Rastreabilidade de alocação e desalocação de contêineres usando Docker Swarm com base em blockchain consorciado

Marco Antonio Marques[1], **Charles C. Miers**[1]

[1]Programa de Pós-Graduação em Computação Aplicada (PPGCA)
Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – Joinville, SC – Brasil

marco.marques@edu.udesc.br, charles.miers@udesc.br

Resumo. *Em um cenário de computação em nuvem, no qual serviços são desenvolvidos e oferecidos a clientes em uma plataforma terceirizada, o monitoramento é um aspecto essencial na otimização da escalabilidade, distribuição dos serviços, detecção e prevenção de falhas, cobrança pelos recursos utilizados, dentre outros aspectos. Neste cenário, cada ator envolvido pode implementar sua própria ferramenta de monitoramento. Esta característica pode levar a possíveis divergências quanto aos eventos coletados. Nestas situações, surge a dúvida sobre qual dos diferentes dados coletados está correto. Este trabalho apresenta uma proposta de solução que colete e promova a validação e o consenso entre os atores participantes quanto aos eventos de ciclo de vida de contêineres coletados, garantindo sua integridade, disponibilidade, irretratabilidade e auditoria.*

1. Introdução e Motivação

A arquitetura de microsserviços permitiu aumentar o desempenho de aplicações, fragmentando-as em partes independentes de desenvolvimento, versionamento, provisionamento e escalabilidade [Jamshidi, 2018]. Contêineres são considerados o padrão de microsserviços na nuvem devido a sua rapidez, facilidade na alocação, gerenciamento escalável e resiliência [Newman, 2015]. Tal fato conduziu ao desenvolvimento de plataformas de orquestração de contêineres, projetadas para gerenciar a implantação de aplicações conteinerizadas em aglomerados de larga escala, sendo capazes de executar centenas de milhares de trabalhos em diversas máquinas [Rodriguez and Buyya, 2018].

Com a criação de provedores de computação em nuvem, empresas passaram a oferecer recursos computacionais, plataformas e aplicações sob demanda. Este cenário permitiu a desenvolvedores e parceiros comercializar seus softwares e produtos para clientes, executando-os nas plataformas de computação em nuvem disponibilizadas pelos provedores. Neste contexto, o monitoramento e acompanhamento da execução do ambiente virtual é ferramenta importante para que todos os atores envolvidos possam acompanhar e otimizar a execução das aplicações no ambiente, rastrear falhas, cobrar e avaliar a cobrança pelos recursos computacionais utilizados, dentre outras atividades.

De maneira geral, o provedor de computação em nuvem disponibiliza um sistema próprio de monitoramento para o acompanhamento por parte dos clientes. Por outro lado, é possível aos clientes implementar um serviço de monitoramento independente [Dawadi et al., 2017]. Esta possibilidade traz como benefício a autonomia dos atores com relação ao monitoramento, podendo adaptá-lo às suas necessidades. Contudo, traz também a

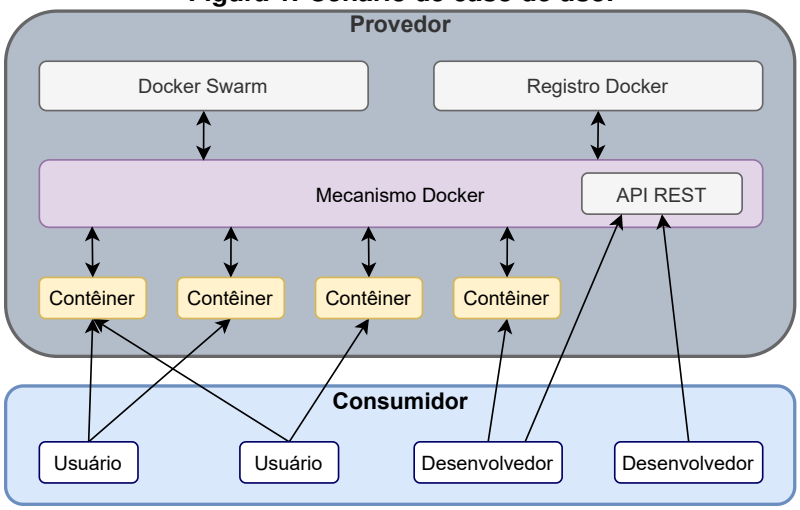
possibilidade de divergências entre os atores quanto às informações coletadas por suas ferramentas, resultando em análises distintas do mesmo cenário. Considerando especificamente os eventos de ciclo de vida de contêineres, divergências na coleta de eventos de inicialização, suspensão e finalização podem resultar em análises imprecisas quanto ao ambiente e às aplicações em execução. As soluções de monitoramento centralizadas, por outro lado, dependem diretamente da confiança naquele ator quanto à garantia da integridade dos dados coletados.

O problema apresentado por este trabalho consiste em como garantir a validação e o consenso entre os atores participantes dos cenários que envolvem virtualização em contêineres quanto aos eventos de ciclo de vida de contêineres coletados, permitindo sua auditoria por qualquer uma das partes, mantendo estes dados de forma a preservar sua integridade, disponibilidade e irretratabilidade. O trabalho está organizado da seguinte forma: a Seção 2 apresenta a definição do problema e os trabalhos relacionados, a Seção 3 descreve a proposta apresentada e como ela atende aos requisitos propostos e a Seção 4 traz considerações e informações sobre trabalhos futuros.

2. Problema e Trabalhos relacionados

Monitorar consiste em acompanhar a execução do ambiente virtual, coletando e disponibilizando para análise, periodicamente, um conjunto de variáveis pré definidas. Em um cenário de computação em nuvem, o monitoramento e gestão deste ambiente é um aspecto essencial da infraestrutura, que possibilita melhorar a escalabilidade e distribuição dos serviços, detecção e prevenção de falhas, análise de desempenho, dentre outros [Jiménez et al., 2015]. A Figura 1 representa um cenário de computação em nuvem, no qual uma aplicação distribuída, hospedada em um provedor de computação em nuvem, é oferecida por um desenvolvedor a um cliente. Neste cenário (Figura 1), há a interação entre

Figura 1. Cenário de caso de uso.



dois grupos de atores: Provedor e Consumidor. O grupo Consumidor é composto tanto pelos desenvolvedores, que hospedam suas aplicações no provedor de computação em nuvem, quanto pelos usuários destas aplicações. Cada um destes atores tem perspectivas e necessidades distintas: para o Provedor, o monitoramento é uma ferramenta chave para a gestão da infraestrutura, facilitando a identificação e retorno de falhas, bem como a cobrança pelos recursos utilizados. Para o desenvolvedor, o monitoramento é importante

para o desenvolvimento e otimização das aplicações, através do qual é possível identificar falhas e monitorar tempos de execução, facilitando a otimização do código e redução dos custos. Já para o usuário da aplicação, o monitoramento é importante para identificar falhas, indisponibilidades nos serviços contratados e controle de custos.

A possibilidade de cada ator ter seu próprio mecanismo de monitoramento dá margem a possíveis divergências quanto aos eventos coletados. Nestas situações, surge a dúvida sobre qual dos diferentes dados coletados está correto. O problema apresentado por este trabalho consiste em como coletar eventos de ciclo de vida de contêineres de modo não intrusivo e garantir sua validação e consenso entre os atores participantes dos ambiente, permitindo sua auditoria por qualquer uma das partes. Para atendimento aos requisitos listados, é necessário que o volume disponível para armazenamento dos eventos coletados seja dimensionado conforme o ambiente de execução, considerando o número de contêineres em operação. Todos os eventos relacionados à criação, suspensão e destruição de contêineres gerados no ambiente de execução devem ser coletados.

Através de pesquisa nos mecanismos de busca acadêmicos Google Scholar, ACM Digital Library, IEEE Xplore e Springer Link, foram selecionados, dentre os trabalhos identificados, a proposta de [Oliveira et al., 2017], que utiliza repositórios distribuídos para armazenamento e o trabalho desenvolvido por [Pourmajidi and Miranskyy, 2018], que utiliza uma blockchain para armazenamento de logs. Contudo, ambos os trabalhos não coletam eventos de ciclo de vida de contêineres, e em [Oliveira et al., 2017] não há validação dos dados coletados.

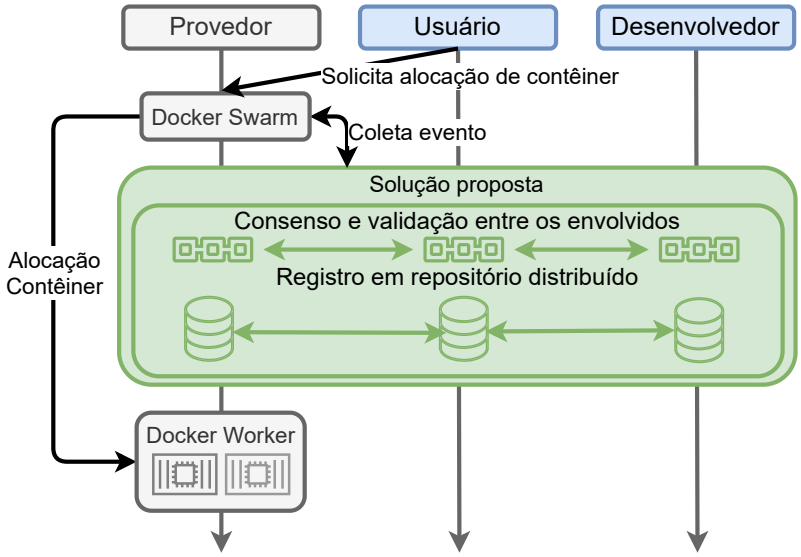
3. Proposta

A questão de pesquisa apresentada traz uma nova problemática ao monitoramento de ambientes de virtualização em contêineres com interações entre diversos atores, decorrente da possível divergência entre os dados coletados por seus sistemas de monitoramento. Contudo, apenas o consenso entre os participantes quanto aos dados coletados não é suficiente para o atendimento pleno ao problema apresentado. Após o consenso, os dados devem ser armazenados em repositório distribuído e seguro, de forma que não possam ser adulterados ou excluídos, sendo também possível auditar os registros de modo a identificar qualquer tentativa de manipulação. A Figura 2 é um diagrama de sequência exemplificando a proposta.

Neste modelo, a solução proposta coleta, via *Application Programming Interface* (API) do Docker, os eventos especificados referentes ao ciclo de vida de contêineres. Após a coleta, estes são formatados e enviados conforme o modelo de repositório utilizado. Os eventos coletados devem ser validados quanto ao seu emissor e conteúdo sendo, em seguida, aplicado um mecanismo para garantir o consenso entre os atores participantes quanto aos dados coletados e validados. Por fim, após o consenso, os dados são armazenados em um repositório distribuído. Caso a validação ou o consenso falhem, os dados não devem ser armazenados.

Neste sentido, a proposta atende a estes requisitos implementando uma solução que, após a coleta dos eventos de ciclo de vida de contêineres via API, os envia a uma blockchain modelo consórcio, composta pelos atores envolvidos, que os validam através da implementação do mecanismo de consenso, armazenando-os de forma distribuída. A implementação de uma blockchain consórcio como repositório de armazenamento dis-

Figura 2. Exemplo de aplicação da solução proposta.



tribuído garante a validação das transações por parte dos atores envolvidos, que também serão responsáveis pelo seu armazenamento.

A blockchain é um livro-razão seguro, compartilhado e distribuído que facilita o processo de gravação e rastreamento de recursos sem a necessidade de confiança em uma autoridade central, permitindo que duas partes comuniquem-se e troquem recursos em uma rede ponto a ponto na qual decisões são tomadas pela maioria, e não por uma única entidade [Salman et al., 2019]. Existem diferentes modelos de blockchain disponíveis, que distinguem-se quanto às suas características de permissionamento e consenso.

O permissionamento de uma blockchain define quem está autorizado a publicar blocos. No modelo não permissionado, qualquer nó pode publicar um novo bloco na rede blockchain. Já no modelo permissionado, apenas nós autorizados podem publicar blocos [Yaga et al., 2018]. Com base no modelo de permissionamento adotado, as blockchains são categorizadas entre pública, consórcio ou privada. Cada uma destas categorias possui uma aplicação distinta, baseada em um conjunto de características da blockchain. A Tabela 1 apresenta alguns critérios de comparação entre as principais características dos três tipos de blockchain [Miers et al., 2019].

Características	Blockchain Pública	Blockchain Consórcio	Blockchain Privada
Consenso distribuído	Todos os nós	Nós selecionados	Nós selecionados
Permissão de verificação	Pública	Restrita	Restrita
Imutabilidade	Sim	Adulterável	Adulterável
Centralização	Descentralizado	Parcial	Centralizado
Processo de consenso	Todos os nós	Nós selecionados	Nós selecionados

Tabela 1. Comparação entre modelos de acesso. Fonte: [Miers et al., 2019]

O consenso distribuído define quais nós podem participar do processo de consenso. A permissão de verificação pode variar entre pública ou restrita, com a identidade dos participantes podendo ser anônima em blockchains públicas ou conhecida, nos modelos consórcio e privado [Tinu, 2018]. Quanto à imutabilidade dos dados, algumas implementações dos modelos consórcio e privada podem permitir alterações. Já quanto ao grau de centralização, os modelos podem variar de centralizado no modelo de blockchain

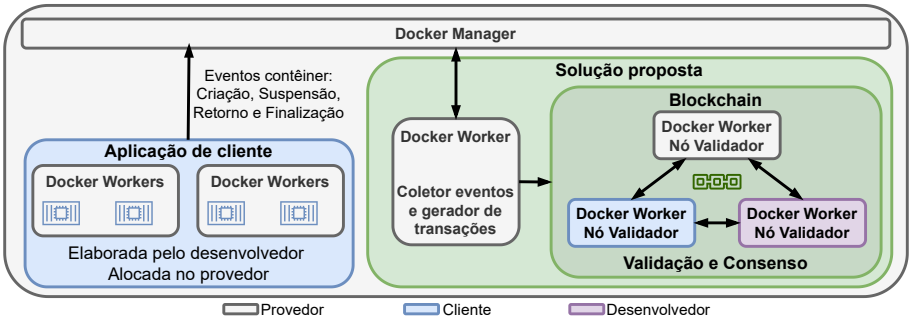
privada a descentralizado, no modelo público. Por fim, o processo de consenso define se qualquer entidade pode participar do processo ou apenas entidades pré-selecionadas.

O cenário no qual o problema alvo do trabalho está inserido é composto por um conjunto de atores pré-definidos, que deverão validar de forma consensual os eventos coletados. Neste modelo, o número de nós validadores da blockchain é conhecido previamente, e dividido entre um grupo de parceiros. Com isso, a blockchain consórcio é o modelo mais adequado às características do cenário proposto. Dentre os modelos de blockchain consórcio disponíveis, o Hyperledger Fabric foi escolhido como modelo a ser implementado na solução proposta devido à sua adequação aos requisitos especificados. Além disso, é um modelo disponível nos maiores provedores de computação em nuvem [Hyperledger, 2020].

A segunda característica que define a blockchain é o mecanismo de consenso, recurso que permite que redes distribuídas ou descentralizadas tomem decisões unânimes quando necessário [Sankar et al., 2017]. Este mecanismo cria um sistema consistente, no qual todos os nós concordam com a ordem dos blocos e seus conteúdos. O Hyperledger Fabric utiliza o mecanismo de consenso RAFT [Ongaro and Ousterhout, 2014], no qual um nó é eleito líder e distribui as transações para os demais, que irão validar e retornar o resultado. Baseado neste resultado, o líder registra a mensagem e informa os demais nós.

A solução proposta possui basicamente dois componentes: uma blockchain Hyperledger Fabric modelo consórcio, com no mínimo três atores (*i.e.,* Provedor, Desenvolvedor e Usuário), responsável pela validação das transações contendo os eventos e posterior armazenamento no *ledger*, e uma aplicação conteinerizada responsável pela coleta dos eventos e geração das transações. A Figura 3 representa, de modo genérico, a proposta apresentada.

Figura 3. Modelo genérico de solução.



Neste modelo, os eventos são coletados diretamente da interface API do Docker. Para realizar o envio das transações, o componente precisa de um *token JSON Web Token* (JWT), gerado previamente, afim de impedir o envio de transações por nós não autorizados. As transações recebidas são validadas entre os nós validadores pertencentes aos atores, utilizando o mecanismo de consenso RAFT, implementado pelo Hyperledger Fabric. Esta validação visa garantir a integridade dos dados coletados. Após a validação, os dados são armazenados na blockchain, de modo que todos os nós componentes terão uma cópia do conteúdo, garantindo também a disponibilidade das dos dados.

4. Considerações e trabalhos futuros

A solução proposta prevê a implementação em ambiente Docker, utilizando o orquestrador Docker Swarm e blockchain Hyperledger Fabric. Para coleta e geração das transações

foi desenvolvido um *javascript* que conecta-se ao *endpoint* da API de eventos do Docker, gera as transações em tempo real e envia à API de transações da blockchain. Como trabalho futuro será implementada a ferramenta Hyperledger Caliper, para realização de *benchmark* da implementação. Após a realização dos testes de funcionalidade e desempenho, será possível avaliar a viabilidade da solução proposta.

Agradecimentos: Os autores agradecem o apoio do LabP2D/UDESC e da FAPESC.

Referências

Dawadi, B., Shakya, S., and Paudyal, R. (2017). Common: The real-time container and migration monitoring as a service in the cloud. *Journal of the Institute of Engineering*, 12:51.

Hyperledger (2020). Hyperledger annual report. https://www.hyperledger.org/learn/publications/hyperledger-annual-report.

Jamshidi, P. e. a. (2018). The journey so far and challenges ahead. *IEEE Software*, 35(3):24–35.

Jiménez, L. L., Simón, M. G., Schelén, O., Kristiansson, J., Synnes, K., and Åhlund, C. (2015). Coma: Resource monitoring of docker containers. In *CLOSER*, pages 145–154.

Miers, C., Koslovski, G., Pillon, M., Simplicio, M., Carvalho, T., Rodrigues, B., and Battisti, J. (2019). *Análise de Mecanismos para Consenso Distribuído Aplicados a Blockchain*, page 50. SBC.

Newman, S. (2015). *Building Microservices*. OReilly, 1 edition.

Oliveira, F., Suneja, S., Nadgowda, S., Nagpurkar, P., and Isci, C. (2017). A cloud-native monitoring and analytics framework. Technical report, Technical Report RC25669, IBM Research.

Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'14, page 305–320, USA. USENIX Association.

Pourmajidi, W. and Miranskyy, A. (2018). Logchain: Blockchain-assisted log storage. pages 978–982.

Rodriguez, M. A. and Buyya, R. (2018). Container-based cluster orchestration systems: Taxonomy and future directions. *CoRR*, abs/1807.06193.

Salman, T., Zolanvari, M., Erbad, A., Jain, R., and Samaka, M. (2019). Security services using blockchains: A state of the art survey. *IEEE Communications Surveys Tutorials*, 21(1):858–880.

Sankar, L. S., Sindhu, M., and Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1–5. IEEE.

Tinu, N. (2018). A survey on blockchain technology- taxonomy, consensus algorithms and applications. *International Journal of Computer Sciences and Engineering O*, 6.

Yaga, D., Mell, P., Roby, N., and Scarfone, K. (2018). Nistir 8202 - blockchain technology overview. Technical report, NIST.