem as regras impostas pelo protocolo da rede. Existem dois tipos de clientes que são estabelecidos nas camadas de execução e consenso da rede. Esses clientes

[2] Mais informações podem ser encontradas em *https://ethereum.org/en/developers/docs/*.

são interdependentes e devem ser executados de maneira conjunta, podendo ser em hosts separados, para fornecer acesso à rede.

Primeiramente, conforme ilustrado pela Figura 4, um nó da rede recebe e executa as transações enviadas para a rede por um cliente (1), por meio da camada de execução, mantendo o banco de dados que representa o estado atual da rede. A camada de consenso implementa o algoritmo de consenso para validação de dados de acordo com o estado da rede mantido pelos clientes em execução. No algoritmo de consenso de prova de participação, ou em inglês, Proof of Stake – POS, se um nó da rede quiser se tornar um validador, ele primeiro deve enviar uma taxa de validador (2) e quando a transação for confirmada, ele poderá apostar algumas moedas para competir com outros validadores (3). Por sua vez, cada nó é responsável por transmitir as transações que recebe dos clientes aos outros nós (4). Quando uma quantidade suficiente de transações é recebida, os validadores elegem um líder com o máximo de moedas apostadas. O líder eleito então cria um bloco e o transmite para a rede (5) onde cada nó valida o bloco, executa todas as transações do bloco e adiciona o bloco na cadeia (6). O bloco também possui uma transação de recompensa especial, sendo que o líder da rodada recebe como recompensa as taxas de transação inferidas nas transações presentes no bloco.

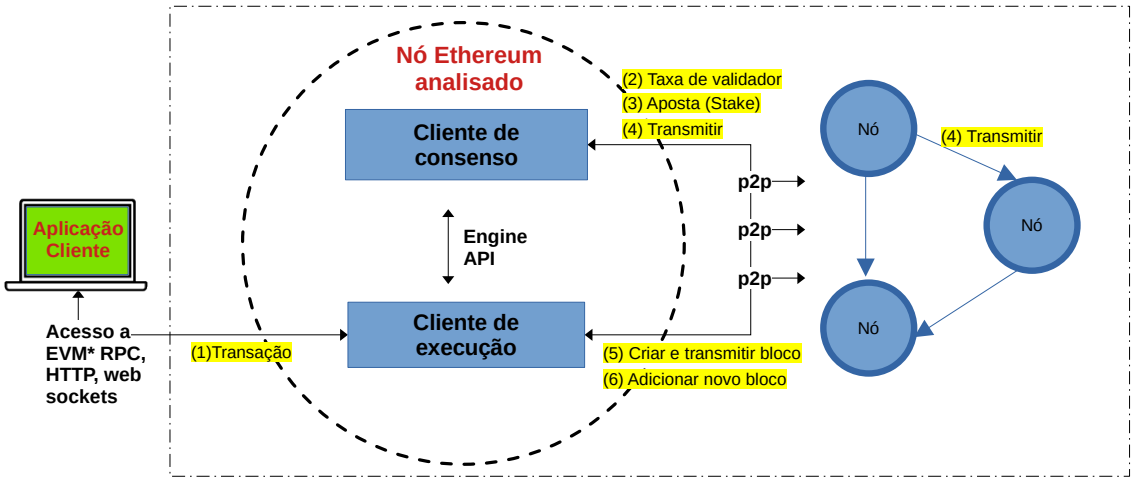


Figura 4 – Modelo de rede pública Ethereum

Manter a propriedade de nós da rede para execução e consenso, em uma rede pública, oferece os benefícios da independência de terceiros, ao fornecer acesso à rede por aplicativos, além de prover a descentralização esperada da rede. O fato da mudança do uso do algoritmo de consenso para o PoS resultou em uma redução significativa do consumo de recursos em comparação ao algoritmo anterior PoW e, portanto, um menor custo para manter um nó da rede. Desta forma, a viabilidade para que uma aplicação tenha acesso aos dados disponíveis na Blockchain, depende estritamente destes nós, que recebem as solicitações de transações, por meio da camada de execução e as submetem ao consenso da rede.

4.1.2 Hyperledger Fabric

A plataforma para redes permissionadas Hyperledger Fabric é uma das mais populares atualmente. Ela é um grande projeto de código fonte aberto envolvendo mais de 35 organizações e 200 desenvolvedores[3]. A rede blockchain da plataforma Hyperledger Fabric usa a estratégia executar-ordenar-validar para processar blocos de transações.

Assim como no Ethereum, podemos organizar essa estratégia em dois elementos essenciais para a arquitetura, que são os componentes P2P e consenso. Contudo, o componente P2P neste contexto tem atribuições extras. Em linhas gerais, os pares primeiramente executam uma transação, i.e., simulam seu funcionamento e proveem o endosso da transação para a aplicação cliente, e posteriormente os pares validam as transações, as mantendo na estrutura de dados encadeada da blockchain. Por sua vez, os ordenadores participam da estratégia após a etapa de execução para realizar o consenso, i.e., determinar o líder que ordena as transações em um novo bloco, e posteriormente envia o bloco aos pares para a etapa de validação.

A Figura 5 ilustra o fluxo de uma transação no Hyperledger Fabric e componentes P2P e consenso representados por pares e serviço de ordenação respectivamente. Inicialmente, um par recebe a proposta de transação da aplicação cliente (1). O par então simula a execução da transação invocando o contrato inteligente que a corresponde e envia uma mensagem de endosso ou sua negativa para a aplicação cliente (2). A aplicação aguarda endossos de outros pares, conforme a quantidade configurada na rede, e então envia a transação para o serviço de ordenação (3). A seguir, esse serviço recolhe transações da rede até alcançar o tempo (*timeout*) ou quantidade de transações limite para gerar um novo bloco. Os blocos são então encaminhados para os pares da rede realizarem a validação (4), que consiste no encadeamento do bloco à blockchain e atualização do seu estado global para consultas rápidas, e.g., variáveis dos contratos inteligentes ou saldos de contas.

Há alguns pontos importantes para observar na arquitetura particular do Hyperledger Fabric que permite também representá-lo pela arquitetura geral da Figura 3. Primeiro, execução e validação (passos 2 e 4) são realizadas pelos pares da rede, i.e., o componente P2P, ainda que estejam em etapas diferentes no fluxo da transação. Por sua vez, o serviço de ordenação é constituído por uma coleção de ordenadores – os serviços Raft e Kafka adotados no Hyperledger Fabric requerem ao menos três ordenadores – que determinam o líder da vez para a geração do novo bloco, i.e., o componente consenso. Logo, um nó participante de uma rede blockchain permissionada Hyperledger Fabric pode conter os componentes par e consenso como mostrado na arquitetura geral (Figura 3) em concordância à arquitetura particular e fases de transações dessa rede (Figura 5).

[3] Mais informações podem ser encontradas em *https://hyperledger-fabric.readthedocs.io*.

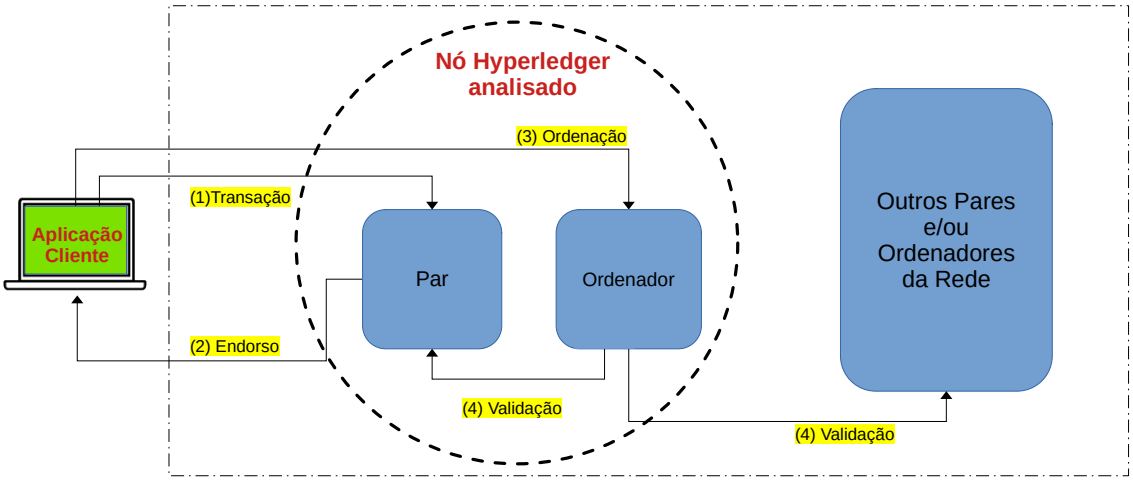


Figura 5 – Componentes e fluxo de transações na plataforma Hyperledger Fabric.

4.2 Avaliação Experimental

A seção anterior apresentou uma proposta de arquitetura geral para um nó participante de uma rede blockchain e a sua aplicação à rede pública Ethereum e à rede permissionada Hyperledger Fabric. Esta seção, avalia experimentalmente o custo e desempenho desse nó de forma conjunta, buscando analisar o compromisso entre esses dois importantes fatores para desenvolvedores e organizações que necessitam dessa informação para planejarem a participação nesses dois tipos de redes blockchain. A avaliação busca responder as seguintes perguntas: *A arquitetura proposta (unificação de dois componentes em um único nó) consome rapidamente recursos computacionais do nó levando a redução de desempenho? A arquitetura requer um nó de alto custo para executar transações com níveis razoáveis de desempenho?*

4.2.1 Metodologia

Este projeto produziu ambientes experimentais Ethereum e Hyperledger Fabric. O ambiente Ethereum foi construído a partir do software cliente *Geth* versão 1.10.26, que é a implementação oficial do protocolo Ethereum. A tecnologia *Docker*[4] foi utilizada para criar dois *contêineres*: um *contêiner* configurado como componente P2P utilizando o cliente *Geth* para receber, retransmitir solicitações de transações e manter o livro razão, e outro *contêiner* configurado como componente de consenso, i.e., respectivamente, os componentes P2P e consenso definidos na arquitetura geral. Os dois contêineres foram iniciados em um mesmo nó, i.e., uma máquina virtual, ambos configurados com interfaces de redes para comunicarem entre si via as portas TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*) do protocolo Ethereum. O nó recebe requisições de transações da aplicação através do componente P2P e as repassa ao componente de consenso, que é responsável pela mineração das transações e geração dos blocos que encadeiam e

[4] https://www.docker.com/resources/what-container

armazenam essas transações. Contudo, o componente consenso não está conectado à rede principal pública Ethereum e os blocos gerados contêm apenas transações da aplicação de nosso ambiente experimental. Para simular uma vazão semelhante à rede principal Ethereum, o tempo de criação de bloco para o protocolo de consenso foi estipulado em 15 segundos.

O ambiente Hyperledger Fabric foi criado a partir das imagens de *containers Dockers* contendo o software cliente oficial dessa plataforma versão 2.2. A fundação Hyperledger disponibiliza um *contêiner* para atuar como ordenador baseado no protocolo RAFT e outro *contêiner* para atuar como par da rede P2P no endosso e validação de transações. Assim, foi iniciado um *contêiner* par e um *contêiner* ordenador comunicando-se entre si via interfaces de redes e portas TCP e UDP do protocolo Hyperledger Fabric em um mesmo nó, seguindo a arquitetura geral proposta. O par recebe requisições de transação da aplicação e segue o fluxo mostrado na Figura 5 para registrá-la na blockchain. Diferente do ambiente Ethereum, foram criados três nós para o funcionamento de uma rede blockchain permissionada. Isso porque a plataforma Hyperledger requer ao menos três ordenadores para construírem blocos de transações, implementando o protocolo tolerante a falhas de *crash* em até dois ordenadores. Uma rede sobreposta utilizando o orquestrador de containers *Docker Swarm* foi utilizada para a comunicação entre os *contêineres* dos três nós, i.e., três ordenadores e três pares.

A ferramenta de aferição *Caliper* (CALIPER, 2019) foi utilizada para atuar como aplicação cliente gerando cargas com emissão de transações para os dois ambientes construídos. As cargas sintéticas geradas emulam uma aplicação blockchain típica com foco na emissão de transações, i.e., inserção de registros, que é usualmente a operação com maior uso de recursos computacionais e latências em blockchains como já observado em trabalhos anteriores (SPENGLER; SOUZA, 2021). Assim, a carga de trabalho submetida em ambas as plataformas representa um conjunto de registros emitidos por segundo (transações por segundo – tps), de forma fixa em um dado período de tempo aqui chamado de rodada. O valor em tps das cargas de trabalho foi incrementado gradativamente até atingir 200 tps, i.e., uma carga de alta intensidade. Para cada carga realizamos 10 rodadas de 60 segundos cada. Os códigos utilizados para os experimentos desta seção estão disponíveis no repositório *Blockchain performance* [5].

Três tipos de recursos computacionais foram utilizados para executar os experimentos nos ambientes que representam as redes blockchain pública e permissionada. Esses tipos são máquinas virtuais (VMs) do serviço *Amazon Elastic Cloud Computing* (EC2) para compor os nós de cada rede. Nos experimentos o poder computacional desses nós foi incrementado gradualmente para analisar o desempenho em função do aumento de carga. Nesse sentido, foram utilizadas as VMs T2 do tipo *small, medium* e *xlarge*, cujas respectivas especificações são apresentadas na Tabela 2.

[5] https://github.com/LABPAAD/blockchain_performance

	Small	**Medium**	**xLarge**
vCPUs	1	2	4
Memória (GB)	2	4	16
Custo/hora (USD)	0,0230	0,0464	0,1856

Tabela 2 – Especificações dos nós que compõem cada tipo de infraestrutura: família AWS T2, processador Intel Xeon 3.0-3.3 GHz e disco SSD de 100 GB.

Foram coletadas para cada carga de trabalho executada as métricas de latência da transação, além do uso dos recursos processamento (CPU), memória, disco e rede para os nós da rede. O Caliper registra a latência e a confirmação (sucesso ou falha) para cada transação e um programa, que foi desenvolvido na linguagem Python e a biblioteca *Psutil 5.9.0*, coleta os dados sobre os recursos monitorados em cada nó dos ambientes.

Intuitivamente, o crescimento da latência de transações pode estar associado ao aumento do consumo de recursos computacionais. Nesse sentido, foram examinados o consumo de recursos em nós de diferentes tipos com a finalidade de observar quais recursos são mais requisitados, preliminarmente ao início dos experimentos.

Foi observado que CPU é o recurso que tem o uso mais impactado com os aumentos de carga (transações por segundo). O uso de memória permanece estável em torno de 1GB, ao passo que o uso da rede (entrada e saída) e disco crescem com o aumento de carga mas ficam distantes das capacidades máximas que são iguais para diferentes tipos de nós, i.e., 1 Gbps de comunicação entre os componentes (containers) e 1 Gbps de leitura/escrita em discos SSD. Portanto, o foco das análises a seguir é no uso de CPU.

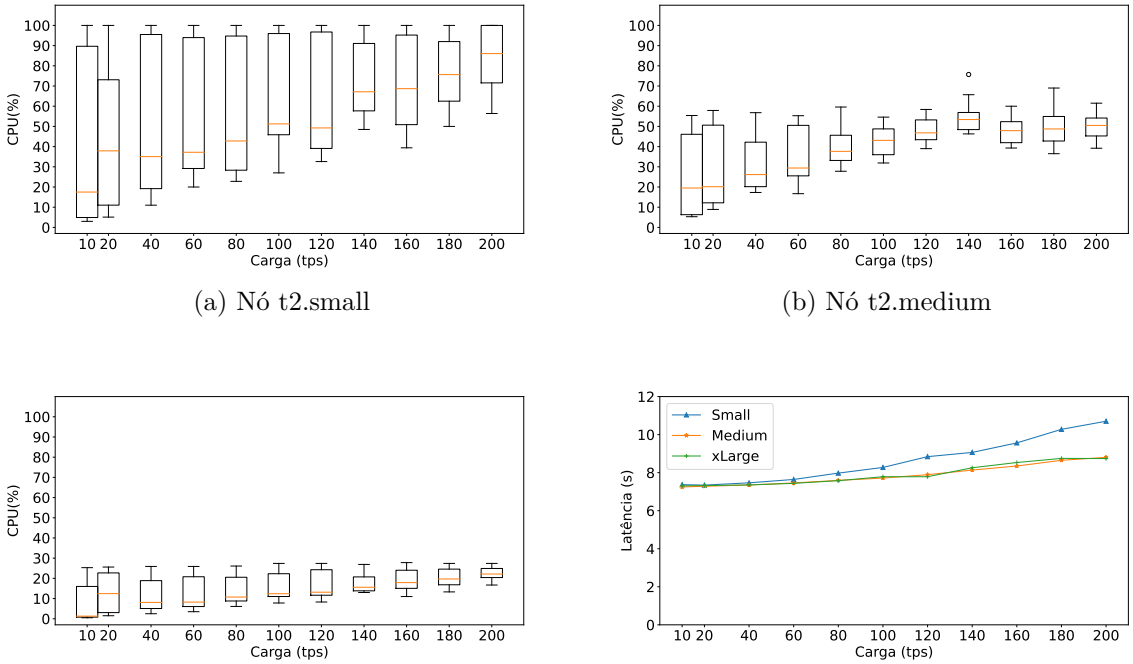
4.2.2 Avaliação de Desempenho

Esta seção busca responder a primeira pergunta de pesquisa: *"A arquitetura proposta consome rapidamente recursos computacionais do nó levando a redução de desempenho?"* Para isso foram relacionados o uso de processamento com a latência média de transações considerando o crescimento da carga e aumento do poder de processamento da CPU.

As Figuras 6 e 7 mostram a variação de uso de CPU e a latência em função da carga de trabalho em transações por segundo (tps) para as medições observadas nos três tipos de nós nas plataformas Ethereum e Hyperledger respectivamente. As três primeiras figuras apresentam *boxplots* para sumarizar a distribuição dos usos de CPU nos três tipos de nós da seguinte forma: o retângulo central se expande entre o primeiro e terceiro quartil, o segmento interior é a mediana, enquanto os indicadores abaixo e acima do retângulo representam o 10º e 90º percentis. Por sua vez, a quarta figura representa as curvas da latência média de uma transação para os três tipos de nós.

A Figura 6 apresenta resultados observados para a plataforma Ethereum. De modo geral, nota-se que a variação do uso de CPU cresce com o aumento da carga de trabalho, que pode ser observado pelas marcas da mediana. O tipo *small* sofre a maior variação de uso CPU visto pelas expansões consecutivas dos boxplots entre o 10º e 90º percentis (i.e.,

80% das medições) e uma parcela relevante das medições (10% das medições) tiveram 100% do uso de CPU como indica a marca do 90º percentil a partir de 20 tps. Pode-se observar na Figura 6-d que o alto uso de CPU aumenta a latência do nó tipo *small* em até 3,4 segundos em relação aos outros. Valor este que consideramos um impacto irrelevante na latência para uma rede pública, um vez que os tempos gastos por transação pela rede principal é atualmente em torno de 12,7 segundos. Logo, a arquitetura proposta consome rapidamente recursos computacionais do nó de menor capacidade, mas não levando a redução significativa de desempenho, i.e., latência da transação.



(a) Nó t2.small

(b) Nó t2.medium

(c) Nó t2.xlarge

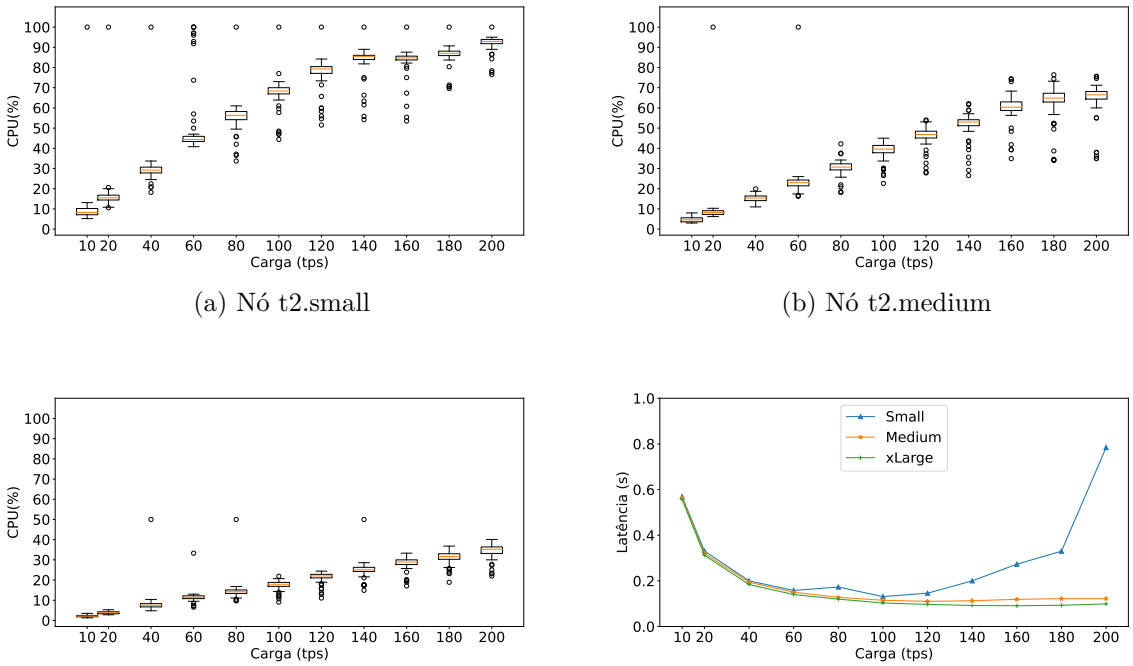
(d) Latência

Figura 6 – Uso de CPU e latência em nó da plataforma Ethereum.

Agora discutimos os resultados observados para nós Hyperledger Fabric, mostrados na Figura 7.[6] Observa-se pouca variação no uso de CPU, dado a menor expansão dos boxplots, mas esse uso cresce com a carga, semelhante ao Ethereum. Usos expressivos de CPU, i.e., 80% de uso em mais de 90% das amostras, ocorrem no nó tipo *small* em cargas intensas (acima 120 tps). É importante então analisar o impacto desse uso expressivo no desempenho do nó. A Figura 7-d mostra que a latência para nós Hyperledger Fabric é baixa (menor que 1 segundo) em comparação aos nós Etherem, como esperado para um rede blockchain permissionada. Contudo, a capacidade do tipo de nó tem impacto relevante na latência para cargas intensas. Note que o nó tipo *small* tem latência até 8 vezes maior que os outros tipos em carga de 200 tps. Interessante observar ainda a redução da latência à medida que a carga aumenta, que não permanece no nó *small* devido o consumo expressivo de CPU. Portanto, o poder computacional de um nó é um fator

[6] A rede utilizou três nós dessa plataforma e que tiveram desempenhos qualitativamente similares e por questão de espaço mostramos resultados de apenas um dos nós.

importante na plataforma Hyperledger Fabric, que tem como característica baixa latência de transação, e um nó básico (e.g., tipo *aws t2.small*) pode ter redução significativa de desempenho em cargas intensas com a arquitetura proposta.



(a) Nó t2.small

(b) Nó t2.medium

(c) Nó t2.xlarge

(d) Latência

Figura 7 – Uso de CPU e latência em rede permissionada Hyperledger Fabric.

4.2.3 Compromisso entre Custo e Desempenho

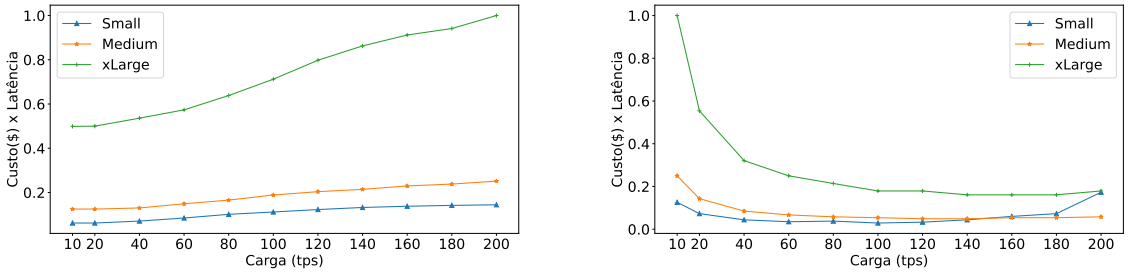
Esta seção busca responder a segunda pergunta colocada: *A arquitetura requer um nó de alto custo para executar transações com níveis razoáveis de desempenho?* Para isso foi aplicado as latências observadas na seção anterior um peso proporcional ao custo do tipo do nó por hora como é usual na precificação de recursos computacionais, o que nos permite analisar o compromisso entre custo e desempenho.

A Figura 8 mostra esse compromisso para os três tipos de nó em função da carga (tps). Cada tipo é representado por uma curva obtida via o produto *custo* × *latência* normalizado, onde o custo se refere aos preços praticados pela Amazon mostrados na Tabela 2. Assim, o tipo de nó com o menor valor desse produto é recomendado nessa análise em função da carga de trabalho. Nós da plataforma Ethereum mostrados na Figura 8-a, notavelmente, tem o melhor compromisso entre custo e desempenho com o tipo *small*, pois não há diferenças relevantes para as latências observadas entre esse nó e os demais, mesmo com o aumento da carga, logo o fator custo é dominante na escolha do tipo de nó. Por sua vez, dois tipos de nós são recomendados para a plataforma Hyperledger Fabric em função da carga, considerando o melhor compromisso entre custo e desempenho como mostra a Figura 8-b. O tipo *small* é recomendado para cargas menores que 140 tps, e a

partir dessa marca, o tipo *medium* é mais indicado, i.e., os ganhos com latência do nó *medium* compensam o seu custo, que é duas vezes superior ao nó tipo *small*.

Alguns entendimentos interessantes sobre negócios envolvendo blockchain podem ser obtidos com essa análise. Primeiro, serviços de infraestrutura em blockchains públicas como Ethereum se tornam viáveis com o baixo custo de um nó nessa rede, em especial após a adoção do consenso Prova de Participação. Tomando como exemplo a Infura, que é o serviço mais popular atualmente, um único nó *small* pode atender com uma carga de 200 tps até 86 pagadores do plano básico[7], gerando uma receita expressiva (cerca de 250 vezes maior) em relação ao custo mensal desse nó.

Sobre blockchains permissionadas com Hyperledger Fabric, existem vários casos de implantação dessa plataforma em redes corporativas[8], mas não há ainda a concepção de uma arquitetura de nó padrão para eles. De fato, a documentação oficial apenas ilustra o uso da plataforma de forma simplista e centralizada com um nó ordenador e vários nós pares, havendo, porém, a orientação à construção de redes personalizadas pelas corporações. A arquitetura avaliada neste trabalho unifica os componentes ordenador e par em um único nó completo, possibilitando estimar um teto de custo (i.e., uma estimativa conservadora) para uma organização participar numa rede blockchain permissionada. Em resposta à pergunta dessa seção, é elucidado que essa arquitetura requer um nó com capacidade intermediária (e.g., *aws t2.medium*) para executar cargas intensivas com alto desempenho.



(a) Ethereum (b) Hyperledger Fabric

Figura 8 – Compromisso entre custo e desempenho (normalizado) por nó em função da carga: melhores compromissos são os valores menores das curvas.

[7] O plano *developer* atende até 200 mil requisições/dia a US$ 50/mês (https://www.infura.io/pricing).
[8] https://www.hyperledger.org/learn/case-studies

5 Cliente Aferidor de redes Blockchain: PBFT-APM

Como resultado prático desse trabalho foi desenvolvido a aplicação cliente PBFT-APM para inferir desempenho de redes blockchains que utilizam o mecanismo de consenso PBFT (Practical Byzantine Fault Tolerance). A abreviação APM (active passive measurement), por sua vez, significa que esse cliente está apito a realizar medições ativas e passivas na rede. Em outras palavras, a medição ativa consiste em enviar requisições sintéticas afim de medir o desempenho da rede, ao passo que a medição passiva consiste em observar as requisições reais da rede e gerar relatórios para análise de desempenho.

Este capítulo detalha o desenvolvimento e as funcionalidades do cliente PBFT-APM, destacando sua importância na inferência precisa do desempenho de redes blockchain baseadas em PBFT. Além disso, serão explicadas as fases do Protocolo PBFT, e haverá uma seção de Avaliação na qual examinaremos o comportamento do cliente com requisições sintéticas.

5.1 Fases do Protocolo PBFT

A proposta do cliente PBFT-APM foi inspirada na aplicação cliente Caliper utilizada neste trabalho. Essa aplicação é uma das mais populares atualmente, sendo utilizada em vários projetos de pesquisa e também comerciais que envolvem medições de desempenho de redes blockchains com diferentes mecanismos de consenso [1]. Contudo, observamos que o consenso PBFT tem peculiaridades que não são tratadas pelo Caliper. Especificamente, ele compreende três etapas de verificação de consistência e integridade de requisições submetidas à rede que requer alto nível de comunicação entre os nós participantes da rede. Por sua vez, essa comunicação tem impacto relevante no resultado final das requisições, em especial, o tempo médio de resposta e o custo da infraestrutura da rede para obter tempos níveis desejados de desempenho, analisados nesse trabalho.

A Figura 9 ilustra as fases do protocolo PBFT. Inicialmente, ocorre a etapa de pré-preparo, na qual a operação é enviada pelo cliente (identificado como C na figura) e o nó primário (denominado P) propõe uma operação específica, acompanhada por um carimbo de tempo, para ser executada em uma visão e sequência determinadas. Essa proposta é transmitida aos outros nós da rede, incluindo os *backups* (B1, B2, B3), para antecipadamente informá-los sobre a operação proposta, permitindo que eles se preparem para confirmá-la. No cenário em questão, supomos que o nó B2 esteja com defeito, assim

[1] https://hyperledger.github.io/caliper/

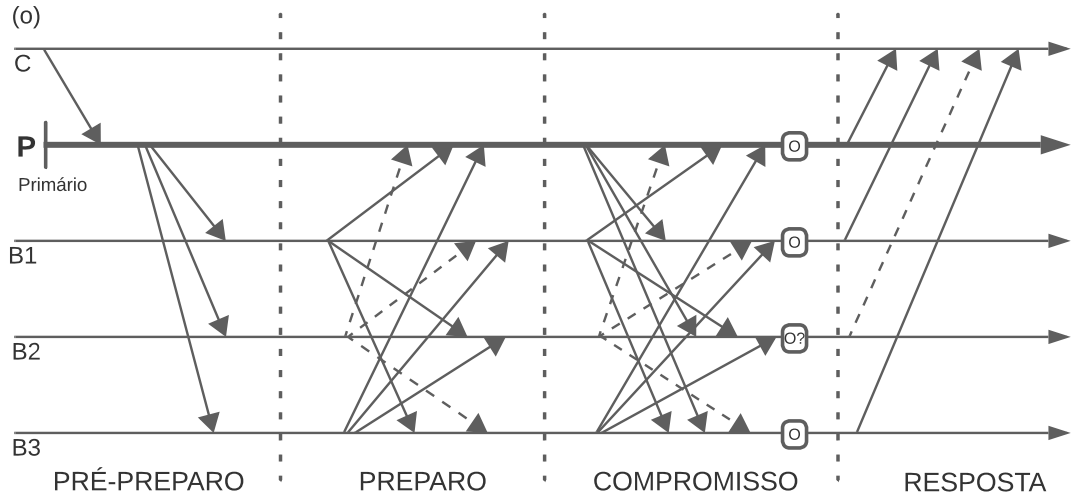


Figura 9 – As diferentes fases do PBFT, adaptado de (STEEN; TANENBAUM, 2023).

os backups não defeituosos concordam com pré-preparo se necessário e enviam mensagens preparo para outros, incluindo o primário.

Na fase subsequente, conhecida como fase de preparo, os nós da rede, incluindo o nó primário, avaliam a validade da operação proposta e a sequência na qual ela deve ser executada. Cada nó gera uma mensagem de "preparo" para indicar sua concordância com a operação, juntamente com o carimbo de tempo correspondente. É possível perceber que independentemente das ações de B2, backups não defeituosos (B1 e B3) concordam em relação à execução da operação "o" após registrar as mensagens pertinentes, e isso possibilita a progressão para a próxima fase.

Seguindo adiante, entramos na fase de comprometimento, na qual os nós da rede confirmam a validade da operação proposta e concordam com a ordem de execução. Cada nó gera uma mensagem de "comprometimento" para demonstrar seu compromisso em executar a operação na sequência designada. Tanto o primário quanto os backups B1 e B3 concordam com a execução da operação "o" devido às mensagens coincidentes, e cada servidor está ciente da cooperação de outros dois servidores para executar a ação, nota-se que o que B2 enviou não tem influência na decisão dos demais.

Após a conclusão bem-sucedida do processo de consenso, quando uma operação é confirmada e executada por um nó da rede, uma resposta é enviada de volta ao cliente que originou a solicitação, essa resposta comunica ao cliente que a operação foi realizada com êxito e pode incluir informações relevantes sobre os resultados obtidos. O cliente confirma a resposta considerando um mínimo de $k + 1$ réplicas, garantindo a participação de pelo menos um servidor de réplica não defeituoso. Independentemente das alegações de B2, o cliente pode confiar que a operação requisitada foi executada pela maioria dos

servidores.

É importante observar que o consenso PBFT é mais adequado às redes blockchains permissionadas. Logo, redes corporativas que agregam organizações distintas e que precisam de um sistema seguro e auditável para compartilhar dados têm com blockchains PBFT, a ferramenta ideal para o gerenciamento de dados. Por outro lado, as redes blockchain públicas populares como Ethereum e Bitcoin utilizam outros mecanismos de consenso diferentes de PBFT por permitirem participantes desconhecidos na rede, i.e., clientes sem permissão/identificação, e também não focarem na qualidade de serviço e desempenho. Esse último aspecto, se deve à essência de redes blockchains públicas, que é atender milhares de requisições de participantes anônimos (i.e., identificados apenas pelo número da chave pública) mantendo consistência e integridade e autenticidade das requisições registradas na blockchain pública. Essas redes já contam com ferramentas de aferição e análise de desempenho como o (CALIPER, 2019) e não são o foco do cliente PBFT-APM.

5.2 Implementação do Cliente PBFT-APM para Hyperledger Fabric

Atualmente, Hyperledger Fabric é o *framework* para desenvolvimento de redes blockchains permissionadas mais popular a utilizar o mecanismo de consenso PBFT. Dessa forma, implementamos o cliente PBFT-APM para esse *framework*. Para facilitar a interação com a rede do ambiente Hyperledger Fabric o cliente utiliza as tecnologias *Node.js* e *TypeScript*, em conjunto com a biblioteca Fabric Gateway. Essa biblioteca é mantida pela fundação Hyperledger, que oferece os recursos essenciais para estabelecer conexões com redes Fabric e realizar o envio de transações de forma eficiente e segura. Fabric Gateway é um serviço introduzido nos nós do Hyperledger Fabric v2.4 e fornece uma API (*Application Programming Interface*) mínima e simplificada para enviar transações para a rede [2].

A API (*Application Programming Interface*) Fabric Gateway permite não apenas efetuar transações, mas também exercer um controle mais detalhado sobre cada etapa desse processo. Isso abrange a avaliação das propostas de transações, a invocação de contratos inteligentes, iniciando da obtenção de endossos, a submissão e espera da ordenação das transações endossadas, e por fim a submissão e espera da transação na blockchain, i.e., efetivação da transação. Note que essas três etapas endosso, ordenação e efetivação são as implementações do protocolo PBFT no Hyperledger Fabric. Importante ainda mencionar que esse conjunto de funcionalidades não só possibilita submissões e consultas de transações seguras e estruturadas na blockchain via um nó gateway da rede, como também possibilita a análise precisa do tempo dedicado a cada etapa da transação, aprimorando assim

[2] https://hyperledger-fabric.readthedocs.io/en/latest/gateway.html

a análise de desempenho detalhada da rede. Portanto, o cliente PBFT-APM que implementamos para Hyperledger Fabric combina as etapas de endosso/ordenação/efetivação em uma único procedimento de submissão da transação (i.e., método *SubmitTransaction*). Contudo, o cliente possibilita medições individuais de cada etapa para medições ativas e passivas de desempenho da rede.

5.3 Funcionamento do Cliente PBFT-APM

O clente em questão é um *script TypeScript* que implementa uma interface de linha de comando (CLI) para interagir com uma rede blockchain baseada no Hyperledger Fabric. O Hyperledger Fabric é um framework de blockchain permissionado, projetado para atender às necessidades de aplicações empresariais, oferecendo recursos avançados como privacidade, modularidade e escalabilidade. Ele utiliza as bibliotecas, gRPC (*Google Remote Procedure Call*) para comunicação remota, o Hyperledger Fabric Gateway para interação com contratos inteligentes, e a biblioteca crypto para operações criptográficas.

As configurações iniciais envolvem a definição de constantes como o nome do canal na blockchain, o nome do contrato inteligente, o MSP (*Membership Service Provider*), e caminhos para certificados e chaves criptográficas. Essas configurações podem ser fornecidas através de variáveis de ambiente ou têm valores padrão. Existem funções específicas para a criação da conexão gRPC (*newGrpcConnection*()), obtenção de identidade (*newIdentity*()), e obtenção de assinante (*newSigner*()). Estas funções abstraem complexidades relacionadas à segurança, envolvendo a leitura de certificados TLS (*Transport Layer Security*), chaves privadas, e a configuração de identidades associadas ao MSP.

O cliente oferece implementações para operações típicas em contratos inteligentes. A função *initLedger*() inicializa o *ledger*, enquanto *getAllAssets*() realiza uma consulta para obter todos os ativos registrados no *ledger*. A função *createAsset*() cria novos ativos no ledger através de transações com o contrato inteligente. As funções *createAssetEndorse*() e *createAssetEndorseBenchmarks*() implementam uma lógica mais avançada para a criação de ativos, envolvendo a fase de endosso antes da confirmação definitiva no *ledger*. A função createAssetEndorse envia transações para o método *CreateAsset* e realiza o processo de endosso e confirmação de forma assíncrona. A função *createAssetEndorseBenchmarks* realiza um processo semelhante, mas adiciona a medição de tempos para avaliação de desempenho. Cada operação envolve chamadas específicas ao contrato inteligente na rede Hyperledger Fabric.

Ainda existem funções, *transferAssetAsync*() que é projetada para transferir a propriedade de um ativo de forma assíncrona, ela envia uma transação para o método *TransferAsset* do contrato inteligente, informando o identificador do ativo e o novo proprietário. A função aguarda a confirmação da transação no *ledger* antes de exibir mensagens de sucesso ou falha. A função readAssetByID() é responsável por consultar as informações

de um ativo específico no ledger. Ela envia uma transação para o método ReadAsset do contrato inteligente, fornecendo o identificador do ativo como argumento. O resultado da transação é decodificado e exibido no console. Função *updateNonExistentAsset*() simula a atualização de um ativo que não existe no ledger, ele envia uma transação para o método *UpdateAsset* do contrato inteligente, informando o identificador do ativo, cor, tamanho, proprietário e valor. Se o ativo não existir, a função captura o erro resultante da transação.

A configuração inicial envolve a clonagem do repositório dentro do ambiente de teste *fabric-samples/test-network*, seguida da navegação até a pasta do projeto e instalação das dependências necessárias via *npm install*. Essas dependências são essenciais para a interação com as redes Fabric e a execução de transações. O processo de *build*, acionado pelo comando *npm run build*, é fundamental para compilar o código fonte do cliente PBFT-APM, garantindo sua prontidão para execução.

A utilização do `cliente PBFT-APM` para operações na blockchain é facilitada por meio de comandos específicos iniciados com `npm start`, seguidos por argumentos que definem a ação desejada.

- `npm start initLedger`: Inicia a blockchain, preenchendo-a com ativos pré-definidos.
- `npm start createAsset [n]`: Cria novos ativos, onde `[n]` é a quantidade desejada.
- `npm start createAssetEndorse [n]`: Cria ativos com operações de Endorse/Submit/CommitStatus para avaliar o desempenho da rede.
- `npm start getAll`: Recupera e exibe todos os ativos presentes na blockchain.
- `npm start getByKey [id]`: Recupera e exibe um ativo específico com base em seu ID `[id]`.
- `npm start transferAsset [id] [newOwner]`: Transfere a propriedade de um ativo, indicando o `[id]` do ativo e o `[newOwner]` como o novo proprietário.
- `npm start updateAsset [id]`: Atualiza um ativo existente com base em seu ID `[id]`.

Um exemplo de utilização científica seria a avaliação do desempenho da rede ao criar 100 novos ativos e medir operações de *Endorse/Submit/CommitStatus*. Isso pode ser realizado com o comando *npm start createAssetEndorse 100*, exemplificando como o cliente PBFT-APM pode ser empregado em experimentos científicos para avaliar a eficiência da rede blockchain implementada no Hyperledger Fabric.

Além da execução convencional do cliente PBFT-APM, há uma alternativa para realizar experimentos e gerar dados de desempenho. Para isso, é possível utilizar um *script Python* denominado runBenchAle.py. Este *script* é projetado para automatizar a

execução do cliente, realizando operações de criação de ativos com operações de *Endorse/Submit/CommitStatus* em intervalos específicos de tempo. O *script* utiliza um conjunto de tempos aleatórios previamente registrados em arquivos CSVs (*Comma-separated values*). Esses tempos representam os intervalos entre operações sucessivas e são essenciais para simular cargas variáveis na rede blockchain.

Ao executar o *script* runBenchAle.py, os tempos exponenciais são lidos do arquivo CSV, e para cada intervalo de tempo, o cliente PBFT-APM é acionado em segundo plano para realizar uma operação de criação de ativo. Os resultados são registrados em um arquivo de saída especificado, como um arquivo CSV. É importante observar que a explicação detalhada dos arquivos de tempos utilizados por este script será abordada na Seção 5.4, proporcionando uma compreensão mais aprofundada do processo experimental.

O código-fonte do cliente PBFT-APM desenvolvido para interação com o mecanismo de consenso PBFT em redes blockchain, especialmente no *framework* Hyperledger Fabric, está disponível publicamente. Você pode acessar e explorar o código do cliente PBFT-APM no seguinte repositório do GitHub: [3]. Este repositório contém a implementação do cliente, incluindo funcionalidades para análise de desempenho e medições ativas e passivas em redes blockchain que utilizam o mecanismo PBFT

5.4 Avaliação

Para avaliar a performance e a adaptabilidade do cliente PBFT-APM em cenários do mundo real, conduzimos uma série de experimentos. Esses experimentos foram projetados para analisar o comportamento do cliente em situações variadas, incluindo o uso de uma distribuição de tempos entre requisições.

Utilizamos tempos entre requisições coletados em um sistema distribuído real para avaliar o cliente PBFT-APM. O dados coletados do sistema de armazenamento em nuvem Dropbox disponibilizado em Gonçalves et al. (2016)[4] foi utilizado nessa avaliação. Especificamente, utilizamos o conjunto de dados "Camp1" que abrangeu um período de abril de 2014 a junho de 2014. Este conjunto de dados contém informações sobre o tamanho dos arquivos, dispositivos envolvidos, volume de dados e carimbos de tempo (*timestamps*).

Foi aplicado o Método da Estimação de Máxima Verossimilhança (MLE) para estimar os parâmetros da distribuição Exponencial, utilizando a coluna de *timestamps*. Posteriormente, foram realizados testes Kolmogorov-Smirnov (KS) para validar a adequação das distribuições aos dados observados. Os parâmetros resultantes dessas análises foram então utilizados na geração de tempos aleatórios, essenciais para a criação de cargas sintéticas representativas.

[3] https://github.com/LABPAAD/blockchain_performance
[4] https://sites.google.com/a/ufpi.edu.br/ggoncalves/traces-html

Em suma, a caracterização do Dropbox por meio da análise do conjunto de dados do "Camp1" proporcionou uma compreensão detalhada dos padrões de requisições no sistema. A aplicação da distribuição exponencial, combinada com testes de ajuste como o Kolmogorov-Smirnov, permitiu-nos modelar de maneira eficaz os tempos entre requisições. Os parâmetros estimados dessas distribuições forneceram uma base sólida para a geração de tempos aleatórios, viabilizando a criação de cargas sintéticas realistas. Essas cargas foram essenciais para avaliar o cliente PBFT-APM em condições que refletem a diversidade de operações observadas em ambientes reais do Dropbox.

Com o intuito de simular uma ampla variedade de cenários de interação com o cliente PBFT-APM, geramos requisições sintéticas, utilizando tempos entre requisições derivados da caracterização do Dropbox. Esses intervalos foram armazenados em um arquivo utilizado como entrada para o cliente PBFT-APM nos experimentos. O cliente, por sua vez, utilizará esses tempos para gerar requisições, seguindo a distribuição de requisições por segundo ou tempo entre requisições da distribuição caracterizada.

A Figura 10 mostra a distribuição Exponencial, utilizando como referência o conjunto de tempos derivados da caracterização do Dropbox. Esse gráfico proporciona uma visualização dos tempos entre requisições simimulados. A curva laranja foi originada de dados sintéticos da distribuição Exponencial caracterizada, cujo tempo médio entre as requisições é 7,57 segundos (parâmetro *lambda*). A curva azul representa os dados reais caracterizados do Dropbox no "Camp1" cuja a média entre os tempos entre requisições é 7,12 segundos. Como pode ser observado o método MLE ajustou adequadamente a distribuição Exponencial aos dados reais. O teste estatístico Kolmogorov-Smirnov descrito no referencial teórico mostra uma distância 0,052 e um alpha 0,05, o que oferece suporte estatístico para que essa distribuição represente os dados reais.

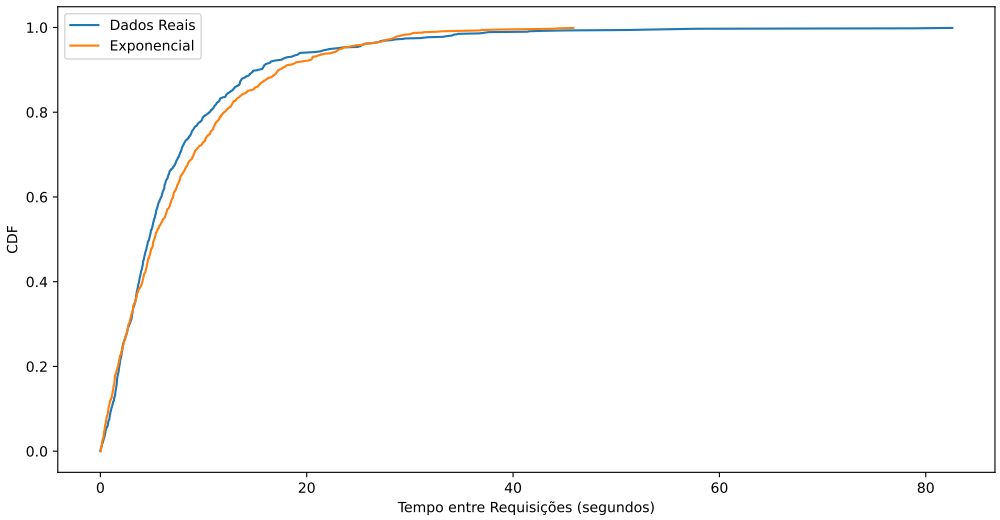


Figura 10 – CDF da distribuição Exponencial para tempos entre transações com dados reais e distribuição de probabilidade ajustada com o método MLE.

Com base na análise anterior, realizamos um teste, empregando os tempos obtidos na caracterização do Dropbox para 1000 submissões. A ênfase desse teste é observar o comportamento do sistema ao utilizar os tempos ajustados, semelhantes aos do Dropbox. Os resultados desse experimento estão refletidos na representação gráfica a seguir, destacando as curvas CDF como uma manifestação direta do desempenho do modelo. Essa visualização permite uma compreensão mais completa do desempenho do ajuste em um cenário simulado.

Na Figura 11 mostra os tempos entre requisições gerados pelo cliente e a distribuição exponencial caracterizada. A curva laranja foi originada de dados sintéticos da distribuição Exponencial caracterizada assim como na Figura 10. A curva azul, por sua vez, representa os dados reais gerados pelo cliente cuja a média entre os tempos entre requisições é 7,009 segundos. Pode ser observado que o cliente foi capaz de gerar requisições sintéticas seguindo a distribuição exponencial utilizada como parâmetro de entrada.

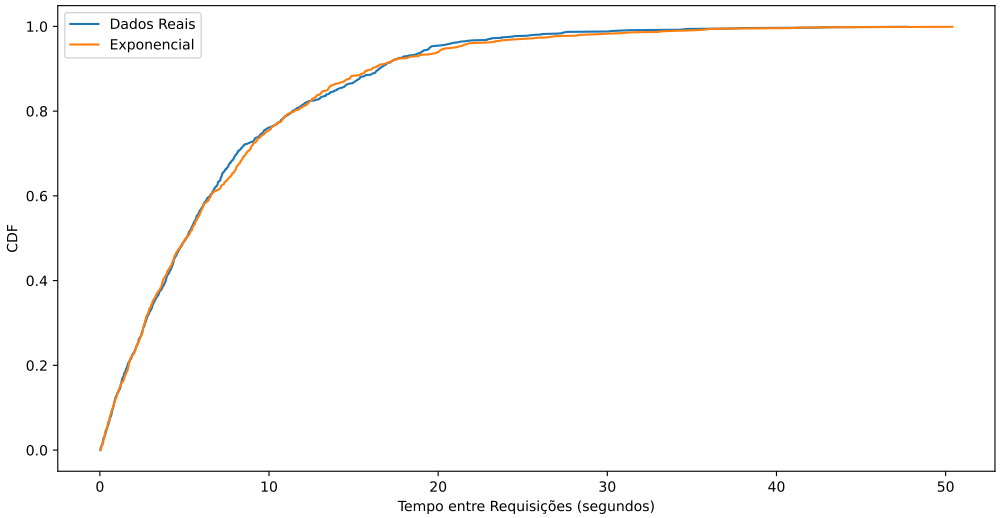


Figura 11 – CDF da distribuição exponencial para tempos entre transações com dados reais do cliente (curva azul) e distribuição exponencial (curva laranja).

As análises do desempenho do cliente PBFT-APM, obtidas por meio dos gráficos, são valiosas. No entanto, é crucial notar que o cliente ainda está em fase de desenvolvimento, tornando os resultados atuais não definitivos. A evolução constante do software destaca a necessidade de uma abordagem flexível, considerando ajustes e refinamentos contínuos. Ao interpretar os resultados, é essencial ter em mente a natureza dinâmica do processo de desenvolvimento do cliente PBFT-APM.

6 Conclusão

Em síntese, este trabalho propõe um modelo de infraestrutura, em específico a arquitetura do nó da rede considerando o nó de consenso e P2P, para comparar e avaliar ao mesmo tempo os fatores desempenho, uso de recursos computacionais e custo para uma organização participar de uma rede blockchain. A avaliação e comparação foi conduzida por experimentos desenvolvidos a partir do modelo da arquitetura proposta tendo em vista nós com diferentes capacidades computacionais na plataformas Ethereum e Hyperledger Fabric. Os resultados evidenciaram os melhores compromissos entre custo e desempenho para cada uma destas plataformas e em cada um dos recursos computacionais utilizados. O modelo foi proposto como solução para o problema de compatibilizar os pares de redes blockchain pública e privada afim de realizar uma avaliação justas em função dos custos, uso de recursos e desempenho. Entretanto, o modelo tem potencial para expansão de propostas de novas arquiteturas, por meio de desenvolvimentos a serem implementadas como trabalhos futuros.

Este trabalho propõe uma avaliação da infraestrutura blockchain, necessária para prover acesso de aplicações à rede, traçando uma comparação de desempenho entre as plataformas Ethereum e Hyperledger Fabric. Foi avaliado um modelo de custo por transação para aplicações em redes blockchain pública e permissionada, considerando simultaneamente o desempenho máximo em função da infraestrutura e carga de trabalho imposta. O trabalho conduziu um experimento com a implementação de aplicações nas duas plataformas para aplicação do modelo em diferentes tipos de infraestrutura. Como resultado, foi fornecido um modelo capaz de estimar o custo da infraestrutura por transação confirmada na blockchain, considerando redes públicas e permissionadas. A partir do modelo proposto, os resultados mostraram os limites de escalabilidade dessas redes e os compromissos entre custo e desempenho para aplicações blockchain.

Com base nas análises conduzidas neste trabalho, é possível reconhecer a importância do cliente desenvolvido como uma ferramenta para a avaliação de redes blockchain baseadas em PBFT. Ao permitir medições ativas e passivas na rede, o cliente oferece uma abordagem para a avaliação do desempenho, proporcionando informações para desenvolvedores e operadores de redes blockchain. No entanto, é importante ressaltar que, como qualquer software em desenvolvimento, ele está sujeito a ajustes e melhorias contínuas. Como trabalho futuro, espera-se que o cliente seja aperfeiçoado e refinado para fornecer uma avaliação ainda mais precisa do desempenho das redes blockchain. Especificamente, melhorias na granularidade das medições com emissão de tempos entre requisições menores que um segundo. O aprimoramento contínuo do PBFT-APM abre caminho para avanços na avaliação de desempenho em blockchain.

7 Publicações

Durante o desenvolvimento do estudo foi feita a publicação de um artigo nos Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Além disso, houve a submissão e apresentação de trabalhos para o XIV SDTI: Seminário de Desenvolvimento Tecnológico e Inovação - 2022 e para o XV SDTI - 2023.

- MENDONÇA, Ronan Dutra; MOURA, Ericksulino Manoel de Araújo; GONÇALVES, Glauber Dias; VIEIRA, Alex Borges; NACIF, José A. M.. Comparação e Análise de Custo e Desempenho entre Nós de Redes Blockchain Permissionadas e Públicas. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC), 41. , 2023, Brasília/DF. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2023 . p. 141-154. ISSN 2177-9384. DOI: https://doi.org/10.5753/sbrc.2023.423.

- MOURA, Ericksulino Manoel de Araújo; GONÇALVES, Glauber Dias. Desenvolvimento de Ferramentas para o Gerenciamento de Acesso a Dados e Aquisição de Informação em Aplicações Baseadas na Tecnologia Blockchain. Apresentado na modalidade PÔSTER durante o XIV SDTI: SEMINÁRIO DE DESENVOLVIMENTO TECNOLÓGICO E INOVAÇÃO - 2022, realizado na UFPI – Campus Ministro Petrônio Portella – Teresina – PI.

- MOURA, Ericksulino Manoel de Araújo; GONÇALVES, Glauber Dias. Monitoramento de DApps em Diferentes Redes Blockchains. Apresentado na modalidade COMUNICAÇÃO ORAL durante o XV SDTI: SEMINÁRIO DE DESENVOLVIMENTO TECNOLÓGICO E INOVAÇÃO - 2023, realizado na UFPI – Campus Ministro Petrônio Portella – Teresina – PI.

Referências

ANDROULAKI, E.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In: *Proc. of the EuroSys Conference*. [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 12 e 25.

ARTAYA, I. P. *KOLMOGOROV-SMIRNOV TEST*. 2019. Citado na página 19.

BABU, G. J. A note on maximum likelihood estimation for mixture models. *Journal of the Korean Statistical Society*, Springer, v. 51, p. 1327–1333, 2022. Disponível em: <https://link.springer.com/article/10.1007/s42952-022-00180-6>. Citado 2 vezes nas páginas 18 e 19.

BALIGA, A. et al. Performance characterization of hyperledger fabric. *Proceedings - 2018 Crypto Valley Conference on Blockchain Technology, CVCBT 2018*, IEEE, p. 65–74, 2018. Citado 3 vezes nas páginas 12, 20 e 22.

CALIPER, H. *Caliper*. 2019. Https://hyperledger.github.io/caliper. (Accessed on 10/23/2023). Citado 2 vezes nas páginas 29 e 36.

CASTRO, M.; LISKOV, B. et al. Practical byzantine fault tolerance. In: *OsDI*. [S.l.: s.n.], 1999. v. 99, n. 1999, p. 173–186. Citado na página 17.

CHAN, S. H.; WEN, H. *Lecture 8: Properties of Maximum Likelihood Estimation (MLE)*. 2015. Disponível em: <https://engineering.purdue.edu/ChanGroup/ECE645Notes-/StudentLecture08.pdf>. Citado na página 18.

CHOI, W.; HONG, J. W. K. Performance Evaluation of Ethereum Private and Testnet Networks Using Hyperledger Caliper. *22nd APNOMS 2021*, IEICE, p. 325–329, 2021. Citado na página 21.

CONG, Z. et al. Comprehensible counterfactual explanation on kolmogorov-smirnov test. *arXiv preprint arXiv:2011.01223*, 2020. Citado na página 19.

EIGER, A. M.; NADLER, B.; SPIEGELMAN, C. The calibrated kolmogorov-smirnov test. *arXiv preprint arXiv:1311.3190*, 2013. Citado na página 19.

FABRIC, H. *Welcome to Hyperledger Fabric*. 2023. Disponível em: `https://hyperledger-fabric.readthedocs.io/en/release`. Acesso em: 04 de outubro 2023. Citado na página 16.

FOUNDATION, H. *About Hyperledger Foundation*. 2023. Disponível em: `https://www.hyperledger.org/about`. Acesso em: 04 de outubro 2023. Citado na página 16.

GONÇALVES, G. et al. The impact of content sharing on cloud storage bandwidth consumption. *IEEE Internet Computing*, IEEE, v. 20, n. 4, p. 26–35, 2016. Citado na página 39.

GREVE, F. et al. Blockchain e a Revolução do Consenso sob Demanda. In: *Proc. of SBRC Minicursos*. [S.l.: s.n.], 2018. Citado 4 vezes nas páginas 11, 14, 16 e 25.

Hyperledger Fabric Project. *Hyperledger Fabric Introduction*. 2023. Disponível em: https://https://hyperledger-fabric.readthedocs.io/en/release-2.2/blockchain.html#what-is-hyperledger-fabric. Acesso em: 04 de outubro 2023. Citado na página 17.

JR, F. J. M. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, Taylor Francis, v. 46, n. 253, p. 68–78, 1951. Citado na página 19.

LAURENCE, T. *Blockchain for dummies*. [S.l.]: John Wiley & Sons, 2019. Citado na página 15.

LEAL, F.; CHIS, A. E.; GONZÁLEZ–VÉLEZ, H. Performance Evaluation of Private Ethereum Networks. *SN Computer Science*, Springer Singapore, v. 1, n. 5, p. 1–17, 2020. ISSN 2662-995X. Disponível em: <https://doi.org/10.1007/s42979-020-00289-7>. Citado 3 vezes nas páginas 12, 21 e 22.

MAGALHÃES, M. N. *Probabilidade e variáveis aleatórias*. [S.l.]: Edusp, 2006. Citado na página 18.

MALIK, H. et al. Performance Analysis of Blockchain based SG with Ethereum and Hyperledger Implementations. *IEEE International Conference on ANTS*, 2019. ISSN 21531684. Citado 3 vezes nas páginas 12, 21 e 22.

MELO, C. et al. Distributed application provisioning over ethereum-based private and permissioned blockchain: availability modeling, capacity, and costs planning. *The Journal of Supercomputing*, Springer, p. 1–27, 2021. Citado 2 vezes nas páginas 21 e 22.

MONRAT, A. A.; SCHELEN, O.; ANDERSSON, K. Performance Evaluation of Permissioned Blockchain Platforms. *IEEE, CSDE 2020*, 2020. Citado 3 vezes nas páginas 12, 21 e 22.

NAKAMOTO, S. *Bitcoin: A peer-to-peer electronic cash system*. 2008. Citado 3 vezes nas páginas 11, 14 e 15.

NOFER, M. et al. Blockchain. *Business & Information Systems Engineering*, Springer, v. 59, n. 3, p. 183–187, 2017. Citado na página 14.

NOVAES, M. Distribuições de probabilidade. *Revista Brasileira de Ensino de Física*, SciELO Brasil, v. 44, p. e20210424, 2022. Citado na página 18.

QUEIROZ, F. G. G. e Leobino Sampaio Sampaio e Jauberth Abijaude Abijaude e Antonio Coutinho Coutinho e Ítalo Valcy Valcy e S. Q. Blockchain e a revolução do consenso sob demanda. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) - Minicursos*, 2018. Disponível em: <http://143.54.25.88/index.php/sbrcminicursos-/article/view/1770>http://143.54.25.88/index.php/sbrcminicursos/article/view/1770. Citado na página 14.

RIMBA, P. et al. Quantifying the Cost of Distrust: Comparing Blockchain and Cloud Services for Business Process Execution. *Information Systems Frontiers*, Information Systems Frontiers, v. 22, n. 2, p. 489–507, 2020. ISSN 15729419. Citado 2 vezes nas páginas 20 e 22.

ROUHANI, S.; DETERS, R. Performance analysis of ethereum transactions in private blockchain. In: IEEE. *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. [S.l.], 2017. Citado 3 vezes nas páginas 12, 21 e 22.

SPENGLER, A. C.; SOUZA, P. S. Avaliação de desempenho do hyperledger fabric com banco de dados para o armazenamento de grandes volumes de dados médicos. In: *Proc. of WPerformance*. [S.l.: s.n.], 2021. ISSN 2595-6167. Citado na página 29.

STEEN, M. V.; TANENBAUM, A. S. *Distributed Systems*. 4th. ed. [S.l.]: distributed-systems.net, 2023. Chapter 8: Fault Tolerance. Citado 2 vezes nas páginas 7 e 35.

THAKKAR, P.; NATHAN, S.; VISWANATHAN, B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. *Proceedings - IEEE, MASCOTS 2018*, p. 264–276, 2018. Citado 3 vezes nas páginas 12, 20 e 22.

WANG, C.; CHU, X. Performance characterization and bottleneck analysis of hyperledger fabric. *Proceedings - International Conference on Distributed Computing Systems*, v. 2020-Novem, p. 1281–1286, 2020. Citado 3 vezes nas páginas 12, 20 e 22.

WOOD, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, v. 151, p. 1–32, 2014. Citado 2 vezes nas páginas 11 e 16.

XU, X. et al. Latency performance modeling and analysis for hyperledger fabric blockchain network. *Information Processing and Management*, Elsevier Ltd, v. 58, n. 1, 2021. ISSN 03064573. Disponível em: <https://doi.org/10.1016/j.ipm.2020.102436>. Citado 3 vezes nas páginas 12, 21 e 22.

XU, X.; WEBER, I.; STAPLES, M. *Architecture for blockchain applications*. [S.l.]: Springer, 2019. Citado na página 11.

YAN, L. *20: Maximum Likelihood Estimation*. 2020. Disponível em: <https://web-.stanford.edu/class/archive/cs/cs109/cs109.1208/lectures/20_mle_blank-announcements-.pdf>. Citado na página 18.

ZHANG, L. et al. Ethereum transaction performance evaluation using test-nets. In: *Euro-Par 2019: Parallel Processing Workshops*. Cham: Springer International Publishing, 2020. Citado 3 vezes nas páginas 12, 16 e 21.

ÁVILA, C. C. e L. A tecnologia blockchain aplicada aos contratos inteligentes. *Revista Em Tempo*, v. 18, n. 01, p. 156–176, 2019. ISSN 1984-7858. Disponível em: <https://revista.univem.edu.br/emtempo/article/view/3210>. Citado na página 11.

TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA "JOSÉ ALBANO DE MACEDO"

Identificação do Tipo de Documento

( ) Tese

( ) Dissertação

(**X**) Monografia

( ) Artigo

Eu, **Ericksulino Manoel de Araújo Moura**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **Comparação e Análise de Custo e Desempenho entre Nós de Redes Blockchain Permissionadas e Públicas** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 10 de Fevereiro de 2024.

Ericksulino Manoel de Araújo Moura

Assinatura