

# Registro de ciclo de vida de contêineres usando Docker Swarm com base em *blockchain* Hyperledger consorciado: Desempenho e impacto

Marco A. Marques<sup>1</sup>, Charles C. Miers<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Computação Aplicada (PPGCA)  
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC – Brasil

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

**Resumo.** *A virtualização baseada em contêiner permite alocação dinâmica, possibilitando atender às necessidades de escalabilidade e tolerância a falhas. Este artigo apresenta uma proposta de uso de blockchain consorciada baseada na solução Hyperledger para registrar eventos de alocação e desalocação de contêineres usando o Docker Swarm. Gravar esses eventos em uma blockchain permite auditar os recursos empregados e a cronologia das operações.*

## 1. Introdução

A virtualização em contêineres é considerada o padrão de microsserviços na nuvem [Vaucher 2018]. Tal fato conduziu ao surgimento de plataformas de orquestração de contêineres, projetadas para gerenciar aplicações containerizadas em aglomerados de larga escala, capazes de executar centenas de milhares de trabalhos em diversas máquinas [Rodriguez and Buyya 2018]. Este trabalho apresenta o uso da tecnologia *Blockchain* modelo Hyperledger Fabric como repositório seguro e auditável de registro de eventos de criação, pausa e destruição de contêineres e avalia como o desempenho da solução proposta pode afetar o registro destes eventos.

A proposta é registrar os eventos gerados pelo Docker Swarm, em ambiente de virtualização em contêineres, e armazená-los em *blockchain* Hyperledger Fabric consorciada. No modelo consorciado, a rede é controlada por um grupo restrito de nós. A utilização da *blockchain* busca agregar características como o fato de ser um livro-razão distribuído, composto de uma cadeia de blocos interligados através do uso de um ou mais *hashes* do bloco anterior [El Ioini and Pahl 2018].

Em situações de existência de autoridade central, a confiança depositada nesta autoridade é baixa, pois não existe mecanismo de verificação da validade dos registros. A tecnologia *blockchain* soluciona o problema de confiança por manter os registros e transações em uma rede distribuída [Sutton and Samavi 2017].

## 2. Definições & Detalhamento

O ciclo de vida dos contêineres é composto pelos seguintes estados: em execução, pausado, parado ou deletado. Um nó é uma instância Docker participando do *swarm*, podendo ser do tipo *master* ou *worker*. Cada nó *worker* é responsável pela execução de tarefas ordenadas pelo nó *master*, e terá os eventos gerados coletados pela aplicação. O resultado da coleta deverá ser armazenado em um repositório temporário para que o agente de transação utilize-os para gerar as transações e enviar para o nó validador da *blockchain*.

A proposta deve ter características de disponibilidade e escalabilidade dimensionadas de modo a garantir o registro de todos os eventos de alocação e desalocação dos contêineres. Considerando quesitos de desempenho como tempo de execução, latência e taxa de transferência, dividiu-se a proposta em três momentos distintos:

**Leitura de eventos e armazenamento em local temporário.** O agente deverá ser capaz de coletar todos os eventos gerados pela execução de contêineres nos nós do ambiente Docker Swarm. A redundância será definida para garantir nível máximo de disponibilidade e a escalabilidade da aplicação deverá variar conforme média de eventos por segundo gerados pelo ambiente de implementação.

**Geração das transações contendo os eventos e envio ao nó validador.** O processo de geração das transações deverá buscar um equilíbrio entre tempo de execução e taxa de transferência. Para isso, deve ser definido o tamanho máximo do bloco e número de canais, dimensionado conforme o tamanho do ambiente e eventos por segundo.

**Validação da transação e inclusão na *blockchain*.** O Hyperledger Fabric possui diversos parâmetros configuráveis, como tamanho do bloco, política de endosso e canais. Um dos maiores desafios para configurar uma *blockchain* eficiente é encontrar o conjunto adequado de valores para estes parâmetros [Thakkar et al. 2018]. Se nenhuma estratégia específica for adotada, todos os nós da rede verificarão todas as transações, limitando o número de contratos inteligentes processados em cada bloco e seu tamanho máximo [Scherer 2017]. Assim, a *blockchain* deve ser dimensionada conforme o ambiente, considerando o número de transações por segundo geradas pelo agente.

### 3. Considerações & Trabalhos futuros

O dimensionamento das variáveis de desempenho da aplicação proposta busca garantir que os eventos gerados pelo ambiente Docker sejam integralmente registrados na *blockchain*. Como trabalho futuro, pretende-se implementar o ambiente de testes para verificar a funcionalidade e viabilidade da solução.

**Agradecimentos:** Os autores agradecem o apoio do LabP2D/UEDESC e a FAPESC.

### Referências

- El Ioini, N. and Pahl, C. (2018). Trustworthy orchestration of container based edge computing using permissioned blockchain. In *Proc. of 5th IOTSMS*.
- Rodriguez, M. A. and Buyya, R. (2018). Container-based cluster orchestration systems: Taxonomy and future directions. *CoRR*, abs/1807.06193.
- Scherer, M. (2017). Performance and scalability of blockchain networks and smart contracts. Master's thesis, Umeå University, Department of Computing Science.
- Sutton, A. and Samavi, R. (2017). Blockchain enabled privacy audit logs. In *Proc. of ISWC*, pages 645–660.
- Thakkar, P., Nathan, S., and Viswanathan, B. (2018). Performance benchmarking and optimizing hyperledger fabric blockchain platform. In *2018 IEEE 26th MASCOTS*, pages 264–276.
- Vaucher, S. e. a. (2018). Sgx-aware container orchestration for heterogeneous clusters. In *Proc. of 38th ICDCS. IEEE 38th Int. Conference Distrib. Comput. Syst. (ICDCS)*.