

Modelagem e Avaliação de Desempenho da Hyperledger Fabric: Uma análise baseada no Fluxo de Transações e no Impacto de Políticas de Endossamento

Carlos Melo¹, Glauber D. Gonçalves¹, Francisco A. Silva¹, Iure Fé¹,
Ericksulino Moura¹ e André Soares¹

¹Universidade Federal de Piauí – PI – Brasil

{carlos.alexandre, ggoncalves, faps, iure.fe}@ufpi.edu.br

{ericksulino, andre.soares}@ufpi.edu.br

Abstract. *Blockchain is a paradigm arising from concepts such as distributed systems, protocols, and security. There are ongoing debates regarding its application in service provisioning and its feasibility in industrial settings, particularly concerning its performance. In blockchains, prolonged response times detrimentally affect user experience, and resource-intensive usage can escalate service provisioning costs. This article assesses the performance of Hyperledger Fabric (HLF), a platform for permissioned blockchains that focus on industrial environments, through modeling and experimentation. The results indicate that block size and transaction arrival rate compromise throughput, latency, and resource utilization in the environment by over 100%, and employing multiple gateways can reduce latency by more than 50%.*

Resumo. *Blockchain é um paradigma proveniente de conceitos como sistemas distribuídos, protocolos e segurança. Há debates sobre sua aplicação no provimento de serviços e da sua viabilidade em ambientes industriais, especialmente em relação ao seu desempenho. Em blockchains, tempos de resposta longos prejudicam a experiência do usuário, e o uso intensivo de recursos pode elevar os custos de provimento de serviço. Este artigo avalia o desempenho da Hyperledger Fabric (HLF), uma plataforma para blockchains permissionadas com foco em ambientes industriais, por meio de modelagem e experimentação. Os resultados indicam que o tamanho do bloco e a taxa de chegada comprometem vazão, latência e utilização no ambiente em mais de 100% e que múltiplos gateways podem reduzir em mais de 50% a latência.*

1. Introdução

Blockchains permissionadas diferem das públicas, como Bitcoin e Ethereum, principalmente ao focar em aplicações e sistemas industriais. Nessas *blockchains*, a identificação de todos os nós da rede é obrigatória, uma vez que a confiança mútua entre as partes envolvidas na concepção da rede pode não existir, o que ocorre Hyperledger Fabric (HLF), uma plataforma aberta mantida pela Hyperledger Foundation [Suankaewmanee et al. 2018].

Na HLF as transações se baseiam em um processo de três etapas: *endorsement*, *ordering* e *committing*, assegurando que cada transação seja validada e incorporada a um bloco que será adicionado à *blockchain* e replicado nos nós. Esse processo é derivado do

modelo de consenso baseado no protocolo *practical Byzantine Fault Tolerance* (pBFT) [Sukhwani et al. 2017] e é a base das aplicações desenvolvidas para a HLF.

As aplicações no âmbito da HLF recebem o nome de *chain codes* e se baseiam no conceito de contratos inteligentes definido para redes Ethereum [Buterin et al. 2013]. Geralmente, transações realizadas em redes permissionadas não estão vinculadas a um consumo de *gas* ou monetário, ou seja, não é preciso pagar nada para que uma transação seja efetuada, o que facilita sua utilização em ambientes industriais que necessitam de uma maior vazão e uma menor latência ao menor custo financeiro possível. Porém, quantificar os recursos necessários ao provimento e a plena utilização dessas aplicações tornou-se uma tarefa importante e diretamente relacionada ao processo de avaliação de desempenho.

O desempenho no contexto de *blockchain* é um fator-chave utilizado para demonstrar a viabilidade da adoção dessa tecnologia e tornou-se um tópico de pesquisa relevante nos últimos anos, com estudos focados na mensuração e na modelagem de desempenho do sistema [Xu et al. 2021, Thakkar et al. 2018, Guggenberger et al. 2022, Sukhwani et al. 2018, Melo et al. 2022, Wu et al. 2022, Jiang et al. 2020, Ke and Park 2023, Yuan et al. 2020]. Porém, devido à volatilidade desse campo de pesquisa, a maioria dos trabalhos realizados não considera as novas versões e readequações inseridas no contexto dessas plataformas e como essas mudanças impactam no desempenho do serviço.

Em [Silva et al. 2023] nós realizamos a modelagem do ambiente Fabric com uma rede de Petri Estocástica (SPN) e obtivemos um conjunto de resultados que pontuaram o impacto do tamanho do bloco sobre a latência e a vazão do sistema. Contudo, o modelo proposto não foi validado e não acomodou novas características trazidas pela versão 2.5 da plataforma lançada em 2023, como a adição de um *gateway* para gestão de transações entre clientes e nós e mecanismos de endosso personalizados. O presente artigo expande o anterior e utiliza de mensuração e modelagem para compor um *framework* para avaliação de desempenho no âmbito da HLF 2.5 e posterior.

Este trabalho oferece duas principais contribuições: (i) modelo de desempenho validado experimentalmente e; (ii) uma avaliação de desempenho generalizada por meio de estudos de caso. O modelo proposto possibilita que provedores de serviços em HLF ajustem parâmetros conforme sua infraestrutura, focando na vazão, latência, e utilização de recursos. Além disso, avaliamos o impacto da política de endossamento e do endosso distribuído sobre essas métricas, possibilitando que organizações que não compartilhem confiança mútua possam ainda assim usufruir dos benefícios trazidos pelo uso de *blockchains*.

As próximas seções estão organizadas como segue: a Seção 2 apresenta os trabalhos relacionados em modelagem e avaliação de desempenho de *blockchains*. A Seção 3 apresenta uma visão geral do fluxo de transações no âmbito da plataforma Hyperledger Fabric. A Seção 4 apresenta o modelo base proposto e discute suas aplicações. A Seção 5 descreve o estudo experimental e o processo de validação do modelo proposto para a plataforma HLF. A Seção 6 apresenta um conjunto de estudos de caso e derivações do modelo proposto, bem como os principais resultados obtidos em cada cenário de avaliação. Finalmente, a Seção 7 resume nossas considerações finais, limitações e trabalhos futuros.

2. Trabalhos Relacionados

A plataforma Hyperledger Fabric foi amplamente discutida em pesquisas anteriores, com enfoque especial em métricas de desempenho. Este artigo busca atualizar o estado da arte e preencher lacunas deixadas por estudos existentes.

Em [Melo et al. 2022], nós introduzimos modelos para avaliar a utilização de recursos computacionais no contexto da HLF. Utilizando Cadeias de Markov de Tempo Contínuo (CTMCs) e Redes de Petri Estocásticas (SPNs) demonstramos a eficácia desses formalismos na modelagem e avaliação de aplicações baseadas em HLF, particularmente na detecção de gargalos de infraestrutura. Contudo, o modelo proposto neste artigo expande o escopo de [Melo et al. 2022], ao avaliar métricas como vazão e latência de transações.

Em outros estudos como [Wu et al. 2022, Jiang et al. 2020, Ke and Park 2023] os autores modelaram métricas gerais de desempenho de sistemas sob a perspectiva da HLF. Jiang et al. [Jiang et al. 2020], por exemplo, utilizaram uma abordagem de modelagem hierárquica para o HLF 1.4, analisando indicadores como: vazão, latência e utilização do sistema.

Enquanto que, [Wu et al. 2022] desenvolveram um modelo baseado em teoria de filas focado no fluxo de uma transação no âmbito da HLF 2.0. Ke e Park [Ke and Park 2023] propuseram modelos de filas considerando diferentes taxas de serviço. Entretanto, esses estudos não incluíram um método formal de detecção de gargalos como a aplicação de uma análise de sensibilidade para a quantificação do impacto causado por cada componente da plataforma sobre a métrica de interesse.

Por fim, em outros trabalhos como [Sukhwani et al. 2018, Yuan et al. 2020], os autores avaliaram a influência de taxas de chegada e tamanhos de bloco na vazão e latência do HLF, mas esses não são os únicos fatores que impactam significativamente essas métricas, como veremos no presente trabalho. Além disso, em Yuan et al. [Yuan et al. 2020], os autores modelaram a plataforma por meio de uma Generalized Stochastic Petri Nets (GSPN), enquanto que em Sukhwani et al. [Sukhwani et al. 2018] os autores usaram Stochastic Reward Networks (SRN). Ambos os modelos são isomorfos as SPNs apresentadas no presente trabalho, ou seja, possuem o mesmo poder de representatividade, porém, não contemplam o fluxo de transações para versões mais recentes da HLF.

O presente trabalho amplia esses estudos, sobretudo estende diretamente nosso trabalho anterior [Silva et al. 2023], ao apresentar um modelo SPN para a plataforma Hyperledger Fabric, com foco na versão HLF 2.5+ abordando o conceito de *gateway* e a possibilidade de o mesmo ser um ponto único de falha e um gargalo pré-plataforma. A metodologia proposta no desenvolvimento deste trabalho inclui uma validação formal e identificação de gargalos no modelo proposto, aumentando a relevância prática das descobertas por meio de um conjunto de estudos de caso.

3. Arquitetura do Hyperledger Fabric

Esta seção apresenta a arquitetura da Hyperledger Fabric e como os componentes se relacionam entre si. A Figura 1 demonstra o fluxo de transações na rede por meio de um processo subdividido em quatro etapas: *proposta*, *endorsement*, *ordering* e *commit*.

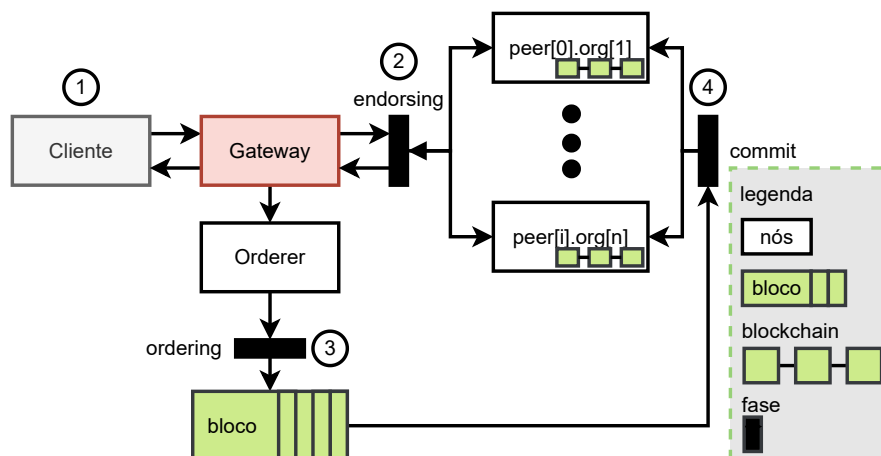


Figura 1. Hyperledger Fabric - Fluxo de Transações

1. A partir da versão 2.5 da HLF, o fluxo inicia com a aplicação cliente enviando uma proposta de transação a um *gateway*, que atua como intermediário entre o cliente e os pares (*peers*);
2. Os pares endossadores irão simular as transações e determinar se as mesmas atendem aos requisitos pre-estabelecidos no *chain code* da aplicação;
3. A transação é então encaminhada aos pares ordenadores (*orderer*) que irão inseri-la em um bloco em conjunto com outras transações;
4. O bloco gerado é posteriormente persistido na *blockchain* através do processo de *commit*.

O conceito de transação depende da aplicação e pode variar desde a criação de um novo objeto até a transferência ou consulta de informações relacionados a este objeto. Na fase de endosso, em uma transação de transferência de bens entre contas, por exemplo, é comum que a simulação realizada verifique se o remetente é de fato o proprietário daquele bem e que o destinatário realmente existe. É comum que os pares endossadores pertençam a diferentes organizações (*org*) geograficamente distribuídas que não possuem confiança mútua umas nas outras.

No retorno da transação ao *gateway*, a mesma será reencaminhada aos ordenadores, responsáveis pela gerência de novos blocos. Os blocos possuem um tamanho predefinido e um tempo máximo de espera para serem formados (*batch timeout*). Caso o tempo máximo seja atingido, o bloco será considerado finalizado, mesmo que ainda esteja parcialmente preenchido. Uma vez ordenadas as transações em um bloco, este será enviado aos endossadores que irão processá-lo e inseri-lo em sua respectiva cópia da *blockchain*, com objetivo de garantir integridade e imutabilidade dos registros.

4. Redes de Petri e Modelo Proposto

Esta seção apresenta o modelo proposto que será utilizado para avaliar o desempenho da plataforma Hyperledger Fabric (HLF). Antes de descrever o modelo, a seção a seguir provê uma visão geral sobre modelagem por meio das redes de Petri estocásticas (SPN).

4.1. Redes de Petri

As Redes de Petri Estocásticas (SPNs) possuem estados cujas transições estão associadas a eventos temporizados estocasticamente distribuídos. Extensões como as SPNs generali-

zadas (GSPN) adicionaram transições imediatas e condições para ocorrência de eventos, o que permitiu uma maior facilidade na modelagem de problemas reais. Por concisão, adotamos o termo SPN para se referenciar a família de modelos derivados das SPN. As Redes de Petri Estocásticas possibilitam a modelagem e análise probabilística de sistemas. A propriedade de ausência de memória da distribuição exponencial nos atrasos de disparo implica que as SPNs são isomórficas às cadeias de Markov de tempo contínuo (Continuous Time Markov Chains, CTMCs), proporcionando medidas de desempenho e confiabilidade [Maciel 2023].

A Figura 2 apresenta um exemplo simples de uma SPN e seus principais componentes. Arcos (arestas direcionadas) conectam lugares a transições e vice-versa. Tokens (marcações) denotam o estado atual do sistema em uma SPN ou o número de recursos disponíveis. No contexto deste artigo, cada *token* que chega no sistema significa uma nova transação a ser entregue ao HLF.

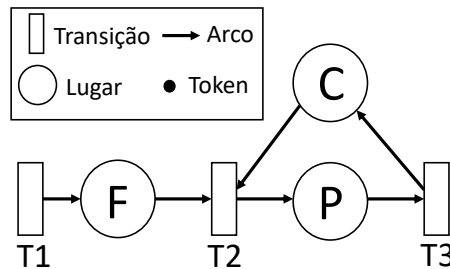


Figura 2. Exemplo simples de uma SPN

O comportamento de uma SPN é definido em termos de um fluxo de *tokens*. A realização de uma ação ou evento (disparo de uma transição) no sistema está ligada a pré-condições. Tokens são criados e destruídos conforme os disparos de transição.

Cada transição possui um tempo associado que segue uma determinada distribuição de probabilidade. No exemplo da Figura 2, há geração de novos *tokens* na transição T1, seguindo uma determinada taxa de chegadas. Os *tokens* então são enfileirados no lugar F, que, neste caso, não possui um limite máximo de enfileiramento. Para efetivamente “entrar” no sistema, o *token* deve passar por um tempo atrelado à transição T2. Esta entrada só ocorrerá se houver capacidade suficiente no lugar C. C representa a capacidade paralela máxima de atuação do sistema. Quando o *token* atinge o lugar P, então começa a contar o tempo da transição T3, finalizando assim um ciclo.

4.2. Modelo Proposto

A Figura 3 apresenta o modelo proposto. Este modelo tem como base o fluxo de transações da plataforma HLF definido em alto nível na Figura 1. Todas as transições no modelo seguem uma distribuição exponencial com tempos distintos entre si.

O fluxo de transações no modelo inicia por meio do disparo da transição **Arrival**, representada em cor cinza no canto superior esquerdo do modelo. Esta área destacada em um retângulo formado por linhas pontilhadas na cor verde destaca as atividades que ocorrem dentro do *gateway*.

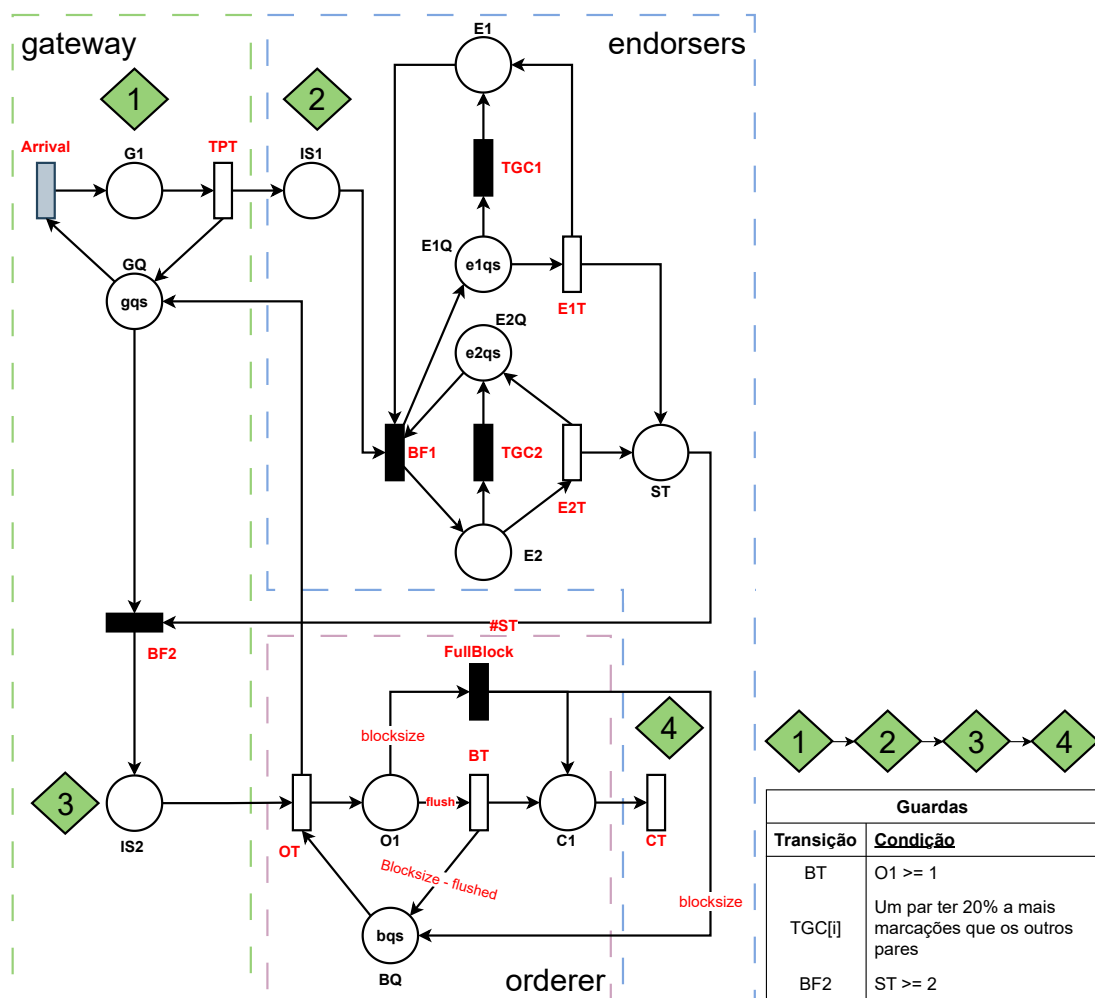


Figura 3. Hyperledger Fabric - Modelo de Desempenho da Plataforma HLF

As transações que chegam por meio de Arrival são direcionadas para o local **G1**, que simboliza o *gateway* no ambiente HLF, G1 está conectado ao local **GQ** (*gateway queue*) por meio da transição **TPT** (*transaction posted*). Transações que excedem a capacidade do *gateway* são descartadas. Transações que entram no *gateway* seguem para o estado intermediário **IS1** via transição temporizada **TPT**, e então direcionadas aos pares endossadores através da transição de bifurcação **BF1**.

Os pares endossadores e suas respectivas capacidades são representados pelos lugares **E1**, **E2**, **E1Q**, e **E2Q**. Um mecanismo de coletor de lixo, **TGC**, está presente em todos os pares endossadores e reduz a probabilidade de *starvation* das transações dentro do sistema e também o desperdício de recursos computacionais gerado pelo processo de envelhecimento de software.

Após o endosso, a transação chega ao estado **ST** (*Stamped Transaction*) por meio do tempo médio de endosso representado pelas transições **E1T** e **E2T**. Uma transação no ambiente modelado recebe endosso de dois pares, ou seja, o lugar **ST** terá até duas marcações que juntas representarão uma única transação. A transação atestada irá aguardar a disponibilidade do *gateway* para recebê-la (**BF2**) e então mover-se ao estado intermediário **IS2** por meio da transição de bifurcação **BF2** que indica que a transação está

entre o *gateway* e o ordenador.

A transição **OT** (*ordering time*) representa o tempo para ordenar transações em um novo bloco por meio do par ordenador (**O1**). O local **BQ** (*block queue*) indica o tamanho do bloco ou a quantidade máxima de transações necessárias para completá-lo [Silva et al. 2023].

Na HLF, um bloco pode ser parcialmente ou totalmente preenchido. Ao atingir a capacidade total de um bloco, as transações em **OQ** movem-se para o local **C1** através da transição **FullBlock**. O fluxo alternativo utilizado para um bloco parcialmente preenchido passa pelo local **BT** (*Batch Timeout*) que auxilia na prevenção de *starvation* estabelecendo o tempo máximo que uma transação estará na fase de ordenação.

A variável **flush** no arco que conecta o local **O1** à transição **BT** garante a transferência de transações para **BQ** sem exceder o tamanho do bloco (*blocksize*). Os blocos entram na fase de persistência por meio do lugar **C1** através da transição **CT**. Novos blocos são *commitados* para os demais pares da rede. Para maior precisão, expressões de guarda em **BF2**, **BT** e **TGCs** e são responsáveis por restringir o disparo destas transições, garantindo a eficiência do modelo e representatividade quando comparado ao ambiente real.

5. Metodologia Experimental e Validação do Modelo Base

Os experimentos para validar o modelo base foram realizados utilizando a rede básica do Hyperledger Fabric. Esta rede, mantida pelos desenvolvedores da HLF, permite a criação, verificação e transferência de ativos entre usuários. O foco dos experimentos recaiu, exclusivamente, sobre transações de criação de ativos, devido à sua maior demanda de recursos quando comparada a operações de consulta e transferência de bens entre contas.

A configuração do ambiente experimental incluiu um nó completo com dois pares endossadores e um par ordenador (*orderer*), implantados em contêineres Docker. A máquina de testes dispunha de quatro núcleos físicos, 8 GB de RAM e 80 GB de armazenamento. O software utilizado consistiu em Ubuntu 22.04, Hyperledger Fabric 2.5 e Docker 24. O ordenador, também era um contêiner Docker, e foi responsável por organizar transações em blocos e assegurar sua distribuição e persistência na rede. Uma interface de linha de comando (CLI) em um contêiner separado facilitou a interação entre a aplicação cliente e a rede *blockchain* através do contêiner que contendo o *gateway*.

No processo de validação do modelo proposto, a aplicação cliente, desenvolvida em TypeScript, enviou uma média de 10 transações por segundo (tps) para o *gateway* da plataforma. O cliente enviou as transações ao sistema seguindo uma distribuição exponencial, a mesma adotada para as transições temporizadas do modelo proposto.

O objetivo dos experimentos foi validar o modelo, e não otimizar o uso de recursos por meio de uma alta demanda que implicaria em uma sobrecarga nos pares. Porém, veremos em nossos estudos de caso que, mesmo com uma taxa de chegada baixa, muitos dos componentes estavam à beira da exaustão de recursos. Deste modo, a taxa de chegada das transações foi ajustada para assegurar a execução de todas as fases no modelo, incluindo tempos necessários a formação de blocos parciais e completos.

Medições foram realizadas no ambiente e os dados obtidos foram utilizados como insumos para avaliação do modelo proposto. A Figura 4 ilustra os resultados obtidos,

exibindo a latência a partir das medições e simulações. A latência é uma métrica derivada da Lei de Little [Jain 1991] que descreve a relação entre o número médio de transações pendentes em um sistema com a sua respectiva taxa de chegada.

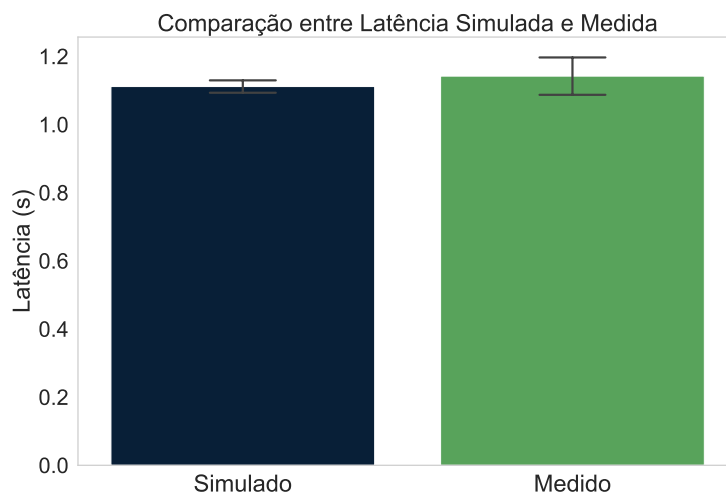


Figura 4. Comparação entre Latência Medida e Esperada

A avaliação experimental envolveu a submissão de 400 transações ao sistema real. Os tempos médios obtidos por cada etapa foram então utilizados para validar o modelo proposto por meio de 400 simulações. A ferramenta Mercury [Silva et al. 2015] foi utilizada para a realização das simulações que tiveram com a latência como métrica-chave. Os resultados mensurados indicam uma latência média de 1131 ms com desvio padrão de 408 ms. Já para os resultados simulados, obtivemos uma latência média de 1110 ms com desvio padrão de 18 ms.

Realizamos um teste t de duas amostras, que resultou em um valor p de 0,3. Com um nível de significância α de 0,05, o teste indicou que não existe diferença estatisticamente significativa entre os resultados obtidos por meio da avaliação do modelo e por meio de experimentos. Portanto, com 95% de confiança, não há evidências para rejeitar a hipótese de que o modelo base proposto representa o sistema.

6. Estudos de Caso

Após a validação do modelo proposto, concentramo-nos em cenários específicos e no estabelecimento de um conjunto de estudos de caso para avaliar métricas de desempenho críticas em ambientes de *blockchain* como latência, vazão e utilização. Além disso, avaliamos o impacto da aplicação de políticas de endossamento sobre o desempenho geral da plataforma e como o endosso geograficamente distribuído impacta o relacionamento entre diferentes organizações.

Os dados de referência para o modelo são as médias temporais de cada fator obtidas experimentalmente por meio de medição. A Tabela 1 resume esses valores, que servirão como parâmetros de entrada no modelo.

Tabela 1. Valores de Entrada para o Modelo e suas Variações

Fator	Valor Base	Intervalo de Variação {min,max}
Taxa de Chegada (λ)	18/s	{5, 30}
Tempo Limite de Bloco (BT)	2000 ms	{1000, 3000}
Tempo de Endosso (E1T e E2T)	160 ms	{80, 240}
Tempo de Ordenação (OT)	15ms	{7.5, 22.5}
Tempo de Commit (CT)	1150 ms	{575, 1725}
Tempo de Postagem de Transação (TPT)	10 ms	{5, 15}
Tamanho das Filas (E1Q, E2Q, GQ, BQ)	10	{10, 50}

6.1. Latência, Vazão e Utilização

A primeira métrica analisada é a latência, calculada para o modelo proposto conforme Equação (1). A Figura 5a evidencia como a taxa de chegada influencia a latência. Observa-se que o aumento na taxa de chegada resulta em um acréscimo na latência, que sai de 1110 ms para uma taxa de 10 tps para 20000 ms, para uma taxa de 100 tps.

$$E(\text{Place}) = \sum_{i=1}^n P(m(\text{Place}) = i) \times i, \text{ onde } P(m(\text{Place}) = i) \text{ é a probabilidade } i \text{ marcações no local} \quad (1)$$

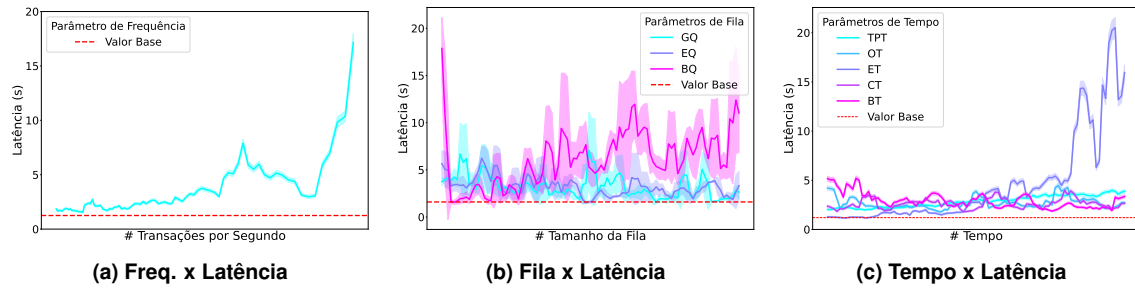


Figura 5. Impacto dos parâmetros sobre a Latência

Como o modelo foi avaliado por meio de simulação, adicionamos uma margem de erro a cada linha dos gráficos. O mesmo se aplica a Figura 5b, que demonstra o efeito dos parâmetros de fila sobre a latência. **A variação desses parâmetros (eixo x) se dá de acordo com o apresentado na Tabela 1.** O tamanho do bloco (BQ) se sobressai, blocos maiores podem resultar em atrasos no preenchimento e no processamento de novos blocos.

A Figura 5c destaca a influência dos parâmetros de tempo sobre a latência. O tempo de endosso (ET) apresenta o maior impacto. Os demais tempos, mesmo diante de variação acima ou abaixo, possuem um menor impacto sobre esta métrica.

A métrica de vazão mede o número de transações realizadas durante um intervalo de tempo [Thakkar et al. 2018]. Este artigo concentra-se no número de transações persistidas na *blockchain* por segundo. A vazão pode ser calculada por meio da Equação 2, com **E(Place)** representando transações em **O1** e **C1**, e $t(\text{Transition})$ indicando o tempo associado à transição **CT** (tempo de commit).

$$\text{Vazão} = \frac{E(\text{Place})}{t(\text{Transition})} \quad (2)$$

A Figura 6a mostra que a taxa de chegada mantém um impacto constante sobre a vazão do sistema. Já na Figura 6b vemos que o impacto negativo do tamanho do bloco (BQ) sobre a vazão é maior que o das demais filas do sistema e essa é fortalecida pelo tempo máximo até a completude de um bloco (BT), como podemos ver na Figura 6c, quanto maior o valor de BT, menor a vazão do sistema. **A variação desses parâmetros (eixo x) se dá de acordo com o apresentado na Tabela 1.**

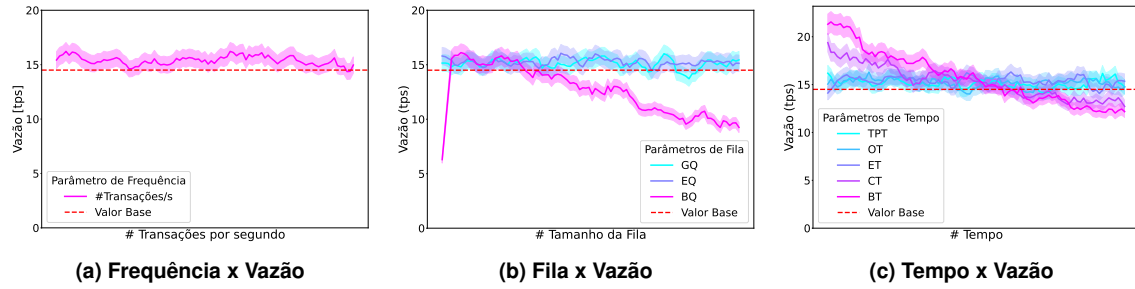


Figura 6. Impacto dos parâmetros sobre a vazão

Blocos menores resultam em uma maior vazão, algo que já havia sido comprovado em [Silva et al. 2023] e que agora é sustentado em um modelo refinado e em uma nova versão do Fabric que dispõe de um *gateway* e um conjunto de estados intermediários que aumentam a precisão e melhoram a acurácia dos resultados obtidos. Transações rapidamente processadas fluem de forma mais eficiente entre endosso, ordenação e commit. Um bloco de tamanho 15 ou maior, reduz significativamente a vazão. Além disso, em blocos muito grandes, um batch timeout de 2 segundos torna-se o único caminho possível para a geração e persistência de novos blocos.

Já a métrica de utilização quantifica o número esperado de marcações em um local específico **E(Place)** [Maciel et al. 2012], dividido pela capacidade total de outro local diretamente ao primeiro relacionado, por exemplo, o *gateway* (G1) e sua fila correspondente (GQ). Em modelagem, essa combinação de lugares caracteriza efetivamente uma fila de recursos, como mostra a Equação 3.

$$\text{Utilização} = \frac{E(\text{Place})}{\text{Capacidade de Recursos}} \quad (3)$$

Focamos na utilização do *gateway* que recebe requisições do cliente e endossadores. Como o *gateway* é um ponto único de falha, uma vez que há apenas um *gateway* no ambiente avaliado, é natural que esteja sobrecarregado. A Figura 7a mostra que o sistema se sobrecarrega com taxas de chegada abaixo de 5 transações por segundo e, ao alcançar uma taxa de 20 tps a utilização do *gateway* atinge 95%.

Este tipo de relação se mantém quando apontamos os fatores relacionados ao tempo e seu impacto sobre a métrica de interesse, como mostra a Figura 7c. O BT possui um impacto significativo: quanto menor o tempo para a formação do bloco, menor a utilização do sistema. Isso é um forte indicativo de que a ordenação impacta mais sobre o *gateway* do que a chegada de novas transações por parte do cliente.

Além disso, a Figura 7b confirma o impacto da ordenação sobre o *gateway* ao mostrar que blocos muito grandes (BQ) reduzem a utilização do sistema.

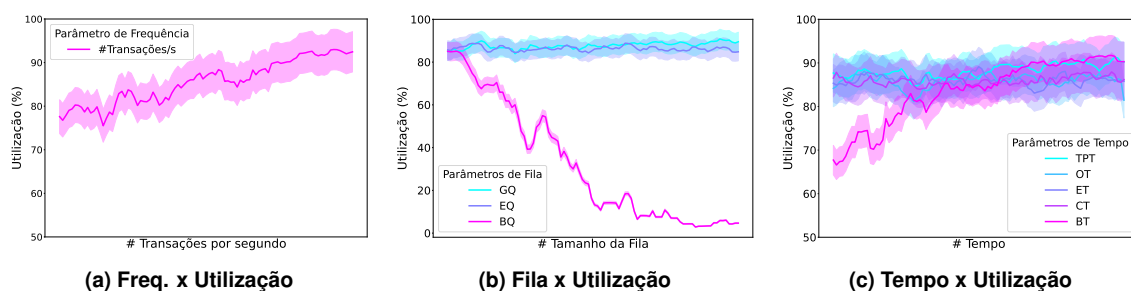


Figura 7. Impacto dos parâmetros sobre a utilização do gateway

6.2. Política de Endossamento

O endossamento de uma transação é baseado em um conjunto de políticas descritas no código fonte daquela *blockchain* ou em seu *smart contract* [Sukhwani et al. 2017]. As políticas especificam quantos e quais nós ou pares precisam concordar ou atestar, através de sua assinatura, a veracidade daquela transação. As três principais políticas de endossamento são **AND**, **OR** e **K-out-of-N** (KooN).

Supondo que tenhamos três pares endossadores, se a política de endossamento do tipo AND for adotada, então os três pares precisam atestar a transação; se do tipo OR, ao menos um dos três deve atestá-la; já do tipo KooN, determinamos quantos pares precisam atestá-la, por exemplo 2-out-of-3, nos indica que dois dos três precisam fazê-lo. Uma vez atestada, a transação volta ao *gateway* para que este a redirecione ao ordenador.

A Figura 8 mostra que a adoção de uma política de endossamento do tipo K-out-of-N proporciona uma maior estabilidade ao sistema em relação à latência e provê uma margem extra na disponibilidade do ambiente, uma vez que alguns nós endossadores podem falhar mas o serviço continuará em operação.

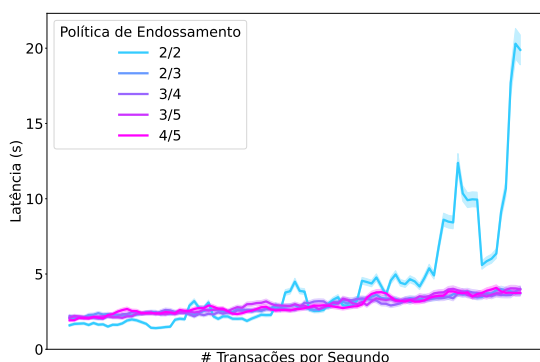


Figura 8. Latência x Política

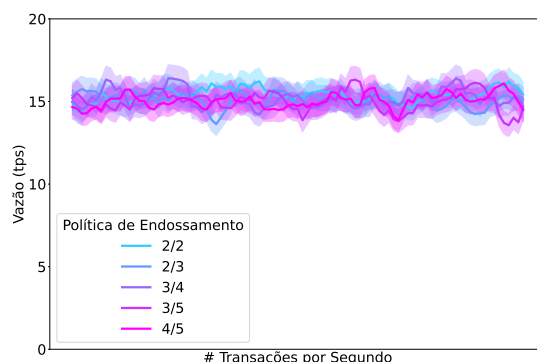


Figura 9. Vazão x Política

Já a Figura 9 mostra que a adoção de políticas de endossamento possui pouco impacto sobre a vazão geral do sistema para os cenários avaliados, mas, a longo prazo, conforme a latência geral do sistema cresce, a vazão tende a cair. Porém, é importante salientar que, o principal impacto proporcionado pela adoção de políticas de endossamento é sobre a segurança do ambiente e do relacionamento entre as organizações que o compõe.

6.3. Endosso Distribuído e Múltiplos Gateways

Para o último estudo de caso ampliamos o escopo do modelo proposto ao inserir um novo *gateway* visando a redução da sobrecarga (uso de recursos superior à 70%) e mitigação

dos impactos negativos sobre o nosso ponto único de falha na entrada do sistema. Além disso, consideramos três endossadores geograficamente distribuídos em uma política **2-out-of-3** por meio da adição de um tempo de latência ao processo de endosso.

O objetivo deste estudo de caso é avaliar como as métricas de interesse se comportam mediante alterações nos componentes que mais impactaram cada uma delas. Por exemplo, para a latência o tempo de endossamento e a taxa de chegada tiveram o maior impacto, enquanto que para vazão e utilização o tamanho do bloco mostrou-se como componente crítico. Utilizamos a Tabela 1 como base para este estudo de caso.

O tempo de endosso para cada par endossador corresponderá respectivamente ao tempo mensurado + latência de rede com base na distância entre *gateway* e endossadores. A ferramenta GCPing¹ foi utilizada e teve como ponto de partida a cidade de Garanhuns - Brasil, e como destino, grandes centros de dados da plataforma de computação em nuvem da Google (GCP).

As cidades-alvo, ou seja, os destinos com servidores viáveis escolhidos com base na menor distância em relação ao ponto de partida foram: São Paulo - Brasil, Santiago - Chile e Carolina do Sul - EUA. A respectiva latência média para cada servidor alvo foi de 81 ms, 136 ms e 197 ms.

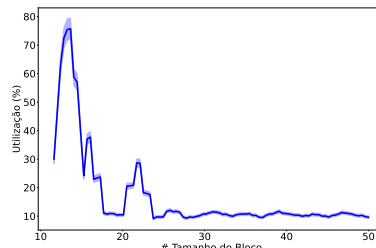
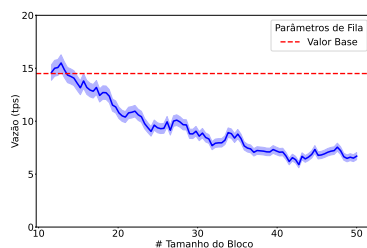
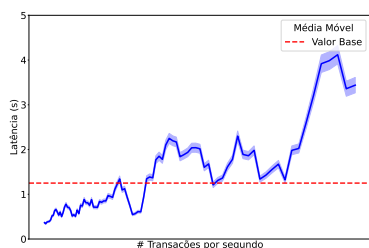
A Figura 10 apresenta o impacto da taxa de chegada sobre o tempo médio de resposta em um ambiente com dois gateways e três endossadores sob a política **2-out-of-3**. O comportamento anterior se mantém, ou seja, quanto maior a taxa de chegada, maior o tempo de resposta, porém os resultados indicam uma maior estabilidade, uma vez que, para o primeiro estudo de caso alcançamos uma latência de 20 segundos para uma taxa de chegada de 18 transações por segundo. Para o estudo de caso atual não passamos de 4,5 segundos de latência para a mesma taxa de chegada (18 transações por segundo), uma melhoria significativa e resultante, principalmente, da redução na sobrecarga dos gateways. É importante salientar que a política de escalonamento considerada para o *gateway* no âmbito do modelo refinado tem como base a mesma probabilidade (50%) de uma transação seguir por um *gateway* ou pelo outro.

Já a Figura 11 apresenta o impacto do tamanho do bloco sobre a vazão no mesmo ambiente. Quanto maior o tamanho do bloco, menor a vazão do sistema, uma vez que blocos grandes utilizarão do tempo para serem preenchidos e não de sua completude. Notamos o mesmo comportamento visto no primeiro estudo de caso, porém, a queda na vazão para blocos a partir de tamanho 40 foi maior, chegando a aproximadamente 6 transações por segundo, enquanto que no cenário do primeiro estudo de caso esse valor foi de aproximadamente 10 transações por segundo. Logo, o endosso distribuído impacta consideravelmente a vazão do sistema, em virtude da latência adicional associada ao tempo necessário para que as transações sejam atestadas.

Por fim, a Figura 12 apresenta o impacto do tamanho do bloco sobre a utilização do *gateway*. Quanto maior o tamanho do bloco, menor a utilização, como previamente observado no terceiro estudo de caso. Porém, a queda na utilização do *gateway* para patamares de aproximadamente 10% é antecipada para blocos a partir do tamanho 25 e no cenário anterior isso só acontecia a partir de blocos de tamanho 40. Com o sistema mais estável é possível aumentar significativamente a taxa de chegada e entregar um ambiente

¹<https://github.com/GoogleCloudPlatform/gcping>

com um melhor desempenho.



7. Conclusão

Este trabalho apresentou uma análise detalhada do desempenho do Hyperledger Fabric, empregando medição e modelagem para identificar gargalos sistêmicos. A ênfase foi nas métricas de latência, vazão (TPS) e utilização. Os experimentos focaram nas fases de endosso, ordenação e confirmação (commit) de transações na rede Fabric. Avaliamos também a utilização em função do tamanho do bloco e da fila de endosso, concluindo que o aprimoramento da capacidade de endosso pode elevar a vazão e reduzir a latência. Por fim, verificamos o impacto de uma política de endossamento sobre as métricas avaliadas. Uma infraestrutura com mais nós endossadores e nós distribuídos em diferentes locais sob uma política onde a maioria dos nós precisa atestar a transação afeta diretamente a latência e a vazão do sistema. Reconhecemos, como limitação, a necessidade de refinar as distribuições temporais utilizadas para simular as etapas do fluxo de transações, que, atualmente, se baseiam em distribuições exponenciais. Trabalhos futuros explorarão distribuições mais adequadas, considerando a expansão da rede em múltiplos nós e a latência. Pretendemos aprimorar o modelo para abordar um espectro mais amplo de métricas e aspectos da plataforma.

Agradecimentos

Este trabalho foi realizado com apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) sob o processo PDPG-POSDOC-AUXPE nº 88881.830176/2023-01. Quaisquer opiniões, descobertas, conclusões, ou recomendações apresentadas nesse material são dos autores e não necessariamente refletem a visão do CNPq.

Referências

- Buterin, V. et al. (2013). Ethereum white paper. *GitHub repository*, 1:22–23.
- Guggenberger, T., Sedlmeir, J., Fridgen, G., and Luckow, A. (2022). An in-depth investigation of the performance characteristics of hyperledger fabric. *Computers & Industrial Engineering*, 173:108716.
- Jain, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*, volume 1. Wiley New York.
- Jiang, L., Chang, X., Liu, Y., Mišić, J., and Mišić, V. B. (2020). Performance analysis of hyperledger fabric platform: A hierarchical model approach. *Peer-to-Peer Networking and Applications*, 13:1014–1025.

- Ke, Z. and Park, N. (2023). Performance modeling and analysis of hyperledger fabric. *Cluster Computing*, 26(5):2681–2699.
- Maciel, P. R., Trivedi, K. S., Matias, R., and Kim, D. S. (2012). Dependability modeling. In *Performance and dependability in service computing: concepts, techniques and research directions*, pages 53–97. IGI Global.
- Maciel, P. R. M. (2023). *Performance, reliability, and availability evaluation of computational systems, volume I: performance and background*. CRC Press.
- Melo, C., Araujo, J., Dantas, J., Pereira, P., and Maciel, P. (2022). A model-based approach for planning blockchain service provisioning. *Computing*, 104(2):315–337.
- Silva, B., Matos, R., Callou, G., Figueiredo, J., Oliveira, D., Ferreira, J., Dantas, J., Lobo, A., Alves, V., and Maciel, P. (2015). Mercury: An integrated environment for performance and dependability evaluation of general systems. In *Proceedings of industrial track at 45th dependable systems and networks conference, DSN*, pages 1–4.
- Silva, F. A., Gonçalves, G. D., Feitosa, L., Soares, A., et al. (2023). Avaliação de desempenho de blockchains permissionadas hyperledger orientada ao planejamento de capacidade de recursos computacionais. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 71–84. SBC.
- Suankaewmanee, K., Hoang, D. T., Niyato, D., Sawadsitang, S., Wang, P., and Han, Z. (2018). Performance analysis and application of mobile blockchain. In *2018 international conference on computing, networking and communications (ICNC)*, pages 642–646. IEEE.
- Sukhwani, H., Martínez, J. M., Chang, X., Trivedi, K. S., and Rindos, A. (2017). Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pages 253–255. IEEE.
- Sukhwani, H., Wang, N., Trivedi, K. S., and Rindos, A. (2018). Performance modeling of hyperledger fabric (permissioned blockchain network). In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pages 1–8. IEEE.
- Thakkar, P., Nathan, S., and Viswanathan, B. (2018). Performance benchmarking and optimizing hyperledger fabric blockchain platform. In *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 264–276. IEEE.
- Wu, O., Li, S., Liu, L., Zhang, H., Zhou, X., and Lu, Q. (2022). Performance modeling of hyperledger fabric 2.0. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering 2022*, pages 357–365.
- Xu, X., Sun, G., Luo, L., Cao, H., Yu, H., and Vasilakos, A. V. (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network. *Information Processing & Management*, 58(1):102436.
- Yuan, P., Zheng, K., Xiong, X., Zhang, K., and Lei, L. (2020). Performance modeling and analysis of a hyperledger-based system using gspn. *Computer Communications*, 153:117–124.