Ferramenta de monitoramento gráfico para suporte à criação e testes de novos mecanismos de consenso em blockchains

Bryan Wolff
Diego Fernandes Gonçalves Martins
Marco Aurélio Amaral Henriques

[1]Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
13083-852 – Campinas, SP, Brasil

bryan.wolff@hotmail.com,{diegofgm,marco}@dca.fee.unicamp.br

Abstract. *The blockchain is a secure and distributed registration system that needs a consensus mechanism capable of defining which new blocks are inserted in the chain, as they are created by the network participants. The analysis of blockchain evolution controlled by a new consensus mechanism can be complex. To simplify such analysis this paper presents a new development tool for monitoring the blockchain dynamic evolution. The application architecture and its integration with a new consensus protocol called Probabilistic Proof-of-Stake (PPoS) is presented in detail.*

Resumo. *A blockchain é um sistema de registro seguro e distribuído que necessita de um mecanismo de consenso capaz de definir quais novos blocos são inseridos à cadeia, a medida que são produzidos pelos participantes da rede. A análise da evolução de uma blockchain controlada por um novo mecanismo de consenso pode ser complexa. Para simplificar tal análise, este artigo apresenta o desenvolvimento de uma nova ferramenta para monitoramento da evolução dinâmica de uma blockchain. É apresentada em detalhes a arquitetura da mesma e sua integração com o protocolo Probabilistic Proof-of-Stake (PPoS).*

1. Introdução

As blockchains públicas têm atraído muita atenção principalmente depois da popularização das criptomoedas. A blockchain pode ser vista como um livro razão distribuído entre todos os participantes da rede, onde as transações inseridas neste livro são imutáveis, após decorrido um tempo de confirmação [Greve et al. 2018]. Na blockchain existem diversos atores que contribuem para que novos blocos sejam confirmados e, desta forma, novas transferências de ativos podem ocorrer por meio das transações presentes nestes blocos. Para isto, toda blockchain conta com um mecanismo de consenso, o qual, segundo Bashir [Bashir 2017], é definido como um conjunto de passos que a maioria dos nós seguem para que seja possível concordar sobre um estado ou valor.

Dentre as propostas para mecanismos de consenso em blockchains, a mais conhecida é o *Proof-of-Work* (PoW) [Nakamoto 2009]. Nela o que importa é a capacidade do participante em resolver um desafio computacional. Milhares de participantes (nós) competem para resolver o desafio e aquele que resolver primeiro ganha uma recompensa e o

direto de criar o bloco. Todos os demais participantes têm seus esforços computacionais desperdiçados e iniciam novamente o processo para tentar ser o ganhador do próximo bloco, o que acaba contribuindo para problemas associados ao alto consumo de energia, pois o trabalho da maioria dos nós não é utilizado.

Além disso, aqueles nós com maior capacidade de investimento em hardware dedicado a mineração (Application Specific Integrated Circuits - ASIC) acabam com maiores chances de produzirem blocos, provocando a centralização do processo em torno de poucos participantes. Por estes motivos, foram propostos outros mecanismos de consenso, onde destaca-se o *Proof-of-Stake* (PoS) [King 2013].

Há diversos protocolos que utilizam o PoS e, em muitos destes mecanismos, a decisão de quem pode criar o próximo bloco é definida por meio de um sorteio, cuja probabilidade de um nó ser sorteado é influenciada pela quantidade de moedas que ele possui. É comum, portanto, que o *stake* seja a própria quantidade de moedas que o nó possui em um endereço de seu controle. Vasin [Vasin 2017] destaca que antes de se candidatar para gerar um bloco, um nó deve enviar moedas a si mesmo, para provar a propriedade das mesmas. O participante deve calcular o hash de prova, o qual é resultante do cálculo de uma função hash sobre elementos do bloco a ser criado. Uma vez calculado, o hash de prova é comparado com um objetivo definido pelo sistema de modo a refletir um certo grau de dificuldade. Caso o hash de prova seja menor que este objetivo, ocorre o sucesso no sorteio e o participante ganha o direito de criar o novo bloco e auferir os ganhos decorrentes de tal criação.

A categoria de mecanismos baseados em PoS diminui o gasto de energia em comparação com o PoW, já que não depende de poder computacional elevado. Por outro lado, o PoS permite aumentar a concentração de riquezas, já que quem possui mais moedas acaba tendo uma chance maior de ser escolhido para criar novos blocos, recebendo com isso novas recompensas. Com base nas ideias do PoS, o Departamento de Engenharia de Computação e Automação Industrial da FEEC - UNICAMP, por meio do grupo de pesquisa Research Group on Applied Security (ReGrAS), está desenvolvendo um novo protocolo para blockchains públicas chamado *Probabilistic Proof-of-Stake (PPoS)* baseado em rodadas de tempo discreto e confirmações probabilísticas.

Como a análise do comportamento de uma blockchain sob a perspectiva de um novo mecanismo de consenso pode ser complexa, torna-se necessário o uso de ferramentas que sejam capazes de fornecer uma visualização gráfica da evolução da blockchain, considerando as particularidades da mesma. Dessa forma, o objetivo deste trabalho é apresentar uma ferramenta gráfica para análise de blockchains, para auxiliar no desenvolvimento de mecanismos de consenso. É esperado que esta ferramenta possa ser implementada em qualquer mecanismo de consenso sendo necessário apenas uma adaptação na forma como a ferramenta gráfica coleta as informações da estrutura de dados de outros protocolos. Para fins de demonstração de suas funcionalidades, a aplicação foi integrada ao protocolo PPoS. Portanto, o restante deste trabalho descreve brevemente o PPoS, detalha a arquitetura da ferramenta e mostra como foi integrada à blockchain. Além disso, são mostrados alguns resultados por meio da visualização da interface gráfica da ferramenta.

2. O protocolo *Probabilistic Proof-of-Stake (PPoS)*

Proposto por Martins e Henriques [Martins and Henriques 2020], o PPoS em desenvolvimento é um novo mecanismo de consenso, onde a criação de blocos é definida por meio de sorteios criptográficos baseados em funções hash que ocorrem a cada rodada discreta com duração de T segundos. A cada rodada, os nós podem participar uma vez do sorteio para definir quem será o criador do próximo bloco da cadeia. Caso um nó não seja sorteado, ele deve aguardar uma próxima rodada para uma nova tentativa. Esta espera evita que aqueles nós com maiores recursos computacionais tenham vantagens significativas em relação a produção de blocos, como ocorre no PoW.

Quando são criados dois ou mais blocos sucessores de um mesmo bloco da blockchain, uma nova cadeia conhecida como *fork* é gerada. Para a obtenção do consenso, os mecanismos devem desestimular a criação de forks e também resolvê-los por meio de critérios que possam eliminar as cadeias que não possuem mais chances de crescer e se tornarem a cadeia principal (cadeia mais longa). No PPoS, os forks são controlados por meio da premissa de que um bloco é aceito somente se sua rodada de criação é menor ou no máximo igual à rodada de seu par de outra cadeia (bloco de mesmo índice), caso ele exista. Isso define um tempo máximo que uma cadeia menor que a principal em número de blocos (cadeia secundária) pode aguardar a chegada de um novo bloco, já que blocos com rodadas fora do intervalo de aceitação definido não são aceitos.

Neste sentido, um bloco com rodada de criação r é aceito apenas se essa rodada pertencer ao intervalo de tolerância definido $[r-tol; r+tol]$, onde tol é um número inteiro de rodadas. Como exemplo, se $tol = 1$ e a rodada local atual é r, o nó aceita blocos com rodada dentro do intervalo fechado $[r-1; r+1]$. Dessa forma, uma cadeia secundária é removida da visão local do nó sempre que ela não puder mais empatar com a principal, ou seja, sempre que seja necessário receber um bloco que já está fora do intervalo de tolerância. Mais detalhes sobre este novo protocolo de consenso podem ser obtidos no trabalho de Martins e Henriques [Martins and Henriques 2020] .

3. Desenvolvimento da ferramenta de monitoramento

Um mecanismo de consenso é um sistema distribuído, onde existe uma dificuldade inerente de visualizar a evolução de sua atividade, considerando todos os nós envolvidos. Esta dificuldade é mais evidente durante a fase inicial de desenvolvimento de novos protocolos de consenso, principalmente em cenários em que o conceito teórico do mecanismo é implementado de maneira distribuída. Dessa forma, uma ferramenta de monitoramento que possa permitir a visualização de todos os cenários de criação e eliminação de blocos pode contribuir significativamente para àqueles que desenvolvem novos mecanismos de consenso.

Esta seção detalha o desenvolvimento da ferramenta gráfica de monitoramento da evolução de uma blockchain controlada por algum novo mecanismo de consenso. O desenvolvimento tomou como base a utilização de algumas bibliotecas consolidadas, buscando facilitar a aceitação e a adoção por parte dos usuários finais.

3.1. Requisitos

O principal requisito da ferramenta é que a evolução da blockchain seja apresentada de forma gráfica em um ambiente interativo. Neste sentido, buscamos combinar

frameworks e bibliotecas que facilitem a apresentação dos dados em tempo real, utilizando ambientes de grande aceitação entre os usuários como, por exemplo, a arquitetura web, onde o usuário pode interagir com a aplicação através de um *web browser* comercial. Neste sentido, devido à sua capacidade de oferecer *frameworks* para o desenvolvimento *web* e sua flexibilidade em trabalhar com estrutura de dados complexas, utilizamos a linguagem Python. A partir da extração de toda a informação necessária do banco de dados da blockchain, existem várias bibliotecas Python que, quando combinadas, podem ajudar no atendimento dos requisitos, como apresentado nas próximas seções.

Outro ponto importante, é que devem ser extraídos dados relevantes da blockchain monitorada para que possam suportar atividades relacionadas ao processo de validação do próprio mecanismo de consenso utilizado. Além disso, a ferramenta deve suportar atividades de monitoramento no nível de usuário da própria blockchain, onde, neste caso, deve apresentar informações sobre quais blocos foram confirmados e quais transações estão presentes em cada bloco.

3.2. Bibliotecas e *frameworks* avaliados

Definidas as características principais do software, foram buscadas bibliotecas em Python que permitem a criação de gráficos, mais especificamente de gráficos direcionados (ou grafos). O objetivo é gerar informações de pontos, os quais podem ou não ser ligados por uma linha, o que é característico em um modelo de blockchain. Por exemplo, o NetworkX [Hagberg et al. 2008] é uma dessas bibliotecas, que permite a criação e manipulação de grafos 2D e 3D. Entretanto não é um pacote de desenho autossuficiente, gerando apenas as informações dos grafos, mas não os representando graficamente.

Sendo assim, esse software necessita de outras ferramentas que possam modelar seus grafos, como por exemplo, o Matplotlib [Hunter 2007] e o Plotly [Parmer et al. 2015]. Como a documentação do NetworkX trabalha em conjunto com o Matplotlib, inicialmente foi estabelecido o uso desta biblioteca para demonstrar o conteúdo gerado pelo NetworkX. Entretanto, mesmo conseguindo gerar um gráfico direcionado com sucesso, o Matplotlib possui poucos recursos de interações com o usuário, tornando estáticas várias informações visuais que precisavam de interatividade. Dessa forma, foi necessário avaliar novas alternativas que pudessem transformar a visualização estática em um modelo dinâmico, capaz de responder em tempo real às mudanças da blockchain.

Neste sentido, a biblioteca Plotly foi capaz de atender estas necessidades. O Plotly é uma biblioteca de visualização de dados para Python capaz de produzir gráficos dinâmicos, além de permitir que os mesmos sejam utilizados em aplicações Web. A biblioteca permite que uma grande quantidade de customizações sejam realizadas nos gráficos, incluindo mudanças na forma de apresentação o que torna o modelo adequado para visualizações de blockchains.

A partir da definição do NetworkX e Plotly como as ferramentas utilizadas para produzir a representação gráfica da blockchain, tornou-se necessário integrar a representação com um servidor *web*, capaz de responder as requisições de um cliente *web* (*browser*) e, assim, oferecer acesso ao sistema por meio de requisições HTTP/HTTPs. Para este fim, foi utilizado o framework Web chamado Dash [Hossain 2019] que, além de um servidor web, oferece recursos para criação de aplicações completas, com suporte

a linguagens como HTML, CSS e Java Script. O framework oferece muitos recursos gráficos editáveis que aumentam consideravelmente o nível de interação entre a aplicação e o usuário final. Além disso, o Dash é totalmente integrado à biblioteca gráfica Plotly, podendo facilmente apresentar os gráficos dinâmicos em sua interface Web e possui uma documentação completa e em constante atualização.

3.3. Arquitetura combinando: NetworkX, Plotly e Dash

Depois de definir todos os componentes utilizados para produzir a ferramenta de monitoramento, foi necessário implementá-la de modo a acessar os dados originais armazenados pela blockchain e, a partir deles, criar um gráfico direcionado e dinâmico que possa ser expresso na aplicação Web. Foi importante definir um layout no NetworkX, de forma a atribuir uma organização aos blocos e às linhas que os conectam, de maneira a produzir uma visualização confortável e o mais próxima possível de uma representação da blockchain.

A partir disso, foi necessário fornecer ao Plotly por meio do NetworkX as informações necessárias, isto é, os dados das coordenadas de cada nó e de suas arestas. Com isso, foi possível observar no Plotly uma figura interativa, com a possibilidade de utilizar ferramentas de zoom, de arraste e outras possíveis interações com o ponteiro do mouse. Essa figura pôde ser inserida na aplicação Dash, em um servidor hospedado localmente, apresentando a blockchain interativa em sua interface web. A sequência de passos desde o acesso ao banco de dados, onde os blocos da blockchain são armazenados, até a representação gráfica no ambiente Dash, é apresentada na Fig. 1.

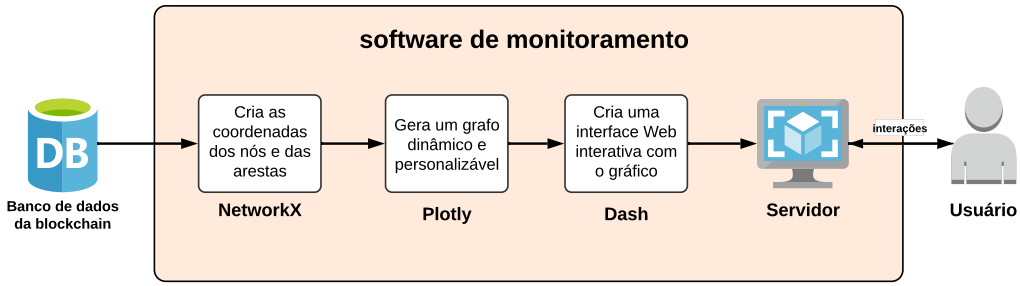

Figura 1. Fluxo de informações da ferramenta de monitoramento

3.4. Integração da ferramenta no mecanismo de consenso

Esta seção mostra como a ferramenta gráfica desenvolvida foi integrada com o mecanismo de consenso PPoS. O diagrama de sequência apresentado na Fig. 2 mostra as iterações temporais entre os diferentes objetos que compõem a interface gráfica e o mecanismo PPoS. Como pode ser observado na Fig. 2, o diagrama apresenta a interface gráfica como uma classe que tem um objeto instanciado a partir de um objeto da classe *Node* iniciado pelo administrador do nó. Ao definir a interface gráfica em uma classe própria, onde sua utilização é feita a partir de instâncias desta classe, a ferramenta não é fortemente vinculada ao código do nó do protocolo de consenso. Desta forma, essa abordagem contribui para sua reutilização em outros mecanismos de consenso, porém é necessário que sejam realizadas modificações na forma como os dados são coletados em

cada protocolo. Isso se dá devido às diferentes estruturas, nomeclatura e organização dos dados armazenados em cada mecanismo. Esta integração não é custosa, uma vez que basta identificar em cada novo projeto blockchain os dados básicos de formação do mesmo. Para os objetivos desta ferramenta, os dados do cabeçalho de cada bloco (normalmente armazenados em algum tipo de base de dados no nó da rede) são suficientes, já que não há um foco aqui em explorar as transações contidas em cada bloco.

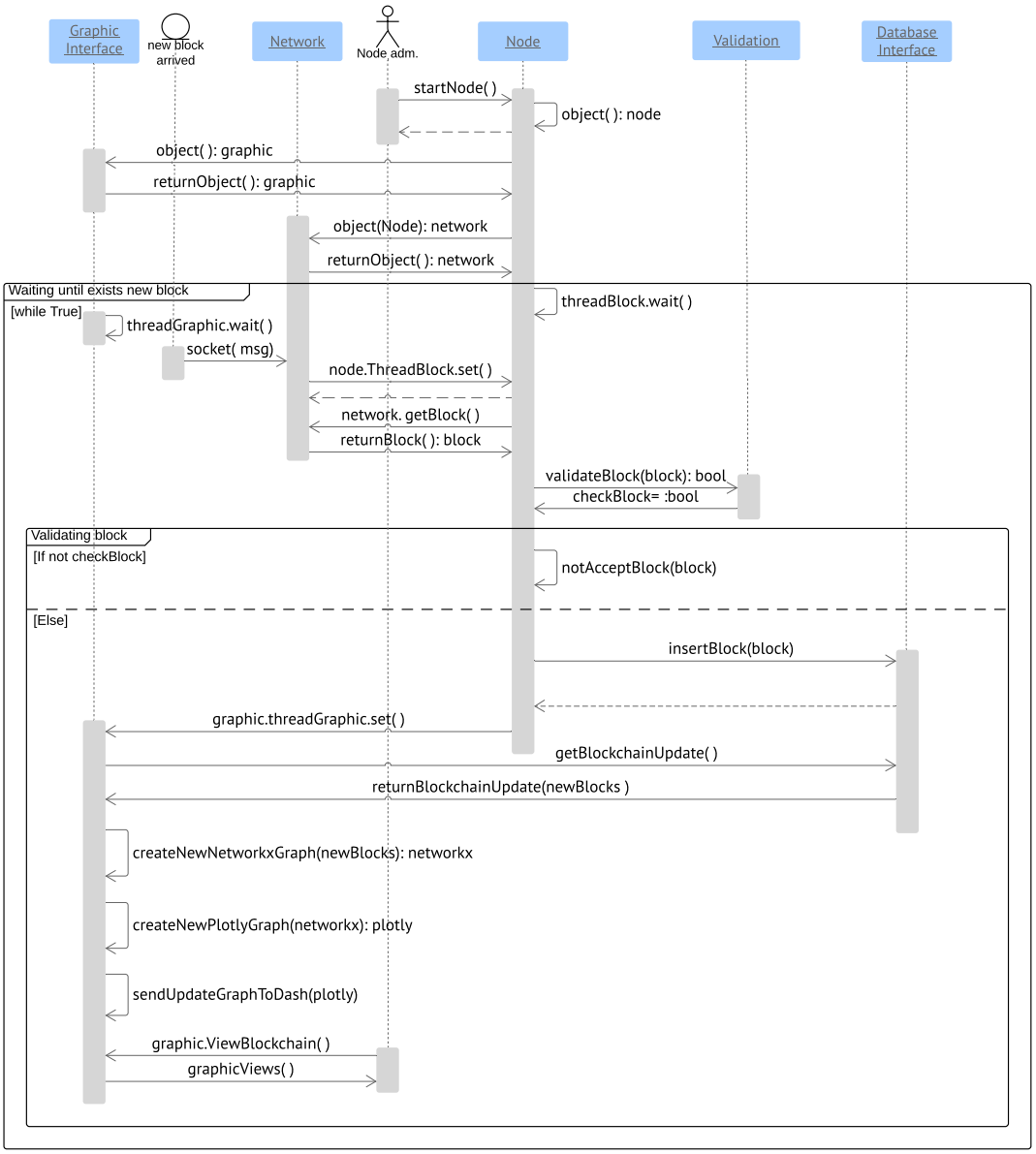

Figura 2. Diagrama de sequência entre o recebimento de um bloco e sua apresentação na interface

Quando o objeto *network* recebe um novo bloco através de seu *socket* conectado aos *peers* ele informa este evento ao objeto *node* por meio do método *node.ThreadBlock.set()* que por sua vez sai do estado de *wait* e inicia o processamento do novo bloco. Quando o nó recebe um bloco válido em uma determinada rodada, o mesmo informa este evento ao objeto da interface gráfica por meio do método

graphic.threadGraphic.set(). A thread da interface gráfica responsável por consultar novas informações no banco é colocada em execução. Dessa forma, a interface gráfica é executada sempre que novos blocos são inseridos no banco de dados pelo nó participante da blockchain. Além disso, as informações que são apresentadas na interface e as existentes no banco podem não estar sincronizadas, já que é possível que um novo bloco seja recebido pelo nó quando a interface já esteja executando operações no banco, ou seja, quando ela já não está no estado de *wait*. Neste cenário, novos eventos para que a interface consulte o banco de dados são perdidos. Ao perder estes eventos, a interface gráfica fica com os dados desatualizados, e, deixa de apresentar o bloco que foi inserido pelo nó durante sua execução anterior. Este estado é mantido até que um novo bloco seja recebido e a interface possa com isso consultar novamente o banco de dados a partir de um novo evento enviado pelo nó.

Para evitar este cenário, entre o recebimento do evento enviado pelo nó para a interface gráfica até o término do acesso ao banco de dados pode ser protegido por meio de um semáforo, evitando assim que outros eventos de atualização do gráfico com novos dados do banco sejam perdidos. Isso garante que após a atualização do banco, outro evento invocando a interface grafica será disparado quando a mesma estiver no estado de *wait* novamente e, assim, o mesmo nao será perdido evitando que a interface fique desatualizada.

Por fim, o usuário representado no diagrama pelo ator *Node adm* pode interagir com a interface gráfica por meio de métodos que permitam a visualização da mesma em seu web *browser*. Como trata-se de uma arquitetura *web*, qualquer outro ator com acesso ao endereço do servidor também pode acessar a aplicação.

4. Trabalhos Relacionados

No cenário de monitoramento de blockchains, a ferramenta *Hyperledger Explorer* possui um grande destaque na literatura. É uma ferramenta criada pelo projeto colaborativo *Hyperledger* [Dhillon et al. 2017] da organização *Linux Foundation* envolvendo várias empresas com o propósito de criar uma plataforma onde qualquer um que esteja interessado em desenvolver um software baseado em blockchain possa ultilizá-lo.

A ferramenta *Hyperledger Explorer* é um aplicativo focado na exploração de uma blockchain corporativa. É uma aplicação *web* que fornece um painel para visualizar várias informações sobre os blocos da blockchain como transações, estatísticas, informação da rede e vários outros itens relevantes para a análise de seu desempenho. Os seus desenvolvedores tinham como objetivo serem capazes de encontrar e visualizar as informações da blockchain e interagir facilmente com as mesmas, consolidando várias ferramentas de análise em uma única interface de usuário amigável.

Além disso, existe a ferramenta *Blockchain Explorer* [Reeves 2011] para blockchains públicas, destinada a análise de informações relacionadas ao número médio de transações, preço dos ativos, transações mais recentes e últimos blocos criados das criptomoedas mais populares, tais como: Bitcoin, Ethereum e Bitcoin Cash, por exemplo.

Apesar de *Hyperledger Explorer* e *Blockchain Explorer* possuírem uma boa capacidade de monitorar os principais conteúdos de uma blockchain sob a ótica dos usuários (foco em transações principalmente), elas não são uma opção muito adequada para aqueles que estão desenvolvendo novas blockchains e precisam acompanhar de forma rápida e

eficiente o desenvolvimento das mesmas de acordo com os mecanismos de controle definidos. Um dos grandes desafios para o desenvolvedor é a detecção e o acompanhamento de cadeias secundárias (forks), em tempo real, de forma a verificar se os mecanismos de consenso e de controle estão sendo eficazes para minimizar o surgimento e, se surgirem, para extinguir tais forks. É para este tipo de aplicação que foi desenvolvida a ferramenta aqui apresentada. Por meio da combinação de recursos gráficos e textuais e com o auxílio da interatividade oferecida, esta ferramenta permite aos desenvolvedores de novos tipos de blockchain acompanhar a evolução da cadeia e avaliar mais amigávelmente e precisamente o comportamento da mesma de acordo com as regras que estão sendo definidas no projeto.

5. Resultados

A partir da arquitetura definida, foi possível produzir a primeira versão da inferface gráfica da ferramenta integrada ao mecanismo de consenso PPoS. O software está disponível no GitHub[1] e os resultados demonstrados nesta seção foram obtidos através da execução desta aplicação em uma máquina virtual com sistema operacional Linux Ubuntu 18.04 com 4Gb de memória RAM e um processsador Intel i5-4590. O software deve ser integrado ao código da blockchain conforme as instruções contidas no repositório GitHub. Além disso, a aplicação poderá ser executada em cada um dos nós que executam o mecanismo de consenso, permitindo assim visualizar a evolução da blockchain na visão de cada nó e, consequentemente, detectar desvios de comportamento locais que possam ocorrer durante tal evolução. Dessa forma, a representação gráfica da ferramenta em um navegador *web* é apresentada na Fig. 3.

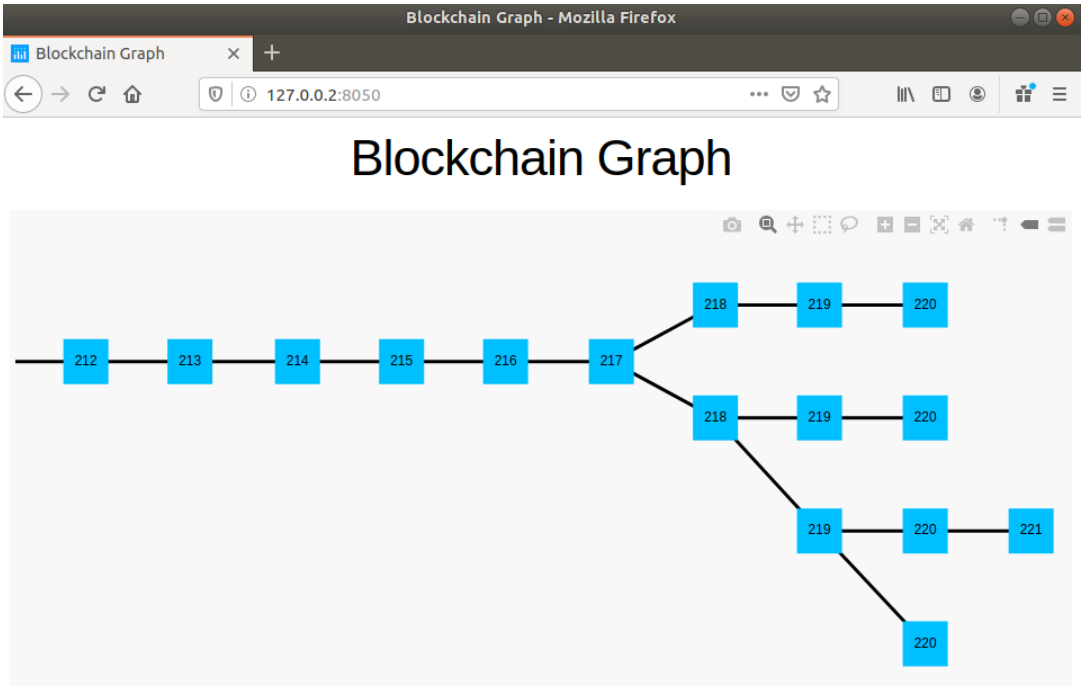

Figura 3. Representação da blockchain para um determinado nó da rede usando interface *web*

[1] https://github.com/regras/bcgui

Nota-se que a representação mostra a blockchain com características que permitem identificar claramente a sequência de blocos representada pelos retângulos, onde dentro de cada um deles é apresentado o índice do bloco. A formação de *forks* é mostrada a partir de bifurcações geradas após alguns blocos, o que representa, o surgimento de novas cadeias que futuramente serão eliminadas pelos critérios estabelecidos no mecanismo de consenso.

Para ilustrar de maneira mais detalhada como a ferramenta gráfica apresenta os *forks* à medida que eles ocorrem, a Fig. 4 mostra um cenário, onde a princípio existem três cadeias concorrendo entre si. Além disso, é considerado que os blocos foram produzidos em rodadas consecutivas e a rodada atual é a de criação do bloco 221. De acordo com a Fig. 4, existem duas cadeias empatadas em tamanho e uma terceira cadeia menor, que ainda pode empatar com as outras. Neste sentido, a ferramenta apresenta as três cadeias com arestas contínuas e blocos com cores sólidas, indicando que todas são válidas e que, portanto, os *forks* produzidos por elas ainda não foram resolvidos pelo mecanismo de consenso da blockchain.

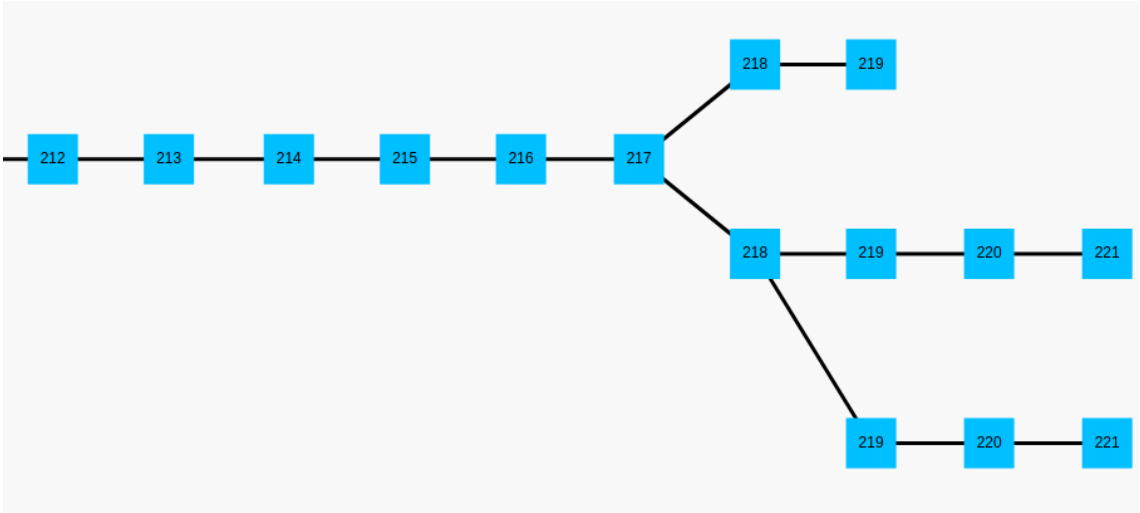

Figura 4. Cenário 1: Blockchain com três *forks* não resolvidos

A Fig. 5 considera que a cadeia menor não recebeu nenhum outro bloco durante a rodada de criação do bloco 221 e, assim, o protocolo eliminou esse *fork*. Dessa forma, para demonstrar que esse *fork* foi resolvido pelo protocolo de consenso, a Fig. 5 mostra essa cadeia com uma aresta pontilhada.

Na Fig. 6, o objeto *network* recebe dois novos blocos válidos (222 e 223), sucessores do 221 que são inseridos na cadeia inferior. A cadeia central, que neste momento é a menor, ainda pode empatar com a inferior caso receba mais blocos. Entretanto, a Fig. 7 ilustra que, como nenhum bloco foi recebido, a cadeia central é eliminada. Para demonstrar que esse *fork* foi resolvido, a interface ilustra essa ocorrência a partir de uma cadeia com uma aresta pontilhada.

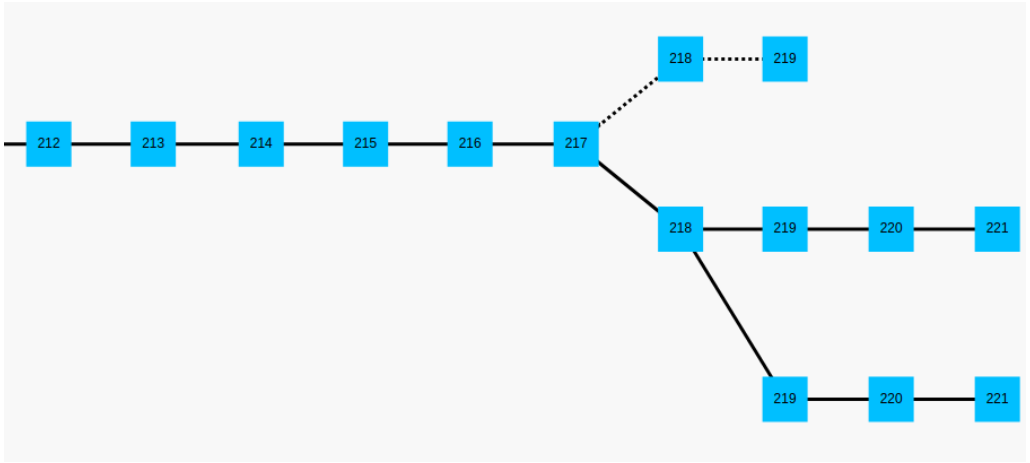

Figura 5. Cenário 2: ferramenta continua mostrando fork resolvido para análises posteriores

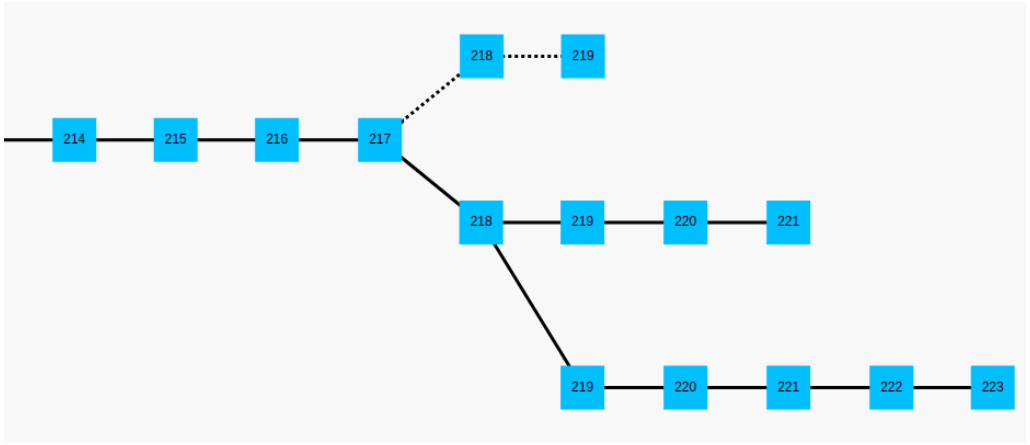

Figura 6. Cenário 3: O nó recebeu dois novos blocos válidos (222 e 223) que são inseridos na cadeia inferior do grafo

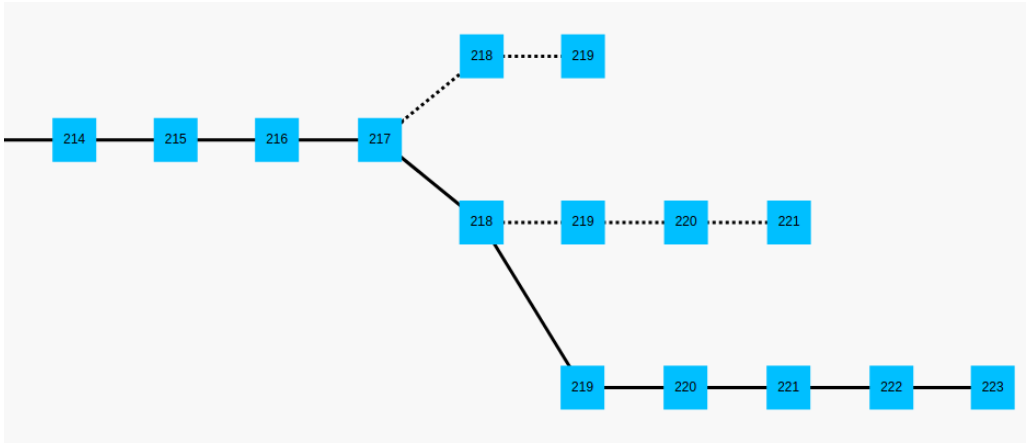

Figura 7. Cenário 4: *fork* central resolvido

Por meio da sequência de imagens mostradas acima é possível notar que esta ferramenta mostra de maneira contínua a evolução da blockchain de acordo com as regras

que foram definidas para a criação de novos blocos. Outro ponto importante, é que as cadeias secundárias são mantidas na visão da interface gráfica para fins de análise post mortem do mecanismo de consenso.

Além disso, a ferramenta oferece diferentes formas de interação com o usuário, onde são permitidas diversas atividades como, por exemplo, visualizar informações referentes a um bloco do PPoS, conforme mostrado no exemplo da Fig. 8.

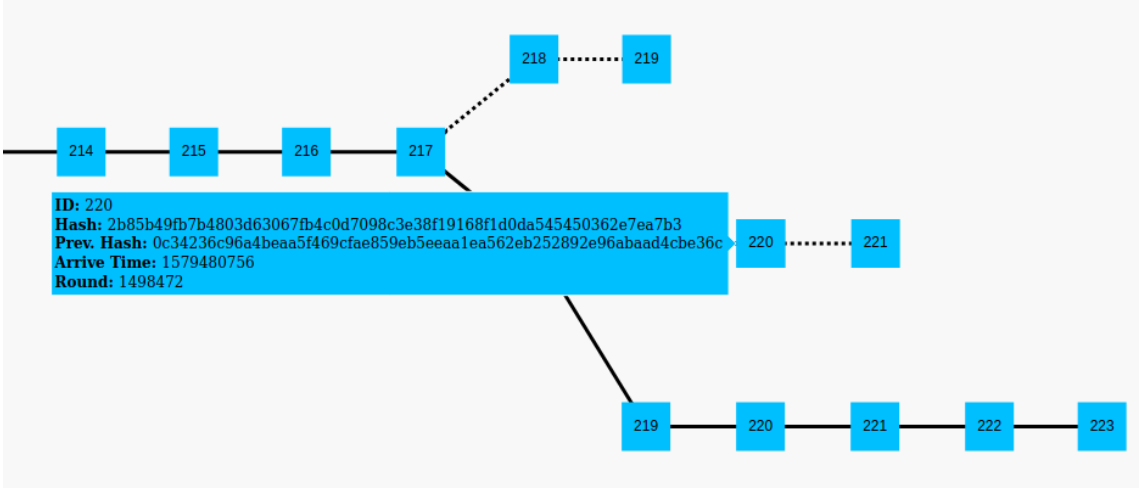

Figura 8. Exemplo de interatividade: ao passar o mouse sobre um bloco, são apresentadas informações contidas no mesmo

Também é possível utilizar recursos de zoom e, para ilustrar isso, a Fig. 9 demonstra um zoom aplicado na cadeia principal da Fig. 7.

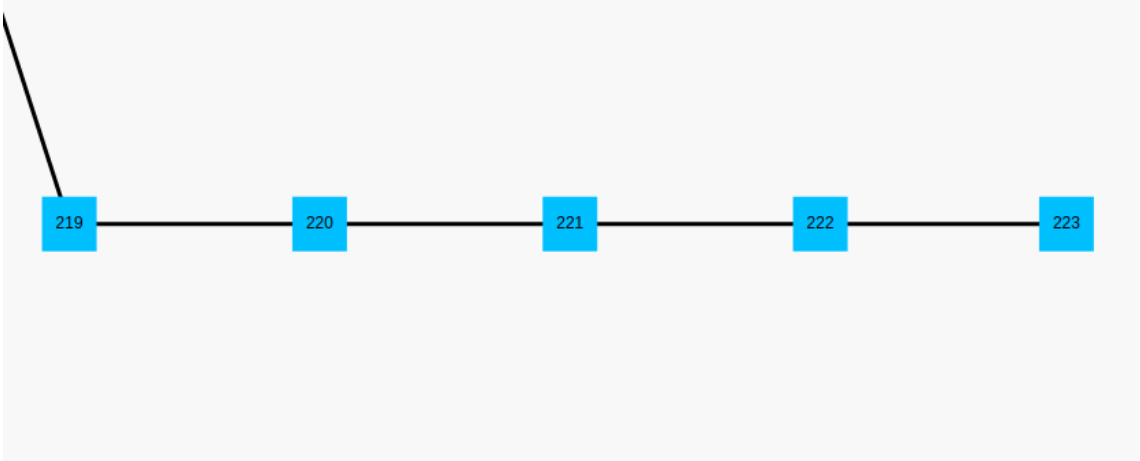

Figura 9. Exemplo de interatividade: zoom aplicado na cadeia principal do cenário 4

É importante destacar que, sempre que há alguma mudança na blockchain monitorada pela aplicação, um zoom já é aplicado automaticamente no final da blockchain de modo a focar nos últimos blocos inseridos. Além disso, existe também a possibilidade de arrastar o gráfico para visualizar os blocos mais antigos da blockchain, como demonstrado na Fig. 10.

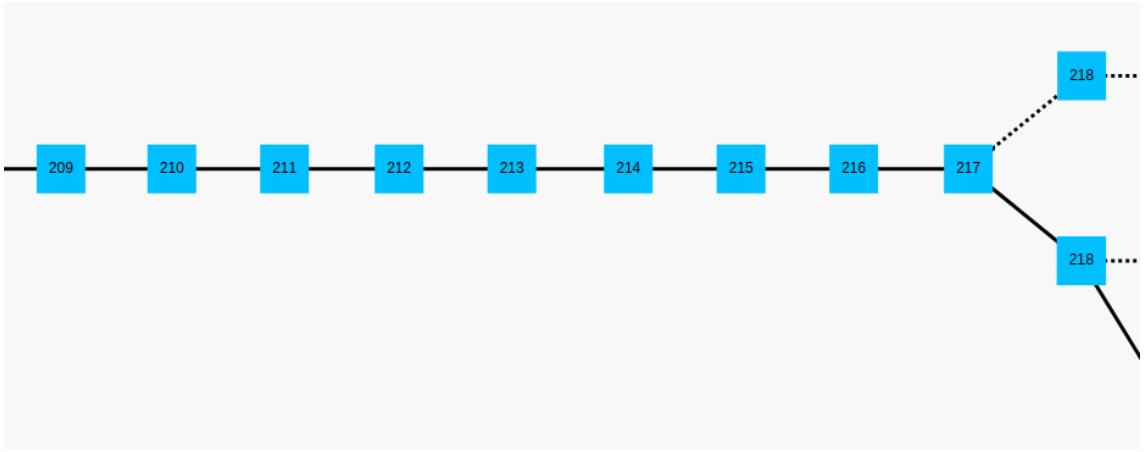

Figura 10. Exemplo de interatividade: ao arrastar o gráfico com o mouse são apresentadas mais informações da blockchain

Todos esses recursos estão disponíveis em uma barra de ferramentas que se localiza na parte superior do gráfico, como foi mostrado na Fig. 3. Se a ferramenta de zoom for utilizada, ou se o gráfico for arrastado para visualização de outras informações, é possível voltar para a visualização dos últimos blocos da blockchain através do botão *home*, ilustrado por uma casa. Além disso, é possível que o usuário exporte uma imagem da representação atual da blockchain através de um arquivo em formato png.

Vale ressaltar que estes resultados foram obtidos através de uma visão local de um único nó participante do consenso. Assim, é possível que sejam apresentadas visões distintas se a interface for executada em outro participante. Isso ocorre, porque em alguns momentos os nós podem não estar sincronizados, principalmente em relação aos últimos blocos recebidos ou criados. Esse é um recurso importante da ferramenta pelos seguintes motivos:

- não depende de nenhum elemento central para coleta e apresentação de resultados, o que pode se tornar inviável à medida que o número de nós da rede cresce;
- permite ao desenvolvedor da blockchain avaliar de forma mais precisa e independente a evolução da mesma à medida que os nós trocam mensagens e o protocolo de consenso evolui em sua busca por uma cadeia única e estável;
- permite que o usuário escolha e monitore quantos nós quiser da rede que mantém a blockchain, bastando para isso conectar diferentes janelas do seu navegador aos diferentes nós que deseja monitorar.

6. Conclusões e Trabalhos Futuros

Um bom mecanismo de consenso é de grande importância para que a blockchain possa evoluir com segurança. Dada a dificuldade em desenvolver um novo mecanismo de consenso para blockchains públicas, uma ferramenta capaz de oferecer informações gráficas sobre a evolução da blockchain em tempo real pode contribuir significativamente para que as funcionalidades deste mecanismo sejam avalidas de maneira visual e em tempo real. A ferramenta construída a partir de uma classe seguindo o paradigma orientado a objetos foi importante para reduzir o vínculo entre a aplicação e a blockchain que ela monitora, aumentando sua capacidade de integração com outros mecanismos ou cenários.

Como trabalhos futuros, é necessário aprimorar a integração da ferramenta com a blockchain, onde será importante definir casos de uso que facilitem o monitoramento de vários nós. Para isso, a classe da ferramenta gráfica poderá utilizar interfaces que permitirão que ela envie os dados para servidores externos ao ambiente do nó. Neste sentido, a partir destes servidores a interface gráfica pode monitorar a evolução da rede *peer-to-peer* controlada pelo mecanismo de consenso de modo a fornecer uma visão global da evolução da blockchain que não seja dependente da visão exclusiva de um único nó.

Além disso, é importante considerar situações que provocam modificações na visualização gráfica da blockchain e que não sejam decorrentes de inserções por parte do nó. Neste sentido, novos eventos, além daqueles apresentados no diagrama da Fig. 2 precisam ser considerados. O envio destes eventos ocorrerá, por exemplo, quando algum bloco for removido da visão local do nó ou mesmo quando um bloco é considerado confirmado, ou seja, quando ele não puder mais ser revertido a partir de uma outra cadeia formada por um *fork*, segundo os critérios estabelecidos em cada mecanismo de consenso. Neste último caso, espera-se que cores diferentes sejam consideradas para representar blocos confirmados na interface gráfica. Por fim, planeja-se mostrar também informações mais detalhadas sobre as transações contidas em cada bloco.

Referências

Bashir, I. (2017). Mastering Blockchain. Packt Publishing.

Dhillon, V., Metcalf, D., and Hooper, M. (2017). The Hyperledger Project. In Blockchain Enabled Applications, pages 139–149. Springer.

Greve, F., Sampaio, L., Abijaude, J., Coutinho, A., Ítalo Valcy, and Queiroz, S. (2018). Blockchain e a revolução do consenso sob demanda. XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Minicurso(1).

Hagberg, A., Swart, P., and Schult, D. (2008). Networkx: Network analisis in python. `https://networkx.github.io/` (Acessado em 04/07/2020).

Hossain, S. (2019). Visualization of Bioinformatics Data with Dash Bio. In Chris Calloway, David Lippa, Dillon Niederhut, and David Shupe, editors, Proceedings of the 18th Python in Science Conference, pages 126 – 133.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. Computing in Science & Engineering, 9(3):90–95.

King, S. (2013). Primecoin: Cryptocurrency with prime number proof-of-work. `http://primecoin.io/bin/primecoin-paper.pdf` (Acessado em 04/07/2020).

Martins, D. F. G. and Henriques, M. A. A. (2020). Avaliação da incidência de forks no algoritmo de consenso Probabilistic Proof-of-Stake (PPoS). In Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. III Workshop WBlockchain, Porto Alegre, RS, Brasil. SBC.

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. `http://www.bitcoin.org/bitcoin.pdf` (Acessado em 04/07/2020).

Parmer, C., Parmer, J., Sundquist, M., and Johnson, A. (2015). Collaborative data science. Montreal, QC. Plotly Technologies Inc.

Reeves, B. (2011). Blockchain Explorer. https://www.blockchain.com/explorer (Acessado em 09/09/2020).

Vasin, P. (2017). Blackcoin's proof-of-stake protocol v2. https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf (Acessado em 04/07/2020).