

Modelagem e Avaliação de Desempenho de Blockchain Permissionada baseada em Hyperledger Fabric

Carlos Melo¹, Francisco A. Silva¹, Glauber D. Gonçalves¹ e André Soares¹

¹Universidade Federal de Piauí – PI – Brasil

{casm, faps, ggoncalves, andre.soares}@ufpi.edu.br

Resumo. *O Hyperledger Fabric é uma plataforma aberta para a criação e manutenção de redes baseadas em blockchains permissionadas que permite o armazenamento e acesso a dados distribuídos de forma segura e auditável. Porém, para um melhor aproveitamento de suas capacidades, é de suma importância que estejamos atentos aos fatores de infraestrutura e arquitetura da rede. Este artigo propõe o uso de uma Rede de Petri Estocástica (SPN) para avaliar o desempenho da plataforma Hyperledger Fabric com diferentes parâmetros. Além disso, apresentamos um estudo de caso capaz de auxiliar os administradores de ambientes permissionados a adequar suas configurações, visando um melhor desempenho para suas aplicações. Através do modelo proposto, é possível identificar o tamanho de bloco ideal em relação ao tempo médio de resposta (MRT).*

Abstract. *Hyperledger Fabric is an open platform for creating and maintaining permissioned blockchain networks that allows for secure and auditable storage and access to distributed data. However, for better utilization of its capabilities, it is of utmost importance to pay attention to infrastructure factors and network architecture. This article proposes the use of a Stochastic Petri Net (SPN) to evaluate the performance of the Hyperledger Fabric platform with different parameters. Additionally, we present a case study capable of assisting administrators of permissioned environments in adjusting their configurations, aiming for better performance for their applications. Through the proposed model, it is possible to identify the ideal block size in relation to the mean response time (MRT).*

1. Introdução

As tecnologias baseadas em blockchain correspondem a um dos pilares da Indústria 4.0 [Smith and Lee 2022]. Além disso, as blockchains permitem que corporações com interesses similares compartilhem custos e recursos computacionais necessários ao provimento de uma aplicação ou serviço.

As redes blockchain já receberam o selo de aprovação em segurança por parte de muitas instituições financeiras¹, como Mercado Pago, PicPay, Méliuz, entre outras. No entanto, em termos de desempenho, elas ainda não foram suficientemente analisadas. Transações entre pares, sejam financeiras ou não, exigem baixa latência, semelhante às redes de cartões de crédito e ao Pix. Essa baixa latência não é observada em redes públicas, como Bitcoin e Ethereum [Sousa et al. 2021].

¹<https://www.infomoney.com.br/mercados/mercado-livre-anuncia-mercado-coin-criptomoeda-propria-negociada-no-marketplace/>

Redes permissionadas, como a Hyperledger Fabric, foram propostas como uma alternativa às grandes redes públicas [Silva et al. 2022]. Nelas, assim como em infraestruturas de computação em nuvem privadas, os provedores de serviço são responsáveis pela infraestrutura necessária. Além disso, todos os nós que compõem a rede são identificados, e os tamanhos dos blocos e o tempo limite para geração de novos blocos (*timeout*) podem ser personalizados.

Até o momento, vários autores propuseram avaliar a viabilidade da adoção de plataformas baseadas em blockchain como alternativas a outros serviços, como bancos de dados distribuídos. Em [Melo et al. 2022], os autores utilizaram Cadeias de Markov de Tempo Contínuo (CTMC) para avaliar a disponibilidade da plataforma Hyperledger Fabric.

Já em [Mendonça et al. 2023], os autores empregaram modelos de custo para determinar a relação entre custo e benefício para o provimento de uma infraestrutura capaz de hospedar uma aplicação baseada em blockchain que atenda aos requisitos mínimos de vazão e escalabilidade. Por fim, em [Silva et al. 2023], os autores avaliam o desempenho de uma rede blockchain, analisando o impacto que as variações no tamanho do bloco e no *timeout* têm sobre a vazão geral do sistema.

O presente trabalho tem como objetivo avaliar o desempenho de uma aplicação baseada em Hyperledger Fabric por meio de uma rede de Petri estocástica (SPN). O modelo proposto permite o cálculo de métricas como o tempo médio de resposta (MRT), a vazão (*throughput*), a taxa de utilização e a taxa de acionamento, com base no tamanho do bloco ou no *timeout*, bem como a taxa de descarte das transações.

2. Arquitetura e Modelo Proposto

A plataforma Hyperledger Fabric é um dos projetos de código aberto mantidos pela Linux Foundation para o desenvolvimento de aplicações baseadas em blockchain.² A abordagem utilizada pela plataforma consiste em um fluxo transacional de três etapas: endosso, ordenação e *commit*, ou seja, transações são persistidas em blocos.

Aplicações baseadas em Hyperledger Fabric são divididas em duas partes: cliente e servidor. O servidor pode ser descentralizado e distribuído em uma rede *peer-to-peer*. A comunicação entre ambas as partes ocorre por meio de transações. Quando o cliente submete uma transação, inicia-se o processo de *endosso*, onde a execução da transação é simulada. As transações endossadas são encaminhadas para a etapa de *ordenação*, onde são armazenadas em blocos. Quando os blocos são completos, seja por tempo ou tamanho, eles são persistidos por meio do *commit*.

Modelamos essas características em uma rede de Petri estocástica (SPN). Um tipo de grafo bipartido e direcionado frequentemente utilizado na descrição de sistemas. As redes de Petri são compostas por lugares, transições, arcos que os conectam, e marcações que indicam disponibilidade de recursos ou estados [Maciel et al. 2017]. A Figura 1 representa o modelo SPN proposto. Com base nesse modelo, é possível antecipar os efeitos de mudanças arquiteturais e seu respectivo impacto nas métricas relacionadas ao desempenho de aplicações baseadas em Hyperledger Fabric.

²Mais informações podem ser encontradas em <https://hyperledger-fabric.readthedocs.io>.

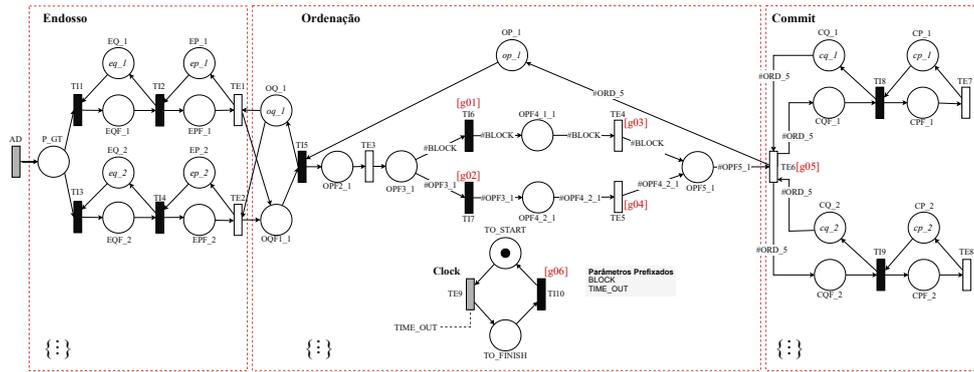


Figura 1. SPN - Hyperledger Fabric.

Para simplificar, o modelo proposto faz uma abstração da aplicação, adotando uma transição genérica de distribuição determinística que pode representar qualquer tipo de aplicação. Essa transição é representada na cor cinza e recebe o nome de (arrival delay - AD), que representa o intervalo de tempo entre transações.

O *endosso* se inicia quando uma marcação chega ao lugar P_GT. Em seguida, um entre dois computadores (acionados pelas transições T11 ou T13) é escolhido para processar a transação. Cada computador possui uma fila de entrada e uma fila de processamento, representadas pelas transições T12 e T14, respectivamente, que são imediatas. Os tempos de endosso são representados por TE1 e TE2 e dependem do número de unidades de processamento disponíveis em cada computador, o que é representado pelo número de marcações em EP_1 e EP_2. Quando não há recursos disponíveis, é criada uma fila para cada computador, representada pelos lugares EQ_1 e EQ_2.

Na etapa de *ordenação*, um único computador é responsável pelo processamento, a fila de entrada terá capacidade igual a oq_1 e a ordenação irá contar com a capacidade op_1. Uma transação entra em processo de ordenação quando uma marcação está presente no local OPF2_1. A transição TE3 indica um pré-processamento individual da requisição para torná-la apta a formar um bloco. Já o lugar OPF3_1 indica a construção de um bloco por meio do acúmulo de transações ordenadas. Diferentes tamanhos de bloco podem ser configurados por meio da variável #BLOCK; uma vez que o número de marcações em OPF3_1 seja igual ao valor de #BLOCK, um novo bloco é finalizado por meio da transição imediata T16.

O componente Clock indica ocorrência de *timeout* para a formação de um bloco. Dessa forma, um novo bloco é criado contendo apenas as transações atualmente presentes em OPF3_1. Em seguida, seguimos para a transição imediata T17, que é acionada apenas se #TO_FINISH = 0. Por fim, o bloco é persistido na rede por meio da etapa de *commit*, que é iniciada quando o bloco formado alcança a transição TE6. O commit é propagado para todos os computadores alocados para essa etapa e sua finalização é indicada por meio das transições TE7 e TE8.

2.1. Métricas

O tempo médio de resposta (*MRT*) pode ser obtido por meio da Lei de Little [Jain 1990], que relaciona o número médio de requisições em andamento (*RequestsInProgress*), a taxa de chegada (*Arrival_Rate*) e o *MRT*. Para o modelo proposto, a equação que

representa o cálculo do MRT é $\frac{RequestsInProgress}{ArrivalRate}$.

O número de requisições em progresso (*RequestsInProgress*) pode ser definido como a soma do número de marcações presentes em cada um dos lugares que representam uma transação em andamento. A esperança estatística de haver marcações em algum desses lugares é encontrada através da equação $Esp(Lugar) = (\sum_{i=1}^n P(m(Local) = i) \times i)$.

A **taxa de descarte** de transações pode ser definida como a relação entre indisponibilidade de recursos e enfileiramento na entrada do sistema, ou seja, pela ausência de marcações nos lugares *EQ_1* e *EQ_2*.

Já a **taxa de utilização** corresponde ao número de marcações esperadas em um lugar em relação à capacidade total do computador. Por exemplo, a utilização média do processamento na etapa de endosso para o computador com identificador 1 é $\frac{Esp(EP_1)}{ep_1}$.

Enquanto que **vazão do sistema** é representada pela taxa de saída para todas as transações que entraram no sistema e os que dele saíram através do *commit*. Já a vazão para cada computador pode ser encontrada pela relação entre o número de transações que passaram pelo dispositivo e o tempo de serviço esperado pela transição temporizada subsequente. Por exemplo, a vazão para o computador com identificador 1 é $\frac{Esp(CPF_1)}{t(TE7)}$.

3. Resultados

Para os parâmetros do modelo, utilizamos inicialmente os valores mostrados na Tabela 1 para capacidade de fila (eq, oq e cq), capacidades de processamento para marcações de endosso (ep), ordenação (op) e commit (cp) e tempos de serviço nas transições (TE).

Tabela 1. Parâmetros de entrada para o modelo proposto.

Tipo	Parâmetros	Valor
Capacidade	ep_1, ep_2, op_1, cp_1, cp_2	6
	eq_1, eq_2, oq_1, cq_1, cq_2	100
Tempo	TE1, TE2, TE3	5 ms
	TE6	10 ms
	TE7, TE8	80 ms
	TE4, TE5	2 ms

As marcações no modelo responsáveis pelo processamento no *commit* são respectivamente cp_1 e cp_2. Os valores para tais marcações foram variados da seguinte forma: cp_1 = cp_2 = [2, 4, 6]. Para obter uma visão macro, variamos também a taxa de chegada com valor mínimo de 0.0025 req/ms, e valor máximo de 0.3 req/ms e incremento de 0.01565 req/ms.

Para fins de análise utilizamos uma requisição por bloco, então configuramos um *timeout* alto (TIME_OUT=10000ms) e um tamanho de bloco baixo (BLOCK=1). Com tais valores, todo o fluxo do modelo passa pela formação de bloco completo.

A Figura 2 resume os resultados do primeiro estudo de caso. Os três gráficos de utilização (Figura 2a, 2b e 2c) possuem comportamentos semelhantes de crescimento em função do aumento da taxa de chegada. No entanto, a utilização do *commit* atinge o pico mais rapidamente em função da variação da capacidade computacional, com baixo impacto no que diz respeito a taxa de chegada. A taxa de descarte (Figura 2d) é calculada

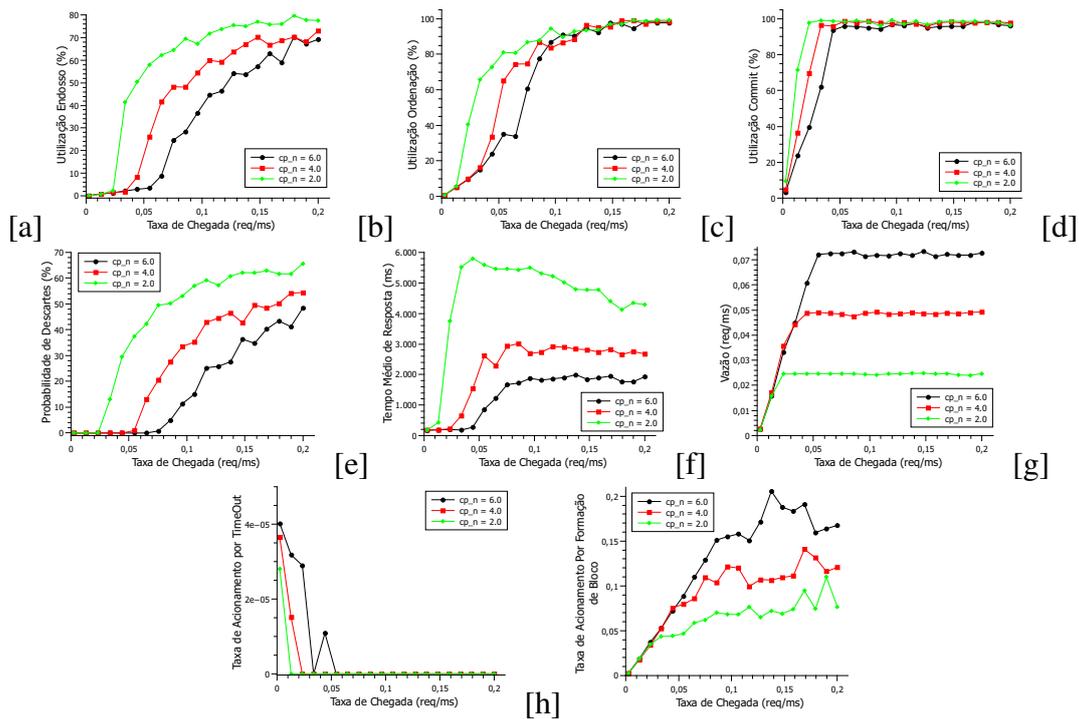


Figura 2. Variação da Capacidade de Commit

em função da utilização do ponto de entrada do sistema. Deste modo, as Figuras 2a e 2d apresentam comportamentos semelhantes. Caso o componente de endosso alcance alto nível de utilização, todos os componentes internos do sistema também estarão sobrecarregados e a taxa de descarte aumenta.

Para o MRT (Figura 2e), comparando $cp_n = 4$ e $cp_n = 6$, temos que para uma taxa de chegada 0.05 req/ms há um baixo impacto se comparada a sobrecarga alcançada quando o $cp_n = 2$. O número de requisições no sistema é tão alto que a taxa de chegada passa a ter um impacto maior sobre o MRT. A vazão (Figura 2f) cresce em um ritmo constante até estagnar. A estagnação ocorre em função da utilização do commit (Figura 2c). No momento que a utilização do commit atinge 100% de utilização não tem como aumentar mais. A taxa de aciornamento por timeout (Figura 2g) possui valores abaixo de zero, uma vez que um timeout de 10000ms força o redirecionamento do fluxo para o de formação de um bloco completo. A taxa de aciornamento por bloco (Figura 2h) seria proporcional à taxa de chegada se não existisse a restrição de recursos principalmente na etapa de *commit*. A taxa de aciornamento por bloco com $cp_n=2$ está abaixo das demais pois menos requisições passam.

4. Conclusão e Trabalhos Futuros

Neste trabalho, propomos um modelo SPN capaz de avaliar o desempenho de uma blockchain permissionada na plataforma Hyperledger Fabric. O modelo proposto calcula o tempo médio de resposta, vazão e utilização, bem como a probabilidade de descarte de requisições, levando em consideração diferentes parâmetros da blockchain, como o tamanho e o tempo limite de bloco. Além disso, o modelo considera variações na capacidade dos recursos de enfileiramento e processamento. Apresentamos um estudo de caso para demonstrar a viabilidade do modelo proposto. Nele, medimos o impacto das

configurações da blockchain (tamanho e tempo limite de blocos), além da capacidade computacional dos componentes arquiteturais que impactam o atraso e a vazão da rede.

A principal limitação do presente trabalho consiste no processo de validação do modelo proposto por meio do processo de experimentação. Tal atividade é um dos possíveis trabalhos futuros, que também incluem, mas não se limitam, à extensão do atual modelo para possibilitar a atribuição de pesos às decisões, por exemplo, para qual conjunto de máquinas participantes da rede blockchain atribuir uma determinada etapa do protocolo com maior probabilidade, e explorar tal funcionalidade como estudo de caso.

Outra importante extensão é a generalização do modelo para que o mesmo consiga representar n -nós sem a necessidade de replicação de lugares, arcos e transições, tornando-se importante como ferramenta de apoio a administradores. Por fim, estabelecer uma relação entre custo e benefício no que diz respeito ao provimento de serviços pode ser estabelecido.

Referências

- Jain, R. (1990). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons.
- Maciel, P., Matos, R., Silva, B., Figueiredo, J., Oliveira, D., Fé, I., Maciel, R., and Dantas, J. (2017). Mercury: Performance and dependability evaluation of systems with exponential, expolynomial, and general distributions. In *Proc. of PRDC*, pages 50–57. IEEE.
- 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.
- Mendonça, R. D., Moura, E. M., Gonçalves, G. D., Vieira, A. B., and Nacif, J. A. (2023). Comparação e análise de custo e desempenho entre nós de redes blockchain permissionadas e públicas. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. SBC.
- Silva, F. A., Brito, C., Araújo, G., Fé, I., Tyan, M., Lee, J.-W., Nguyen, T. A., and Maciel, P. R. M. (2022). Model-driven impact quantification of energy resource redundancy and server rejuvenation on the dependability of medical sensor networks in smart hospitals. *Sensors*, 22(4):1595.
- Silva, F. A., Gonçalves, G. D., Fé, I., Feitosa, L., and Soares, A. (2023). Avaliação de desempenho de blockchains permissionadas hyperledger orientada ao planejamento de capacidade de recursos computacionais. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*.
- Smith, J. and Lee, J. (2022). Blockchain applications in industry 4.0: A comprehensive review. *IEEE Transactions on Industrial Informatics*, 18(3):1652–1663.
- Sousa, J. E. d. A., Oliveira, V., Valadares, J., Dias Goncalves, G., Moraes Villela, S., Soares Bernardino, H., and Borges Vieira, A. (2021). An analysis of the fees and pending time correlation in ethereum. *International Journal of Network Management*.