Avaliação de desempenho do *Hyperledger Fabric* com banco de dados para o armazenamento de grandes volumes de dados médicos

Ana Caroline Fernandes Spengler[1], **Paulo Sergio Lopes de Souza**[1]

[1]Instituto de Ciências Matemática e de Computação – Universidade de São Paulo (USP)
São Carlos - SP – Brasil

ana.spengler@usp.br, pssouza@icmc.usp.br

Abstract. *Blockchain has been used as a distributed and secure infrastructure for storing and making data available. However, there are no more profound studies on the performance of blockchain storage and retrieval of heterogeneous data related to databases found in the literature. This paper presents the performance of blockchain with applications that demand heterogeneous data using a database. Our experimental studies consider using Hyperledger Fabric and Hyperledger Caliper, both acting on a heterogeneous medical database with CouchDB. Our results show that the performance of blockchain is influenced by different factors, such as the arrival rate of the requests, the operation performed (read or write), and the characteristics of the stored tables. Our results are helpful to developers of blockchain applications since they point out decisive aspects for the performance of such applications.*

Resumo. *O blockchain é utilizado para o armazenamento e disponibilização de dados de maneira distribuída e segura. Apesar desse uso, não são encontrados na literatura estudos detalhados sobre o desempenho do blockchain no armazenamento e na recuperação de dados heterogêneos associados a bancos de dados. Este artigo apresenta o desempenho do blockchain para aplicações que demandam dados heterogêneos com o uso de banco de dados. Os estudos experimentais executados consideram o uso do Hyperledger Fabric e Hyperledger Caliper, ambos atuando com uma base de dados médicos reais e heterogêneos com o CouchDB. Os resultados dos experimentos mostram que o desempenho do blockchain é influenciado por diferentes fatores, como a taxa de chegada das requisições, operação realizada (leitura ou escrita) e características das tabelas armazenadas. Nossos resultados auxiliam desenvolvedores de aplicações blockchain, pois apontam aspectos decisivos para o desempenho de tais aplicações.*

1. Introdução

Uso de *blockchain* em escopos diferentes de cripto moedas já é uma realidade [Gupta 2017]. Ele vem sendo utilizado como uma infraestrutura distribuída e segura para armazenamento e disponibilização de dados homogêneos e heterogêneos.

Um exemplo desse uso é a iniciativa da *IBM* anunciada em 2016 para facilitar transações entre bancos internacionais. O uso do *blockchain* neste caso permite que os participantes tenham acesso aos registros de forma global e mais eficiente, eliminando a necessidade de intermediários nessas transações [D'Alto 2017]. Outro exemplo são as

aplicações na educação, como a iniciativa do governo do Quênia em fazer uso do *blockchain* para o registro dos dados escolares do país. Neste último caso o *blockchain* fornece dados para análises sobre a distribuição de recursos entre as escolas, evasão escolar e onde os recursos disponíveis foram aplicados [Bore et al. 2017].

No contexto de aplicações médicas, existe uma demanda por aplicações capazes de proporcionar disponibilidade e segurança de dados críticos. Há trabalhos que apontam o potencial do uso do *blockchain* nesse cenário, como por exemplo [Esposito et al. 2018]. As aplicações de *blockchain* estão sendo usadas na área da saúde tanto para registro de dados médicos dos pacientes como para acompanhamento de consultas e procedimentos pelo hospital e planos de saúde [Hölbl et al. 2018].

Para facilitar o desenvolvimento e popularizar o uso do *blockchain*, propostas de implementações *Open-Source* foram desenvolvidas, como o *Hyperledger*. O *Hyperledger* provê uma interface de programação (API) para aplicações que facilite o desenvolvimento de aplicações com requisitos de interoperabilidade, escalabilidade e disponibilidade. O *Hyperledger Fabric* é a implementação *Hyperledger* mais utilizada atualmente, e é a plataforma escolhida para desenvolvimento neste estudo [Dhillon et al. 2017].

A literatura disponível mostra que há um interesse em pesquisas que avaliam o desempenho do *blockchain* considerando diferentes métricas, como latência e vazão. No entanto, há uma ausência de trabalhos que avaliem o uso de dados heterogêneos com o *blockchain*armazenando os dados em um banco de dados. Os requisitos de desempenho no contexto da área da saúde são fundamentais para garantir as demandas das aplicações.

Este artigo mostra resultados de uma pesquisa em andamento, sobre o desempenho do *blockchain*, quando ele é usado para armazenar dados heterogêneos associados a um banco de dados com informações de pacientes internados em um grande hospital.

Os estudos são baseados em resultados experimentais utilizando o *Hyperledger Fabric* [Hyperledger 2020] e o *benchmark Hyperlegder Caliper* [Project 2020]. Os dados usados nesse estudo estão disponíveis na base de dados *MIMIC-III* [Johnson et al. 2016] e foram armazenados no *CouchDB*. Os experimentos consideraram diferentes operações de escrita e leitura de dados, sobre tabelas de dados com características distintas, variando as taxas de envio de requisições, e a quantidade de transações.

Os nossos resultados mostram que o desempenho do *blockchain* é influenciado por diferentes fatores, como a taxa de chegada das requisições, tipo de operação realizada (leitura ou escrita) e características das tabelas armazenadas.

Este texto está organizado em sete seções. A Seção 2 apresenta os trabalhos relacionados ao estudo apresentado. As principais características do *blockchain* estão na Seção 3. A Seção 4 apresenta o planejamento dos experimentos realizados. Os principais resultados obtidos nos experimentos são apresentados na Seção 5 e na Seção 6 as nossas ponderações sobre tais resultados são apresentadas. Para finalizar o artigo, a Seção 7 contém nossas considerações finais sobre o trabalho desenvolvido.

2. Trabalhos relacionados

O artigo [Chung et al. 2019] demonstra como um *blockchain* pode ser otimizado para atender demandas de escalabilidade dessas aplicações para a indústria. Foi introduzida uma metodologia de avaliação empregada para avaliar uma aplicação desenvolvida com

o *Hyperledger Fabric*. Essa aplicação faz o compartilhamento de dados heterogêneos armazenados no *CouchDB*. Foram alterados parâmetros de configuração como, tamanho do bloco, número de nós validadores e etapas da transação, além de questões de infraestrutura, como número de servidores e CPUs disponíveis. Diferente deste trabalho, os autores não consideram o uso de uma base de dados reais como o *MIMIC-III*.

Em [Roehrs et al. 2019] também é apresentada uma avaliação de desempenho do *blockchain* para distribuição de dados médicos. Porém, diferente do nosso estudo, esse artigo mostra a avaliação de desempenho de uma implementação própria do *blockchain*, visando comparar a mesma com outras implementações similares. Já este estudo avalia o *Hyperledger Fabric* (uma plataforma para o desenvolvimento de aplicações *blockchain* de propósito geral) em um contexto diferente do que vinha sendo avaliado, como ilustrado nos trabalhos descritos a seguir.

O trabalho descrito em [Dinh et al. 2017] propõe o *benchmark BlockBench* para comparar a performance de distintas plataformas de desenvolvimento *blockchain* como *Hyperledger*, *Ethereum* e *Parity*. Desempenho foi caracterizado com base nas métricas vazão, latência e tolerância falhas, porém, os autores evidenciam a necessidade de uma posterior generalização dos seus resultados.

Em [Baliga et al. 2018] é proposta a caracterização do desempenho do *Hyperledger Fabric* em termos de vazão e latência com o uso do *Hyperledger Caliper*. Esse estudo modifica diferentes configurações da rede criada para verificar quais fatores apresentam maiores impactos no desempenho. Diferente do estudo apresentado, a análise realizada em [Baliga et al. 2018] não cobre aspectos sobre o tipo e o tamanho dos dados.

3. *Blockchain*

O *blockchain* é o protocolo criado junto com o *Bitcoin*. Esse protocolo é uma combinação de fatores tais quais as redes *Peer-to-Peer*, para a descentralização do sistema; uma cadeia pública que armazena os dados chamada de *blockchain*; o conjunto de regras de consenso para a validação das transações; e o mecanismo de autenticação global descentralizado para validar a cadeia de registros (algoritmo *Proof-of-Work*) [Antonopoulos 2017].

A estrutura básica do *blockchain* é o bloco, onde as transações estão armazenadas. Esses blocos, ligados em cadeia, estão distribuídos em uma rede *Peer-to-Peer* (*P2P*) na qual os participantes são chamados de nós. Em uma rede pública, como a do *Bitcoin*, não há restrições de capacidade e identidades dos nós [Antonopoulos 2017]. Existem também protocolos que utilizam de uma rede privada, em que há uma limitação na quantidade e na identidade de nós [Buterin 2015].

O processo de criação de um bloco é denominado mineração, sendo o mecanismo pelo qual os registros são validados. Esse nome deve-se ao fato do processo se assemelhar ao modo como outros produtos de valor, como ouro, são extraídos, devido ao esforço dedicado nesse trabalho. O problema computacional popularmente utilizado, e lançado em conjunto com o *Bitcoin*, é o *Proof-of-Work* (*PoW*).

Ao ser propagado pela rede, o novo bloco é inserido na cópia local da cadeia de todos os nós participantes da rede [Antonopoulos 2017]. Antes de inserir o bloco na cópia local, o nó realiza a verificação de cada transação armazenada pelo bloco. Após a execução dessa etapa, se todas as transações foram consideradas válidas, o nó adiciona o

bloco na sua cópia local do *blockchain*. Dessa forma, o processo de validação ocorre duas etapas, a validação das transações e do bloco. A validade de uma transação é determinada pela aplicação. Após os dados serem inseridos na cadeia eles são imutáveis.

A arquitetura do *Hyperledger Fabric* se diferencia do *Bitcoin* em alguns aspectos, tais como a composição da cadeia distribuída ser formada no *Fabric* por dois componentes de armazenamento: o *world state database* e o *blockchain*. O primeiro componente pode ser caracterizado como o estado da cadeia em um devido momento. Os dados são armazenados no formato chave-valor e podem estar nos bancos de dados *LevelDB* ou *CouchDB*. O segundo componente, o *blockchain*, armazena o registro de cada transação que modificou o valor do *world state*, como um registro global. No *Fabric* as transações são coletadas e arquivadas na cadeia de forma ordenada, garantindo assim a recuperação do valor do *world state* [Hyperledger 2020].

O *LevelDB* é um banco de dados não relacional e padrão do *Fabric*, sendo o ideal quando há somente um valor armazenado [Dean and Ghemawat 2020]. O *CouchDB* é um banco de dados não relacional orientado a documentos e usado quando o valor do par chave-valor é composto por múltiplos valores. Essa escolha se deve à estrutura do *CouchDB*, em que os dados são armazenados em documentos *JSON*, permitindo assim utilizar diversos tipos de dados [Foundation 2020].

Cada rede criada dentro do *Hyperledger Fabric* inclui um *Ordering Service*, uma abstração que tem a função de estabelecer qual é a ordem das transações. O nó *Orderer*, que implementa o *Ordering Service*, é essencial para manter a cadeia de cada nó da rede consistente e permite que os nós da rede foquem em validar e armazenar as transações na cadeia, sem necessidade de lidar com a ordenação das transações [Hyperledger 2020].

O nó *Orderer* tem como função o processo de coletar as transações propostas, ordená-las e agrupar nos blocos que serão distribuídos pela rede. No *Hyperledger Fabric*, os blocos criados pelo *Ordering Service* são irrefutáveis, ou seja, uma vez que a transação é escrita em um bloco, a sua posição na cadeia é imutável [Dhillon et al. 2017]. Essa característica diferencia o *Fabric* de outras implementações do blockchain, pois as transações são adicionadas ao bloco de maneira determinística.

Dentro do *Hyperledger Fabric*, há dois tipos de transação: *query* e *update*. A primeira é a mais simples, consistindo apenas da leitura de um valor armazenado no blockchain. Essa transação não precisa do processo de validação e consenso. Já a transação de *update* altera a cadeia, sendo necessária a validação dos demais participantes da rede. O processo de validação de uma transação de *update* dentro da rede do *Fabric* ocorre em três etapas [Hyperledger 2020].

Na primeira fase, os nós validadores (determinados pelo *smart-contract*) fornecem um parecer sobre a modificação proposta porém não aplicam a atualização à sua cópia da cadeia. Esses nós validam as transações de modo individual e retornam a resposta (se a transação é válida ou não). Quando há um número suficiente de respostas (configurado no *smart-contract*) essa fase termina e a transação pode ser descartada ou seguir para a próxima etapa [Hyperledger 2020].

Na segunda fase, as validações são agrupadas como transações e inseridas em um bloco. As transações aprovadas na primeira fase, são enviadas para o nó *Orderer*, que tem a função de ordenar as transações e criar o bloco que será distribuído na rede. O

estabelecimento da ordem das transações é necessário, pois é com base nela que os nós validam as transações e o bloco na etapa seguinte [Hyperledger 2020].

Na terceira fase, os blocos citados anteriormente são distribuídos para os nós da rede, onde cada transação é validada em cada nó (incluindo os nós não caracterizados como validadores) antes de ser aplicada à sua cópia local da cadeia. Cada nó valida as transações seguindo a ordem determinada na fase anterior de maneira independente. A ordenação das transações garante que os nós concordem ou não sobre as transações na mesma sequência, de forma que a cadeia se mantém a mesma nos diferentes nós da rede [Hyperledger 2020].

O conjunto de propostas do *Hyperledger* inclui o *Hyperledger Caliper* que é um conjunto de *benchmarks* que têm como objetivo definir um padrão para avaliar o desempenho de soluções *blockchain*. Atualmente, o *Hyperledger Caliper* permite obter resultados sobre a quantidade de transações aceitas na cadeia (*success rate*), vazão e latência [Project 2020]. De modo geral, o *Caliper* gera um *workload* para o sistema que se deseja avaliar, *system under test* (*SUT*), e continuamente monitora sua resposta. A partir desse resultado, o *Caliper* gera um relatório. O *framework* utiliza arquivos de configuração, onde são definidas a caracterização do *benchmark*, configuração da rede e carga de trabalho utilizada [Project 2020].

4. Descrição do experimento

O processo de planejamento, execução, análise e apresentação dos experimentos realizados neste estudo é fundamentado no trabalho de [Wohlin et al. 2012], visando permitir a reprodutibilidade e garantir o rigor na análise dos resultados obtidos. Informações complementares das configurações, bem como as tabelas com os resultados, estão disponíveis em https://github.com/anacspengler/WPerformanceExecution.git.

O objetivo deste experimento é avaliar quantitativamente o desempenho do *blockchain* com dados heterogêneos, armazenados em bases de dados. A plataforma *blockchain* escolhida para ser o objeto de avaliação foi o *Hyperledger Fabric* (versão 1.4.1). Esta versão tinha seu código estável e documentação disponível no período que os experimentos foram executados.

O foco qualitativo desse experimento é estabelecer o desempenho do *Hyperledger Fabric* com o uso de dados médicos encontrados na base de dados do *MIMIC-III* [Johnson et al. 2016]. A perspectiva desse experimento é quantificar o desempenho do *Hyperledger Fabric* quando aplicadas diferentes quantidades de transações (ou requisições) de inserção e leitura, com diferentes taxas de chegada (ou de envio) dessas transações e em tabelas com diferentes tamanhos e níveis de indexação. Esse experimento está no contexto de avaliar o *blockchain* como ferramenta de armazenamento distribuído para dados médicos heterogêneos.

4.1. Planejamento

As execuções serão realizadas na rede do laboratório de pesquisa (LaSDPC) usando dados médicos heterogêneos e reais. Os dados tabulados na base de dados do *MIMIC-III* [Johnson et al. 2016] foram coletados de pacientes clínicos que foram admitidos no Centro Médico *Beth Israel Deaconess Medical Center* em *Boston, Massachusetts*. As informações foram coletadas durante a rotina do hospital, incluindo arquivos de unidades

de terapia intensiva (UTI), dados retirados de registro eletrônico do hospital e o registro de óbitos da previdência social. Os dados do *MIMIC-III* são divididos em 26 tabelas.

Este estudo utilizou 4 das 26 tabelas disponíveis no *MIMIC-III* para as transações de escrita e leitura. As tabelas selecionadas permitem avaliar qual o impacto da quantidade de dados e das indexações no desempenho das inserções. A Tabela *PATIENTS* possui 8 campos distintos para cada entrada na tabela e não tem necessidade de indexar seus dados. Esta foi a Tabela utilizada para as transações de leitura (busca). A Tabela *D_ITEMS* contém 10 campos diferentes e requer duas indexações para as inserções. A Tabela *PRESCRIPTIONS* utiliza 19 campos para cada entrada e requer 5 indexações. Por fim, a Tabela *INPUTEVENTS_MV* contém o maior número de dados, com um total de 31 campos por entrada.

4.1.1. Formulação de hipóteses

Os experimentos visam comprovar três hipóteses relacionadas ao desempenho do *blockchain* associado a um banco de dados quando usado para acessos a dados heterogêneos.

A primeira hipótese é que a inserção no *blockchain* é mais custosa computacionalmente quando comparada às operações de busca, e isso afeta o desempenho na inclusão de grandes volumes de dados. A segunda hipótese é que a taxa de chegada das requisições e o total de requisições impactam o desempenho das transações no *blockchain*. A terceira hipótese é que a complexidade das tabelas nos bancos de dados, em termos de número de colunas e indexação, influencia no tempo de resposta das transações com o *blockchain*.

4.1.2. Projeto do experimento

Esta subseção descreve as métricas utilizadas para avaliar o desempenho, os fatores e níveis escolhidos, e a configuração da rede *blockchain*. As métricas de avaliação para esse experimento são latência e vazão. Para as transações *read* a latência é calculada considerando o tempo em que a resposta é recebida menos o tempo que ela foi submetida. Já a vazão é o total de transações executadas pelo tempo total em segundos. A latência das transações *update* é o tempo desde que as transações são submetidas até o resultado ser conhecido pela rede, o que engloba o tempo de propagação e o tempo da validação da transação de acordo com a politica de consenso. A vazão é a taxa em que as transações são submetidas em toda a rede pelo tempo total em segundos [Performance 2018].

Os fatores considerados nesse experimento são as quantidades de transações e a taxa de envio/chegada das transações ao *blockchain*. A quantidade de transações realizadas neste estudo é fruto do produto entre os cinco níveis de transações (1.000, 2.500, 5.000, 7.500 e 10.000 transações), os quatro níveis de taxa de envio das requisições (3, 5, 7, e 10) e, finalmente, as cinco operações: quatro inserções (em diferentes tabelas) e uma leitura na tabela *Patient*. Ao todo foram feitas 520.000 transações sobre o *blockchain*.

A faixa de valores para a quantidade de transações representa diferentes volumes de transações a serem processadas, entre 1.000 requisições para baixa carga e 10.000 requisições para alta carga de trabalho. Essa quantidade de transações, em rajada, é representativa para um hospital do porte do Hospital das Clínicas (HC) da Faculdade de Medi-

cina (FM) da USP, segundo dados de 2014 [de Imprensa 2014]. A faixa de valores para a taxa de envio das requisições foi baseada no volume já investigado para *blockchain* no contexto do *Bitcoin* que atinge vazão de sete transações por segundo [Croman et al. 2016].

Com o objetivo de padronização para a avaliação de desempenho do *blockchain*, o *Hyperledger* publicou um conjunto de termos básicos e principais métricas que devem ser usadas para avaliar um *blockchain* [Performance 2018].

O *Hyperledger Fabric* não apresenta um algoritmo de consenso como o *Bitcoin* com o *PoW*. Isso se deve à infraestrutura do *Fabric* que acrescenta o conceito de *ordering service* em que um nó da rede chamado *orderer* é responsável por ordenar as transações. Os blocos criados por esse nó são considerados válidos. Para este experimento foi utilizada a implementação *Solo* do *ordering service*, usando sua configuração padrão.

Todos os nós estão na mesma rede, *cluster* do *ICMC-Lasdpc*. Os *smart-contracts* foram desenvolvidos em *JavaScript*. A rede é composta por treze nós (todos no mesmo *channel*), em que quatro atuam como clientes, gerando a carga de trabalho (workload) através do *Hyperledger Caliper*, outros cinco atuam como nós na rede blockchain com o *Hyperledger Fabric* e quatro hospedam o *CouchDB*. Os cinco nós que dão suporte para a rede *blockchain* estão divididos em um nó *orderer*, responsável por ordenar as transações, e quatro nós divididos em duas organizações, simulando organizações do mundo real que desejam compartilhar informações (há dois nós para cada organização). Esses quatro nós das organizações também atuam como validadores de transações.

Os sistemas de *software* instalados para a execução deste experimento foram somente os pré-requisitos de instalação encontrados na documentação do *Hyperledger Fabric* e *Hyperledger Caliper*. Como o experimento foi realizado utilizando dados heterogêneos reais, os dados foram armazenados no banco de dados não-relacional *CouchDB*, o qual o *Fabric* dá suporte. Foram criados quatro nós *CouchDB* e quatro nós *Fabric*. Cada nó *CouchDB* foi associado a um nó *Fabric* pertencente à sua organização. Por meio da virtualização, o nó *Fabric* e o *CouchDB* estão na mesma máquina.

A carga de trabalho e a coleta das métricas de avaliação foram geradas pelo *Caliper*. As operações de busca e inserção de dados foram realizadas para avaliar as diferenças na latência e na vazão dos dois tipos de transações executadas pelo *Fabric*, a *read* e *update*, respectivamente. Para cada inserção, os dados são lidos das tabelas do *MIMIC-III*, distribuídos pela rede e armazenados no *CouchDB*. Durante o processo de distribuição das transações, elas também são validadas pelos nós antes de serem armazenadas na cadeia e no *CouchDB*. Esse processo de validação ocorre em três etapas, como descrita na seção 3. Na busca não ocorre o processo de validação, pois o nó mais próximo a quem fez a requisição é que responde com a informação solicitada.

Nesse experimento foram realizadas cinco operações, quatro inserções e uma busca. A escolha dessas operações foi baseada nas tabelas encontradas no *MIMIC-III*, a primeira inserção é da tabela *PATIENTS* a qual inclui informações com relação aos pacientes atendidos no hospital. A segunda inserção é da tabela *D_ITEMS* dicionário de items que estão em outras tabelas do *MIMIC-III*, com exceção aos que se referem a teste de laboratório. Finalizando, a próxima inserção é da tabela *PRESCRIPTIONS* que contém o registro de pedidos de medicamentos. A última inserção realizada é da tabela *INPUTEVENTS_MV* ingestão de fluidos por pacientes na UTI monitorados pelo sistema *iMDSoft*

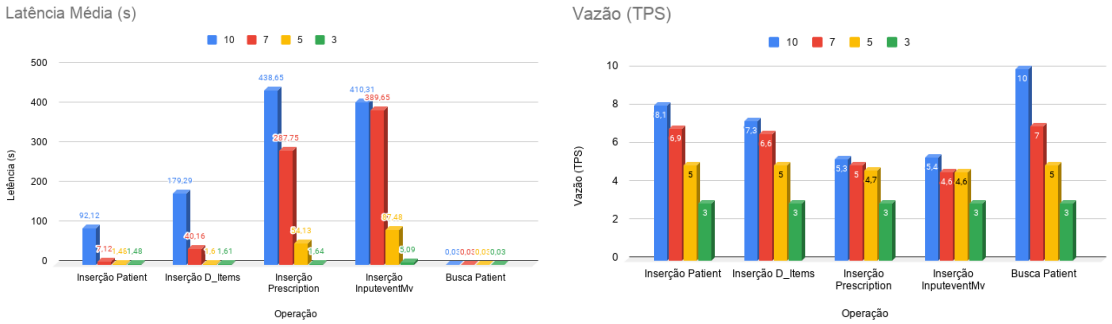
Metavision. A busca se baseia na tabela *PATIENTS*, em que é realizada uma busca na rede por informações de um determinado paciente de acordo com um *id* selecionado. O *id* do paciente é único para cada paciente admitido no hospital.

O ponto de observação utilizado nesse experimento foi configurado através do *Hyperledger Caliper*. Com o *Caliper* é possível configurar um monitor para coletar as informações sobre as transações e gerar os relatórios de desempenho. Além do relatório, também foi coletado o tempo de execução individual de cada transação com o objetivo de explorar medidas além da média.

As operações de inserção de dados são as primeiras a serem executadas, sendo a de leitura dependente da inserção da tabela paciente. As diferentes operações são executadas em blocos, onde uma tarefa só é inicializada após a finalização da anterior. A quantidade total de requisições assim como a quantidade de requisições a serem feitas por segundo (*send rate*) são parâmetros determinados na execução, porém a taxa real de requisições executadas por segundo (vazão) será limitada pela capacidade da infraestrutura.

5. Resultados

Esta seção apresenta os principais resultados obtidos com as execuções dos experimentos. Os dez gráficos apresentados nas Figuras 1 a 5 estão organizados primeiramente pela quantidade de transações efetuadas (10.000, 7.500, 5.000, 2.500 e 1.000) e então, para cada gráfico, são mostradas as quatro variações de taxa de envio (3, 5, 7 e 10). Cada Figura de 1 a 5 mostra os resultados para latência no gráfico (a), e a vazão no gráfico (b).



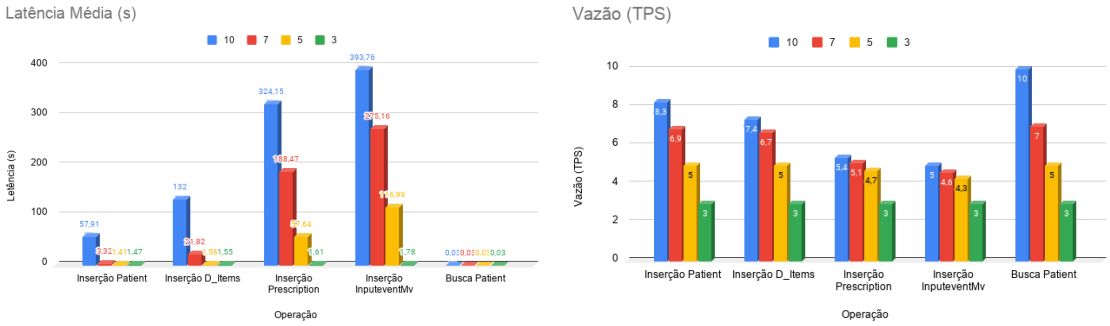
(a) Latência **(b) Vazão**

Figura 1. Resultados para 10.000 transações

As Figuras 1 a 5 mostram que a latência teve um comportamento similar para a variação das quantidades de transações e para a variação das taxas de envio, com uma diminuição das latências médias de escrita, conforme ambas (quantidade de transações e taxas de envios) diminuem. As latências observadas nas escritas sobre as diferentes tabelas, como esperado, mostram tempos médios maiores para tabelas com mais indexações e maior quantidade de campos, isto é, dados a serem manipulados a cada transação.

As latências observadas para a leitura dos dados permaneceram no mesmo patamar de valor (0,03s) em virtude da baixa quantidade de transações e de taxa de envio usadas nos experimentos, frente ao tempo de atendimento dessas requisições.

Considerando a vazão, observa-se que os resultados para menores taxas de envio (5 e 3) ficaram em vários casos limitados às respectivas taxas. A vazão não atingiu o teto

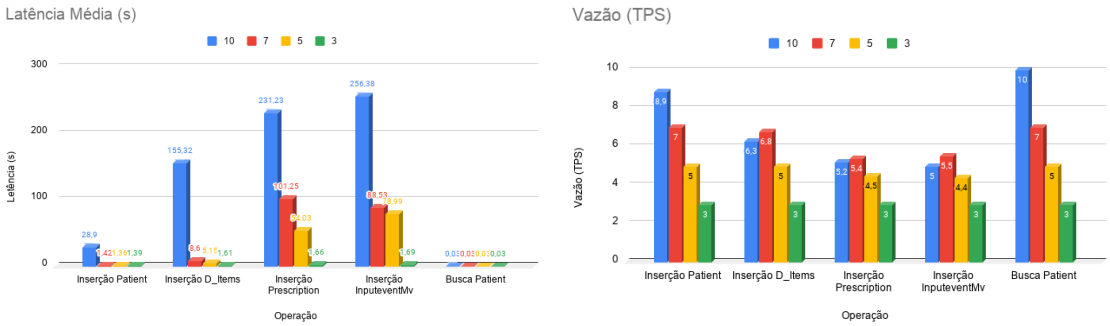


(a) Latência **(b) Vazão**

Figura 2. Resultados para 7.500 transações

da taxa de envio 5, nas escritas das Tabelas *PRESCRIPTIONS* e *INPUTEVENTS_MV* por requisitarem mais indexações e/ou manipularem mais dados. Porém, mesmo nesses casos obteve vazões altas de, no mínimo, 4.3 TPS.

Analisando-se as taxas maiores (10 e 7), observou-se que, de um modo geral, neste experimento as vazões aumentaram quando quantidades menores de transações foram usadas nas escritas de diferentes tabelas, indicando que possivelmente o processo de escrita é feito em etapas que se sobrepõem no tempo e concorrem pelos recursos da plataforma computacional. Há casos pontuais onde esse aumento não foi observado, como por exemplo na inserção em *D_ITEMS*, com taxa 10, e 7.500 e 5.000 transações. Porém, este não foi o caso geral dos experimentos realizados. As vazões das leituras foram limitadas pelas taxas de envio, pelos motivos já explicados.

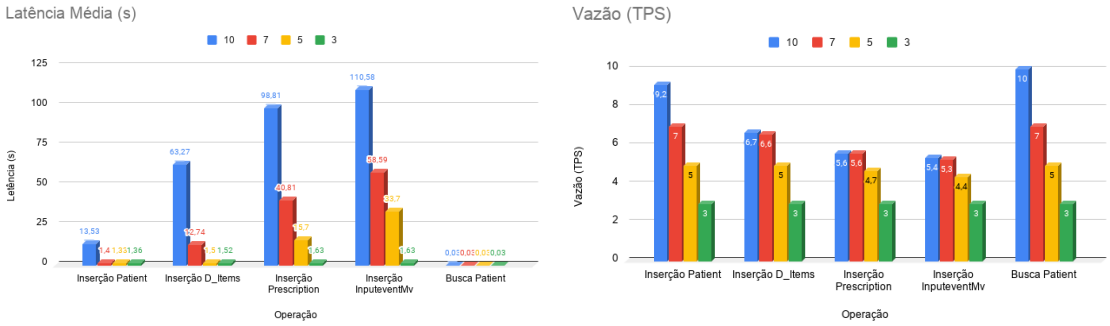


(a) Latência **(b) Vazão**

Figura 3. Resultados para 5.000 transações

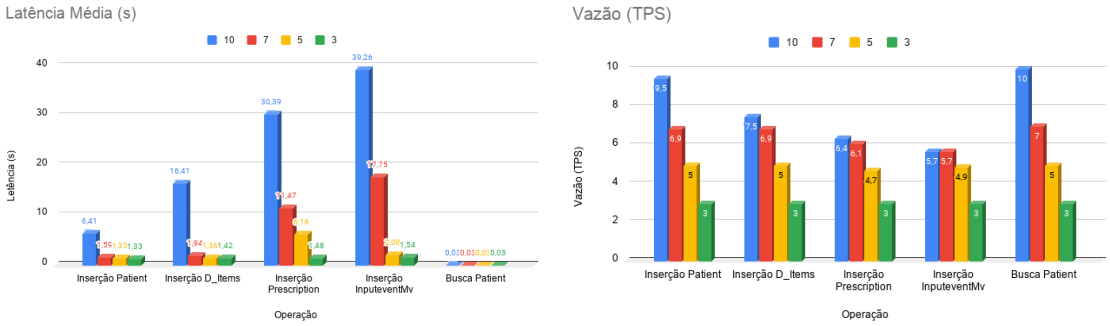
6. Discussões

A primeira hipótese formulada para os experimentos é sobre a comparação entre o desempenho, em termos de vazão e latência, das operações de inserção e busca, uma vez que as mesmas são caracterizadas como transações diferentes pelo *blockchain* de uma forma geral e, em particular pelo *Fabric*. Para todos os cenários avaliados neste experimento, o tempo de resposta das inserções mostrou ser significativamente maior que o da busca (veja Gráficos das Figuras 1a a 5a). Observa-se nesses gráficos que a latência das buscas



(a) Latência **(b) Vazão**

Figura 4. Resultados para 2.500 transações



(a) Latência **(b) Vazão**

Figura 5. Resultados para 1.000 transações

se manteve constante nos cenários avaliados (com uma média de 0.03s) e a vazão acompanha a taxa de envio das requisições. Isso indica que o atendimento das requisições de busca, nos experimentos realizados, estão limitadas pela taxa de envio de tais requisições para a rede, o que mantém a latência média constante. Por outro lado, para as inserções, diferentes tempos de latência e vazão foram obtidos em função de diferentes fatores como quantidade de transações e taxa de envio dessas.

A segunda hipótese refere-se ao impacto no desempenho da taxa de chegada das requisições e do total de requisições submetidas para processamento no *blockchain*. As Figuras 1 a 5 mostram que o *blockchain* tem diferentes desempenhos com esses fatores.

O desempenho das inserções nos experimentos foi mais afetado pela taxa de chegada das requisições. Com a mesma quantidade de transações, a latência das inserções pode variar 221% a depender da taxa escolhida para uma mesma quantidade de entradas, como foi o caso da inserção da Tabela *INPUTEVENTS_MV* para 7.500 entradas, observando a variação entre a latência com taxa de 10 e 3. Considerando apenas as variações das quantidades de transações, ainda para as inserções, nota-se que a diferença entre as latências é menor e tende a valores aproximados para taxas de requisição menores. É notável essa tendência na Tabela *PATIENTS* com taxa de chegada de 3 requisições, em que os valores variam entre 1,48 e 1,33.

Considerando as buscas, o cenário é ainda não conclusivo, pois nossos experimen-

tos não demonstraram variações, pelo fato de não termos encontrados os limites superiores de demanda que afetassem vazão e latência para buscas.

Analisando a terceira hipótese sobre o impacto do número de colunas (campos) das tabelas e suas indexações, percebe-se que os acessos à Tabela *PATIENTS* resultaram em um menor tempo de latência e uma maior vazão que as demais tabelas para as escritas, em virtude de não apresentar a necessidade de indexar seus dados. Quando comparada com a inserção da Tabela *D_ITEMS*, por exemplo, que contém duas indexações para o armazenamento no *CouchDB* e um número próximo de campos a cada inserção, é possível perceber o impacto dessa indexação no desempenho da transação.

Na maioria das execuções, a latência da inserção na Tabela *INPUTEVENTS_MV* é maior que em *PRESCRIPTIONS*, sugerindo que a quantidade de dados na Tabela influenciou mais no tempo de resposta que a indexação em si. Percebe-se que há uma diferença menor entre as vazões de *INPUTEVENTS_MV* e *PRESCRIPTIONS* nas diferentes quantidades de transações efetuadas, quando comparando tal diferença com a latência.

7. Considerações finais

Este artigo discutiu o desempenho do *blockchain em conjunto com um banco de dados* para armazenar dados heterogêneos. Foi usada a base de dados médicos *MIMIC-III* composta por tabelas com dados de pacientes [Johnson et al. 2016]. Os experimentos foram executados em uma rede *blockchain* com nove nós usando *Hyperledger Fabric*. Para gerar as requisições foi usado o *Hyperledger Caliper*.

Os resultados mostram que as escritas consomem muito mais recursos computacionais, e apresentam menores vazões e maiores latências, quando comparadas às leituras de dados. As taxas de chegada das requisições nos nesses experimentos, como esperado, afetaram a latência e a vazão da rede *blockchain* e impactaram fortemente o desempenho das operações de escrita. Essa demanda das inserções no *blockchain* deve ser considera por projetistas de aplicações distribuídas suportadas pela plataforma *blockchain*, pois elas comprometem o desempenho do sistema, principalmente sua escalabilidade. As características das tabelas também afetaram o desempenho nos nossos experimentos. Tabelas com grandes volumes de dados apresentaram maiores diferenças de latência e vazão que as tabelas que apresentaram maiores necessidades de indexação.

Os resultados produzidos neste artigo contêm informações que devem ser consideradas no projeto de novas aplicações *blockchain*. Como trabalhos futuros novos experimentos devem ser realizados estendendo os dados já apresentados. Tais experimentos devem considerar maiores taxas de envio e quantidades de transações que permitam estressar as operações de busca. Além disso, deve-se explorar o impacto de outros componentes do *blockchain* como aumentar o número de nós na rede, modificar o tamanho dos blocos e modificar a política de validação.

Referências

Antonopoulos, A. M. (2017). *Mastering Bitcoin: Programming the Open Blockchain*. O'Reilly Media, Inc., 2nd edition.

Baliga, A., Solanki, N., Verekar, S., Pednekar, A., Kamat, P., and Chatterjee, S. (2018). Performance characterization of hyperledger fabric. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 65–74.

Bore, N., Karumba, S., Mutahi, J., Darnell, S. S., Wayua, C., and Weldemariam, K. (2017). Towards blockchain-enabled school information hub.

Buterin, V. (2015). On public and private blockchains. Acessado em 02/07/2019.

Chung, G., Desrosiers, L., Gupta, M., Sutton, A., Venkatadri, K., Wong, O., and Zugic, G. (2019). Performance tuning and scaling enterprise blockchain applications.

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Gün Sirer, E., Song, D., and Wattenhofer, R. (2016). On scaling decentralized blockchains. volume 9604, pages 106–125.

D'Alto, L. (2017). Ibm announces major blockchain solution to speed global payments. Acessado em 02/07/2019.

de Imprensa, A. (2014). Hospital das clínicas de sp completa 70 anos em franca expansão. Acessado em 04/06/2021.

Dean, J. and Ghemawat, S. (2020). Leveldb. Acessado em 22/10/2020.

Dhillon, V., Metcalf, D., and Hooper, M. (2017). The hyperledger project. In *Blockchain enabled applications*, pages 139–149. Springer.

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1085–1100, New York, NY, USA. ACM.

Esposito, C., De Santis, A., Tortora, G., Chang, H., and Choo, K. R. (2018). Blockchain: A panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Computing*, 5(1):31–37.

Foundation, A. S. (2020). Apache couchdb. Acessado em 22/10/2020.

Gupta, M. (2017). *Blockchain for dummies: IBM Limited Edition*. Wiley, 1nd edition.

Hyperledger (2020). hyperledger-fabricdocs documentation. Acessado em 22/10/2020.

Hölbl, M., Kompara, M., Kamisalic, A., and Nemec Zlatolas, L. (2018). A systematic review of the use of blockchain in healthcare.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Sci Data 3*.

Performance, H.; Group, S. (2018). Hyperledger blockchain performance metrics white paper. Acessado em 12/04/2021.

Project, C. (2020). Hyperledger caliper documentation. Acessado em 22/10/2020.

Roehrs, A., da Costa, C. A., da Rosa Righi, R., da Silva, V. F., Goldim, J. R., and Schmidt, D. C. (2019). Analyzing the performance of a blockchain-based personal health record implementation. *Journal of Biomedical Informatics*, 92:103140.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*, pages 123–151.