PROPOSTA DE ESTRATÉGIA PARA MAXIMIZAR A CONFIABILIDADE NO PROVISIONAMENTO DE APLICAÇÕES NA BORDA

MARCOS P. KONZEN[1]; FABIO D. ROSSI[2]; PAULO S. SOUZA[3]; JÚLIO C. B. MATTOS[4]

[1]*Universidade Federal de Pelotas – mkonzen@inf.ufpel.edu.br*
[2]*Instituto Federal Farroupilha – fabio.rossi@iffarroupilha.edu.br*
[3]*Unipampa – paulosilas@unipampa.edu.br*
[4]*Universidade Federal de Pelotas – julius@inf.ufpel.edu.br*

1. INTRODUÇÃO

A Computação na Borda (*Edge Computing*) tem se consolidado como um paradigma de computação distribuída, de arquitetura descentralizada que disponibiliza recursos computacionais mais próximos dos usuários (geralmente a um salto), como uma forma de estender os serviços de nuvem para a borda da rede (SATYANARAYANAN, 2017). A ideia principal é processar as tarefas computacionalmente intensivas em um servidor de borda próximo do usuário. Além disso, devido à característica da arquitetura descentralizada, a Computação na Borda possibilita atender aos requisitos de mobilidade dos usuários, através do posicionamento e migração de aplicações para garantir uma qualidade de serviço (*Quality of Service - QoS*) consistente (SONKOLY et al., 2021).

Atualmente, há um grande interesse por parte dos pesquisadores em desenvolver aplicações que possam ser processadas na borda da rede, pois permite executar aplicações móveis emergentes que exigem baixa latência (ZHOU; ZHOU; HOPPE, 2023). Neste sentido, a arquitetura de computação *Serveless* e contêineres têm ganhado destaque na implantação de aplicações e microsserviços na borda, devido à capacidade de abstrair o gerenciamento de infraestrutura dos desenvolvedores, minimizando a complexidade associada à infraestrutura da borda (RAITH; NASTIC; DUSTDAR, 2023).

No entanto, devido ao modelo altamente distribuído e com menos recursos de suporte, as infraestruturas de borda são mais suscetíveis a falhas que podem causar a indisponibilidade dos recursos e afetar a confiabilidade na entrega dos serviços. No nível do usuário, essas falhas podem causar erros na aplicação, comprometendo o nível de QoS, impactando negativamente na Qualidade de Experiência (*Quality of Experience, QoE*), pois não mantém os Acordos de Nível de Serviços (*Service Level Agreement, SLAs*) firmados entre provedor e consumidor (SOUZA et al., 2022).

Embora diversos esforços foram feitos para melhorar o desempenho das aplicações na borda, a orquestração de aplicações consciente da confiabilidade da infraestrutura de borda ainda é um desafio, principalmente quando levamos em consideração o modelo de implantação das aplicações na borda, a heterogeneidade da infraestrutura, a limitação de recursos, a imprevisibilidade das falhas, os requisitos das aplicações, dos usuários e dos provedores de borda (APAT, NAYAK e SAHOO, 2023).

Dito isto, o objetivo deste trabalho é implementar uma estratégia de provisionamento consciente de confiabilidade de aplicações na borda, enquanto atende simultaneamente aos requisitos de desempenho. Como parte do desenvolvimento deste trabalho, propomos o algoritmo *ETHOS* (*Edge-Trusted HOSt*)

(KONZEN et al., 2023), uma heurística que otimiza o posicionamento de aplicações sensíveis à confiabilidade em cenários de Computação na Borda.

2. METODOLOGIA

Para implementar e avaliar nossa proposta utilizamos o simulador *EdgeSimPy*[1] (SOUZA, FERRETO e CALHEIROS, 2023), um framework de simulação escrito em Python para modelagem e avaliação de políticas de gerenciamento de recursos em cenários de Computação de Borda. O *EdgeSimPy* apresenta uma arquitetura modular que incorpora várias abstrações funcionais de uma infraestrutura de borda, como servidores de borda, dispositivos de rede e aplicações. Além disso, o *EdgeSimPy* permite modelar de forma integrada estratégias de posicionamento de aplicações baseadas em recursos de ambientes de borda, como consumo de energia, latência e requisitos de confiabilidade, o que permite representar a estratégia proposta em nosso trabalho.

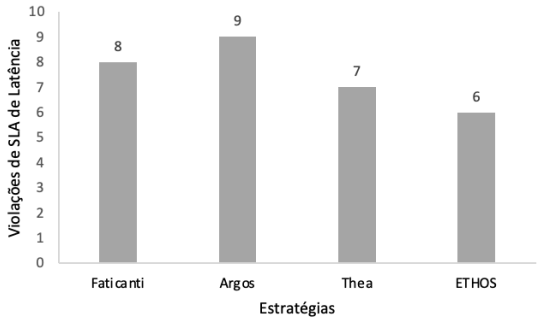
O cenário de estudo considerou uma infraestrutura com 18 servidores de borda de diferentes capacidades, conectados por uma topologia de rede *Mesh* (ARAL et al., 2021) com 208 links com a mesma latência. Os servidores de borda são classificados em três níveis de confiabilidade: Nível 0 (valor 0) não confiável, Nível 1 (valor 1) confiabilidade média, e Nível 2 (valor 2) totalmente confiável. Nesse cenário, modelamos diferentes aplicações com diferentes quantidades de serviços, e com diferentes requisitos de latência. Foram simulados ao todo 60 serviços com demandas heterogêneas. Cada serviço possui um requisito de nível de confiabilidade: nível 0 (sem requisito de confiabilidade), nível 1 (requisito de confiabilidade médio) e nível 2 (requisito de confiabilidade alto). Assumimos que uma violação de SLA de confiabilidade ocorre quando um serviço é hospedado por um servidor de borda que possui nível de confiabilidade menor do que o requisito de confiabilidade do serviço.

Comparamos nossa abordagem com outras três estratégias de posicionamento de aplicações em borda, *Faticanti* (FATICANTE et al., 2020), *Argos* (SOUZA et al., 2022) e *Thea* (SOUZA et al., 2023). Esses trabalhos utilizam o modelo de confiabilidade baseado em confiança no provedor que gerencia servidores de borda, restringindo a quantidade de recursos considerados confiáveis. Já o nosso trabalho amplia o modelo de confiabilidade para confiança individual em cada servidor de borda, independente do provedor que o gerencia.

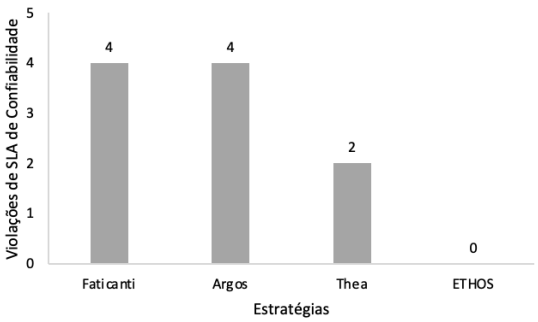
3. RESULTADOS E DISCUSSÃO

Como resultados parciais, avaliamos nossa proposta com as outras estratégias baseada nas métricas de violação de SLA de latência e violações de confiabilidade. As violações de SLA de latência se referem ao número serviços que tiveram sua latência excedida em relação ao limite exigido pelo serviço. Já as violações de confiabilidade estão relacionadas ao número de serviços que foram provisionados em servidores de borda que possuíam níveis de confiabilidade abaixo do requisito exigido pelo serviço. A Figura 1 resume os principais resultados.

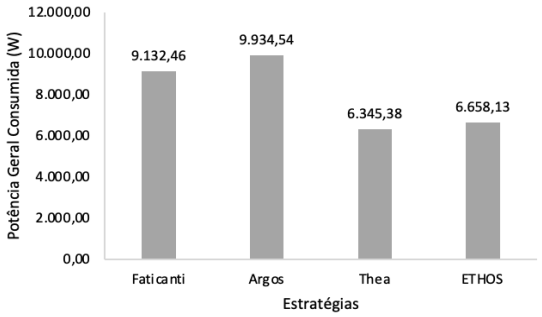
[1] https://github.com/EdgeSimPy



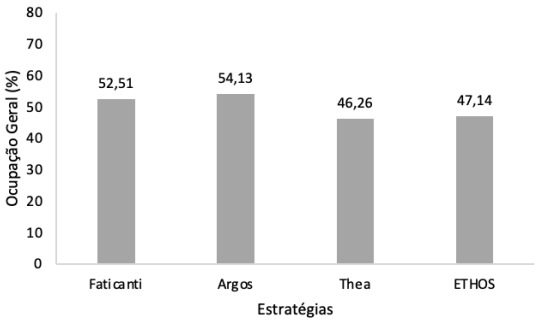
(a) Violações de *SLA* de Latência



(b) Violações de *SLA* de Confiabilidade



(c) Potência Geral Consumida (W)



(d) Ocupação Geral (CPU e RAM) (%)

Figura 1 - Comparação dos resultados das estratégias avaliadas.

Com relação às violações de SLA de latência (Figura 1(a)), o ETHOS obteve melhores resultados, pois, diferentemente das outras propostas, ele seleciona o servidor com base na confiabilidade individual de cada servidor, e não restringe a confiança apenas do provedor. Além disso, o ETHOS provisiona o servidor com recursos suficientemente próximos aos requisitos do serviço. As estratégias de Faticante e Argos provisionam os serviços de forma indiscriminada, sem levar em consideração os requisitos de confiabilidade. Thea leva em consideração o nível de confiabilidade apenas do provedor, limitando o escopo de servidores confiáveis a serem utilizados. Como o ETHOS leva em consideração a confiabilidade individual de cada servidor, independentemente do provedor que o gerencia, essa estratégia amplia o escopo de servidores confiáveis, conseguindo atender aos requisitos de confiabilidade dos serviços (Figura 1(b)).

O ETHOS também leva em consideração o consumo de energia dos servidores, ou seja, procura selecionar servidores que consomem menos energia (Figura 1(c)). Além disso, comparamos a ocupação geral dos recursos, demonstrando que o ETHOS, consegue otimizar a quantidade de recursos utilizados para atender a demanda dos serviços (Figura 1(d)), sem que outras métricas sejam sacrificadas, demonstrando a viabilidade da proposta.

4. CONCLUSÕES

Nesse trabalho, apresentamos uma estratégia baseada na confiabilidade dos servidores de borda para otimizar o posicionamento dos serviços das aplicações

com o objetivo de minimizar as violações de SLA de confiabilidade e desempenho, melhorando o QoS dos usuários, além de otimizar a utilização dos recursos.

Experimentos simulados mostram que nossa estratégia consegue reduzir as violações de SLAs de latência entre em relação a outras abordagens. Em relação à violação de SLAs de confiabilidade, nossa abordagem conseguiu atender aos requisitos de confiabilidade de todos os serviços. Além disso, os resultados mostram que nossa estratégia não sacrifica outras métricas, como a potência consumida e a taxa de ocupação dos recursos, demonstrando a viabilidade da nossa proposta. Em trabalhos futuros, pretende-se estender nossa heurística para fazer a orquestração sensível a falhas para o provisionamento de funções *Serverless* na borda, assim como modelar padrões de falhas em um cenário de borda.

5. REFERÊNCIAS BIBLIOGRÁFICAS

APAT, H. K.; NAYAK, R.; SAHOO, B. A comprehensive review on Internet of Things application placement in Fog computing environment. **Internet of Things**, v. 23, p. 100866, 2023.

KONZEN, M. P. et al. Estratégia de Posicionamento de Aplicações Sensíveis à Privacidade e Latência em Bordas Federadas. **Anais do XXIV Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD 2023)**, p. 73–84, 2023.

FATICANTI et al. Deployment of Application Microservices in Multi-Domain Federated Fog Environments. **International Conference on Omni-layer Intelligent Systems (COINS)**, 2020.

RAITH, P.; NASTIC, S.; DUSTDAR, S. Serverless Edge Computing—Where We Are and What Lies Ahead. **IEEE Internet Computing**, v. 27, n. 3, p. 50–64, 2023.

SATYANARAYANAN, M. The Emergence of Edge Computing. **Computer**, v. 50, n. 1, p. 30–39, jan. 2017.

SONKOLY et al. Survey on Placement Methods in the Edge and Beyond. **IEEE Communications Surveys Tutorials,** v. 23, n. 4, p. 2590–2629, 30 jul. 2021.

SOUZA, P. S. et al. Location-Aware Maintenance Strategies for Edge Computing Infrastructures. **IEEE communications letters**, v. 26, n. 4, p. 848–852, 2022.

SOUZA, P. S.; FERRETO, T.; CALHEIROS, R. N. EdgeSimPy: Python-based modeling and simulation of edge computing resource management policies. **Future Generation Computer Systems**, v. 148, p. 446–459, 1 2023.

SOUZA, P. et al. Thea - a QoS, Privacy, and Power-aware Algorithm for Placing Applications on Federated Edges. **31st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)**. 2023.

ZHOU, N.; ZHOU, H.; HOPPE, D. Containerisation for High Performance Computing Systems: Survey and Prospects. **IEEE Transactions on Software Engineering**, p. 1–20, 2022.