Estratégia de Posicionamento de Aplicações Sensíveis à Privacidade e Latência em Bordas Federadas

Marcos P. Konzen[1,3], **Paulo S. S. Souza**[2], **Fábio D. Rossi**[1], **Júlio C. B. Mattos**[3]

[1]Instituto Federal Farroupilha (IFFar) - Campus Alegrete
Alegrete/RS - Brasil

[2]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre/RS - Brasil

[3]Universidade Federal de Pelotas (UFPEL)
Pelotas/RS - Brasil

{marcos.konzen,fabio.rossi}@iffarroupilha.edu.br,

julius@inf.ufpel.edu.br, paulo.severo@edu.pucrs.br

Resumo. *Sobre ambientes de computação em borda federada, provedores de infraestrutura compartilham recursos com o objetivo de minimizar limitações de escalabilidade, e atender a crescente demanda por recursos das aplicações. No entanto, a heterogeneidade da infraestrutura, gerenciada por diferentes provedores de borda, impõe desafios relacionados ao equilíbrio entre os requisitos de desempenho e privacidade das aplicações. Algumas estratégias existentes tentam resolver essa questão implementando decisões de posicionamento baseadas no nível de confiança do usuário no provedor de borda. No entanto, isso acaba limitando a quantidade de servidores de borda considerados confiáveis. Diante disso, este artigo apresenta ETHOS (Edge-Trusted HOSt), uma estratégia que classifica e seleciona servidores de borda baseado em níveis de confiança individual em cada servidor, independentemente do provedor que o gerencia. Experimentos demonstram que o ETHOS é capaz de reduzir as violações de privacidade e, ao mesmo tempo, minimizar as violações de latência em comparação com outras abordagens da literatura.*

1. Introdução

A crescente necessidade de fornecer uma ampla gama de serviços e aplicações para usuários móveis, influenciada por diversos fatores, está exigindo cada vez mais recursos de processamento, menores tempos de reposta e uma maior confiabilidade no processamento dos dados. Tais aplicações exigem processamento em tempo real, são sensíveis à latência, além de manipularem dados sensíveis dos usuários e do ambiente ao redor [Xia et al. 2020]. Neste contexto, não é factível que estas aplicações sejam executadas em sua totalidade diretamente nos dispositivos móveis dos usuários, já que estes possuem limitações de recursos computacionais e de consumo de energia [Wang et al. 2019b] [Lin et al. 2019]. Para endereçar essa demanda, a computação na borda (*Edge Computing*) se consolidou como um paradigma que estende os recursos de nuvem. A computação na borda compreende em uma infraestrutura que utiliza uma camada de recursos computacionais que estão mais próximos dos usuários, trazendo como vantagens a baixa latência,

menor consumo de energia e menor custo de transmissão de dados [Feng et al. 2022] [Lin et al. 2019].

Embora a computação de borda supere algumas limitações em relação a problemas de latência e largura de banda, ela ainda apresenta alguns desafios. A maioria das infraestruturas de borda compreendem pequenos data centers com servidores heterogêneos interligados por uma rede pública de dados [Wang et al. 2019a]. Portanto, estes modelos possuem escalabilidade vertical limitada, motivando a necessidade de que os provedores devam explorar soluções em termos de escalabilidade horizontal. Com esse objetivo, alguns trabalhos exploram o conceito de bordas federadas, na qual os provedores de infraestrutura compartilham recursos uns com os outros para obter uma melhor utilização da infraestrutura [Faticanti et al. 2020] [Souza et al. 2023a]. Isto permite que os provedores ofereçam recursos suficientes para provisionar as aplicações, fornecendo uma qualidade de serviço (*Quality of Service, QoS*) aprimorada e, consequentemente, minimizar as violações de Acordos de Nível de Serviços (*Service Level Agreement, SLA*), impactando positivamente na qualidade de experiência (*Quality of Experience, QoE*) dos usuários finais [Xia et al. 2020].

Toda essa mobilidade de serviços e dados entre dispositivos móveis, borda e nuvem, ocorre através do suporte da virtualização. Máquinas virtuais e contêineres possibilitam a implementação de estratégias de alocação de recursos para provedores de infraestruturas [Kaiser et al. 2022]. Estas estratégias são definidas para otimizar a alocação de recursos de acordo com os requisitos e restrições dos provedores e dos usuários, como desempenho da rede, consumo de energia, custos operacionais, segurança, privacidade e atender aos *SLA*s [He et al. 2020]. Para se obter o máximo dessa infraestrutura heterogênea, as aplicações de borda são desenvolvidas no modelo de arquitetura de software baseada em tarefas fracamente acopladas, ou serviços. Desta forma, cada serviço pode ter diferentes requisitos de desempenho, segurança e privacidade, o que possibilita o posicionamento de forma independente entre os diferentes servidores de uma borda federada [Faticanti et al. 2020].

O posicionamento de aplicações na borda apresenta uma série de desafios. Por exemplo, posicionar tarefas em servidores muito distantes uns dos outros pode comprometer o nível de *QoS* e causar violações de *SLA*s da aplicação [Souza et al. 2022]. Além disso, os provedores de borda federada podem implementar diferentes políticas de proteção de dados em suas infraestruturas, impactando no nível de confiança das aplicações e, consequentemente, nas decisões de alocação de tarefas que exigem requisitos de privacidade de dados mais elevados [He et al. 2020] [Souza et al. 2023a]. Neste caso, tarefas que processam informações sensíveis à segurança e à privacidade devem ser colocados em servidores que cumpram os requisitos mínimos de segurança e privacidade exigido pela aplicação para evitar violações de segurança.

Para encontrar um plano de colocação adequado para várias aplicações na borda é necessário avaliar as restrições pelo lado do usuário, como privacidade e latência, e pelo lado dos provedores as restrições de infraestrutura, o que o torna um problema não trivial [Apat et al. 2023]. Alguns trabalhos propõem estratégias de alocação de aplicações em bordas federadas cujo objetivos são diminuir as violações de *SLA*s de latência e privacidade e maximizar a utilização de recursos dos provedores [Faticanti et al. 2020] [Souza et al. 2022], além de diminuir o consumo de energia [Souza et al. 2023a]. No en-

tanto, estas estratégias consideram o nível de confiança baseado em provedor, ou seja, o conjunto de recursos confiáveis fica restrito à infraestrutura do provedor de borda em que o usuário confia. Com isso, o escopo de recursos considerados confiáveis fica limitado, prejudicando a eficiência das estratégias de posicionamento.

Neste artigo apresentamos o ETHOS (*Edge-Trusted HOSt*), uma extensão do Thea proposto por Souza et al. [Souza et al. 2023a], que otimiza o posicionamento de aplicações baseada na confiança individual de servidores em uma borda federada. De forma geral, este trabalho traz as seguintes contribuições:

- Propomos uma heurística chamado ETHOS, um estratégia de posicionamento de aplicações que são sensíveis à latência e a requisitos de privacidade em bordas federadas. Implementamos diferentes níveis de confiabilidade individual em servidores de borda, o que torna possível selecionar o servidor mais adequado para posicionar as aplicações.
- Avaliamos a nossa proposta através de experimentos que simulam o posicionamento de aplicações compostas em um ambiente de borda federada, que demonstram que a nossa estratégia pode reduzir ainda mais as violações de privacidade e de latência, se comparadas abordagens mais recentes, sem sacrificar outras métricas.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta uma visão geral sobre bordas federadas e o problema de posicionamento de aplicações. A seguir, na Seção 3, são apresentados os trabalhos relacionados. Nas Seções 4 e 5 descrevemos a estratégia proposta e a metodologia deste trabalho, respectivamente. Na Seção 6 apresentamos a avaliação da nossa proposta. Por fim, na Seção 7, as conclusões e trabalhos futuros.

2. Background

A Computação de Borda, proposta pelo *European Telecommunication Standard Institute (ETSI)* [Institute 2023], tem atraído muitas pesquisas por se tratar de um modelo de arquitetura descentralizada que estende os recursos de nuvem para a borda da rede. Este paradigma aproveita uma camada de recursos computacionais entre o usuário e a nuvem que utiliza uma infraestrutura formada por servidores de borda distribuídos geograficamente em pequenos data centers [Lin et al. 2019]. Os dispositivos de borda (nós de borda) possuem poder computacional, armazenamento e taxas de comunicações mais robustos que um dispositivo de usuário final e compreendem em uma arquitetura heterogênea de hardware e software [Kar et al. 2023].

A infraestrutura de borda geralmente é formada por servidores de borda, gateways e roteadores inteligentes, estações de base e outros dispositivos que formam pequenos data centers, que são nuvens de pequena escala cujo objetivo é apoiar aplicações móveis que demandam muitos recursos computacionais e energia, provisionando recursos com menor latência [Feng et al. 2022]. Com isso, o objetivo é processar o máximo da carga de trabalho das aplicações nos nós de borda, ao invés de transferi-las para a nuvem. A computação de borda é uma plataforma altamente virtualizada, habilitando vários serviços para aplicações móveis emergentes que exigem baixa latência [Kaiser et al. 2022].

Embora a computação de borda apresente vantagens em relação a baixa latência e largura de banda, este modelo possui escalabilidade vertical limitada [Buyya et al. 2018].

Isto é, os sites de borda compreendem em alguns servidores de borda instalados em pequenos espaços físicos com alimentação de energia e refrigeração limitados e conectados a uma rede de dados pública, impondo limitações de recursos disponibilizados por um provedor de borda. Uma abordagem viável para mitigar as limitações de capacidade é empregar estratégias de escalabilidade horizontal, que incorpora recursos de diversos provedores [Wang et al. 2019a]. As bordas federadas implementam um modelo de cooperação, aumentando a disponibilidade de recursos para as aplicações. Uma federação de borda permite que provedores de infraestrutura forneçam desempenho otimizado de aplicações para usuários finais, enquanto beneficiam os provedores através de um melhor aproveitamento dos recursos [Jeong et al. 2021].

Em uma borda federada provedores podem estipular contratos entre si para alocar recursos, melhorando a escalabilidade e suprindo a falta de recursos para aplicações de grande escala [Faticanti et al. 2020]. Para obter o máximo dessa infraestrutura heterogênea, as aplicações de borda são desenvolvidas baseadas em um modelo de aplicações compostas, ou seja, são divididas em serviços independentes que se comunicam por meio de protocolos de passagem de mensagens, criando um fluxo de trabalho onde os serviços se comunicam pela rede [Brik et al. 2020]. Desta forma, cada serviço pode ter diferentes requisitos de desempenho, segurança e privacidade, e pode ser gerenciado e dimensionado de forma independente, o que amplia as possibilidades de posicionamento destes serviços na borda [Souza et al. 2023a].

Na Figura 1 representamos um cenário de borda federada composta por três diferentes de provedores de borda (Provedor 1, Provedor e Provedor 3), dividido em células hexagonais, conforme modelo de Aral et al. [Aral et al. 2021]. Neste cenário, cada célula representa a área de cobertura de cada estação de base equipada com um servidor de borda (SB), e interligadas através de um conjunto de links. As células que compõem a infraestrutura de borda são gerenciadas por diferentes provedores de borda, que fornecem recursos para os usuários alocarem os serviços de suas aplicações. Cada provedor de borda pode implementar diferentes políticas de proteção de dados em sua infraestrutura, o que influencia no nível de confiabilidade dos seus servidores. A expectativa dos usuários é que suas aplicações sejam posicionadas em servidores de borda que atendam aos seus *SLA*s de latência e privacidade. A premissa inicial deste trabalho consiste em que, mesmo que o Provedor 1 não seja confiável, pode existir um servidor confiável que faça parte de seu conjunto de servidores. E mesmo que o Provedor 2 seja confiável, pode existir um servidor comprometido que faça parte de seu conjunto de servidores.

3. Trabalhos Relacionados

Alguns trabalhos propõem estratégias de alocação de aplicações em bordas federadas cujos objetivos principais são diminuir as violações de *SLA*s de latência e privacidade e maximizar a utilização de recursos dos provedores. O modelo de compartilhamento de recursos entre provedores em uma borda federada permite firmar acordos entre outros provedores para alugar recursos visando atender possíveis demandas que excedem a sua capacidade. Neste sentido, Faticante et. al. [Faticanti et al. 2020] propõe uma estratégia de provisionamento de aplicações (chamada neste trabalho de Faticante) baseadas em microsserviços que são alocados em uma borda federada de acordo com a sua ordem de posição no fluxo de execução de suas aplicações. Nesta estratégia, o objetivo é alugar o mínimo possível de recursos de outros provedores para que o custo de alocação seja

Provedor	Servidores de Borda (SBs)	Confiabilidade Média
Provedor 1	SB1, SB4, SB7	Baixa
Provedor 2	SB6, SB8, SB9	Média
Provedor 3	SB2, SB3, SB5	Alta

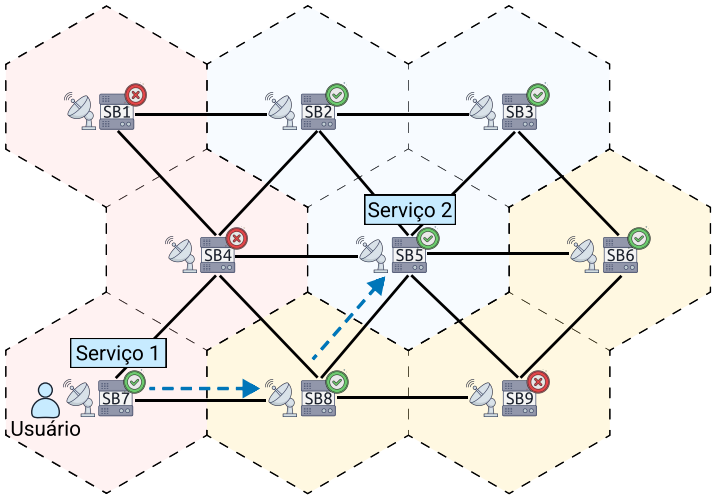


Figura 1. Cenário de borda federada gerenciada por três diferentes provedores de borda, com diferentes níveis de confiabilidade. Os usuários alocam suas aplicações em servidores de borda que atendem aos requisitos de latência e privacidade.

o menor possível e, ao mesmo tempo, garantir a execução das aplicações com base em restrições de localização.

Embora o modelo de federação de borda permita uma maior elasticidade na alocação de aplicações através do compartilhamento de recursos entre provedores, surgem algumas preocupações com a privacidade dos dados, já que os provedores podem ter diferentes políticas de segurança em suas infraestruturas. Souza et. al. [Souza et al. 2022] propõem um algoritmo heurístico, chamado Argos, que migra aplicativos baseados em microsserviços de acordo com a mobilidade dos usuários em bordas federadas, considerando os requisitos de privacidade dos serviços e os níveis de confiança dos usuários nos provedores de infraestrutura. O trabalho fornece uma solução para o conflito entre as políticas de proteção de dados implementados pelos provedores de infraestrutura e os requisitos de segurança e privacidade de aplicações sensíveis à privacidade.

Neste mesmo contexto, Souza et. al [Souza et al. 2023a] apresentam uma estratégia, o Thea, que otimiza a *QoS* (por meio da redução da latência e aumento da privacidade) do usuário final e os interesses dos provedores de infraestrutura (consumo de energia). A estratégia utiliza uma abordagem de provisionamento de aplicações compostas em bordas federadas e, diferentemente das estratégias de Argos e Faticante, o Thea define uma ordem de colocação que prioriza as aplicações maiores compostas por serviços com maiores requisitos de privacidade, evitando o provisionamento de aplicações em servidores de borda que violem os requisitos de privacidade.

Apesar de atingir resultados satisfatórios em relação à problemas de latência, violações de privacidade e consumo de energia, o Thea ainda apresenta um número con-

siderável de violações de privacidade, mesmo se comparado com estratégias que alocam indiscriminadamente serviços para servidores de borda não confiáveis. Assim como o Argos, o Thea utiliza o modelo de confiabilidade baseado em confiança no provedor, ou seja, o algoritmo considera que servidores de borda confiáveis são os que pertencem ao provedor de infraestrutura em que o usuário confia. Desta forma, o modelo de elasticidade fica comprometido, pois limita a quantidade de servidores de borda confiáveis para alocar aplicativos de grande escala sensíveis à privacidade.

Com o objetivo de tornar mais eficiente a utilização dos recursos em uma borda federada e diminuir as violações de privacidade, propomos neste trabaho uma estratégia que estende as funcionalidades do Thea, otimizando o posicionamento de aplicações baseada na confiança individual em servidores de borda federada. Assim, é possível estender o modelo de confiança abrangendo uma quantidade maior de servidores de borda que podem ser considerados confiáveis e, consequentemente, minimizar as violações de privacidade e de latência.

4. Estratégia Proposta

Nesta seção apresentamos o ETHOS (uma extensão da estratégia Thea), um algoritmo heurístico que posiciona aplicações em infraestruturas de borda federadas baseado nos requisitos de sensibilidade à latência e privacidade das aplicações, e no nível de confiabilidade dos servidores de borda. Nossa estratégia é demonstrada no Algoritmo 1.

Algoritmo 1: Algoritmo de posicionamento ETHOS

```
1  A = Lista de aplicações
2  E = Lista de servidores de borda
3  foreach aplicação A_i ∈ A do
4  |   ζ_i = norm(escore de latência (A_i)) + norm(escore de privacidade (A_i))
5  end
6  A' = Aplicações ordenadas de acordo com ζ (decr.)
7  foreach aplicação A_j ∈ A' do
8  |   S' = Lista de serviços que compõem A_j
9  |   foreach serviço S_j ∈ S' do
10 |   |   foreach servidor E_i ∈ E do
11 |   |   |   V_SLA = Cálculo da violação de SLAs de E_i
12 |   |   |   CSA = Cálculo do custo de serviços afetados de E_i
13 |   |   |   CL = Cálculo do custo de latência de E_i
14 |   |   |   PW = Cálculo da potência consumida de E_i
15 |   |   |   C_SLA_i = norm(V_SLA) + norm(CL) + norm(CSA)
16 |   |   end
17 |   |   E' = E ordenado por C_SLA (cresc.)
18 |   |   foreach E_k ∈ E' do
19 |   |   |   if E_k tem capacidade para hospedar S_j then
20 |   |   |   |   Provisiona S_j em E_k
21 |   |   |   |   break
22 |   |   |   end
23 |   |   end
24 |   end
25 end
```

Inicalmente, o algoritmo recebe a lista de aplicações compostas ($\mathcal{A}$) que devem ser provisionadas e a lista de servidores de borda ($\mathcal{E}$) disponíveis na infraestrutura de

borda federada. Aplicações que possuem requisitos de latência e privacidade mais rígidos possuem uma probabilidade de terem seus *SLA*s violados devido a limitação de recursos da borda. O ETHOS utiliza uma função de pontuação ζ (Alg.1, linha 4) para definir a ordem de provisionamento das aplicações. O escore de latência de ($\mathcal{A}_i$) é calculado com base no seu requisito de latência e na quantidade de servidores de borda próximos o suficente do usuário da aplicação para não violar a latência. O escore de privacidade de ($\mathcal{A}_i$) é baseado na demanda de recursos da aplicação (CPU e RAM) e dos requisitos de privacidade dos serviços que compõem ($\mathcal{A}_i$).

O algoritmo posiciona primeiro todos os serviços $\mathcal{S}_j$ que compõem a aplicação $\mathcal{A}_j$, para somente depois posicionar os serviços da aplicação seguinte. Para cada serviço $\mathcal{S}_j$ o ETHOS seleciona o servidor de borda mais adequado para alocar o serviço. Diferentemente das outras estratégias, o ETHOS avalia a confiabilidade de cada servidore de borda, independentemente do provedor que o gerencia. Desta forma, todos os servidores que compõem a infraestrutura de borda federada podem potencialmente hospedar o serviço (Alg. 1, linhas 10-20).

O ETHOS utiliza algumas funções de custo para ordenar a lista de servidores de borda candidatos a hospedar o serviço $\mathcal{S}_j$. A função $\mathcal{V}_{SLA}$ (Alg.1, linha 11) calcula o total de violações de *SLA*, que é a soma das violações de latência e de privacidade caso o serviço fosse hospedado naquele servidor. Já a função que calcula o custo de serviços afetados ($\mathcal{CSA}$) (Alg.1, linha 12) verifica o custo de privacidade de serviços que ainda não foram provisionados e que poderiam contar com esse servidor baseado no nível de confiança, ou seja, quanto mais serviços não provisionados confiam neste servidor, maior será o seu custo.

O custo de latência ($\mathcal{CL}$) (Alg.1, linha 13) calcula a latência entre o servidor $\mathcal{E}_i$ e o servidor que hospeda o serviço anterior ($\mathcal{S}_{j-1}$) da cadeia de serviços da aplicação $\mathcal{A}_i$, ou seja, quanto maior a latência, maior será o seu custo. Com isso, o ETHOS tenta manter os serviços de uma mesma aplicação alocados uns próximos aos outros. Como o consumo de energia é um fator relevante em infraestruturas de borda, a função que ($\mathcal{PW}$) (Alg.1, linha 14) calcula a potência que seria consumida para executar o serviço. Este parâmetro pode ser usado, por exemplo, para priorizar servidores que consomem menos energia. A última função calcula o custo total de *SLA* ($\mathcal{C}_{SLA}$) (Alg.1, linha 15), usado para classificar os servidores de borda do menor para o maior custo (Alg.1, linha 17).

Em seguida, algoritmo itera sobre a lista ordenada de servidores de borda e verifica se eles possuem recursos disponíveis suficientes para hospedar cada serviço. Ao encontrar um servidor apropriado para alocar o serviço, ETHOS provisiona o serviço $\mathcal{S}_j$ no servidor $\mathcal{E}_k$, e após, passa para o próximo serviço a ser provisionado (Alg.1, linha 18-20).

5. Metodologia

Para implementar e avaliar nossa proposta usamos o simulador EdgeSimPy [Souza et al. 2023b], 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 privacidade, o que permite representar a estratégia proposta em nosso trabalho.

Utilizamos o cenário proposto por Souza et al. [Souza et al. 2023a], o que possibilita compararmos de forma mais justa a nossa estratégia com as demais. Tal cenário considera 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 especificados em três diferentes modelos, conforme descritos na Tabela 1.

Tabela 1. Especificações dos servidores de borda [Ismail and Materwala 2021]

Modelo	CPU	RAM	Energia (ocioso)	Energia (máximo)
Modelo 1	32 núcleos	32 GB	265 Wh	1387 Wh
Modelo 2	48 núcleos	64 GB	127 Wh	559 Wh
Modelo 3	36 núcleos	64 GB	45 Wh	276 Wh

Os servidores de borda são gerenciados por três diferentes provedores, e cada provedor de borda implementa diferentes políticas de proteção e segurança de dados em sua infraestrutura. Um dos objetivos deste trabalho é que aplicações que exigem requisitos de privacidade sejam hospedadas em servidores de borda que possuam níveis de confiabilidade mínimos exigidos por aquele serviço, independentemente do provedor de borda. Diferentemente dos outros trabalhos, na nossa estratégia atribuímos um valor de confiabilidade individual para cada servidor de borda, e não mais vinculado ao provedor que o gerencia. Os servidores de borda são classificados em três níveis de confiabilidade: Nível 0 (valor 0) não confiáveis, Nível 1 (valor 1) confiabilidade média, e Nível 2 (valor 2) totalmente confiável. Como as outras estratégias citadas na literatura (Argos e Thea) atribuem um valor de confiabilidade baseado no provedor de borda, para podermos simular e comparar de forma justa, calculamos o valor de confiabilidade do provedor como sendo a média dos valores de confiabilidade dos servidores de borda que são gerenciados pelo mesmo provedor.

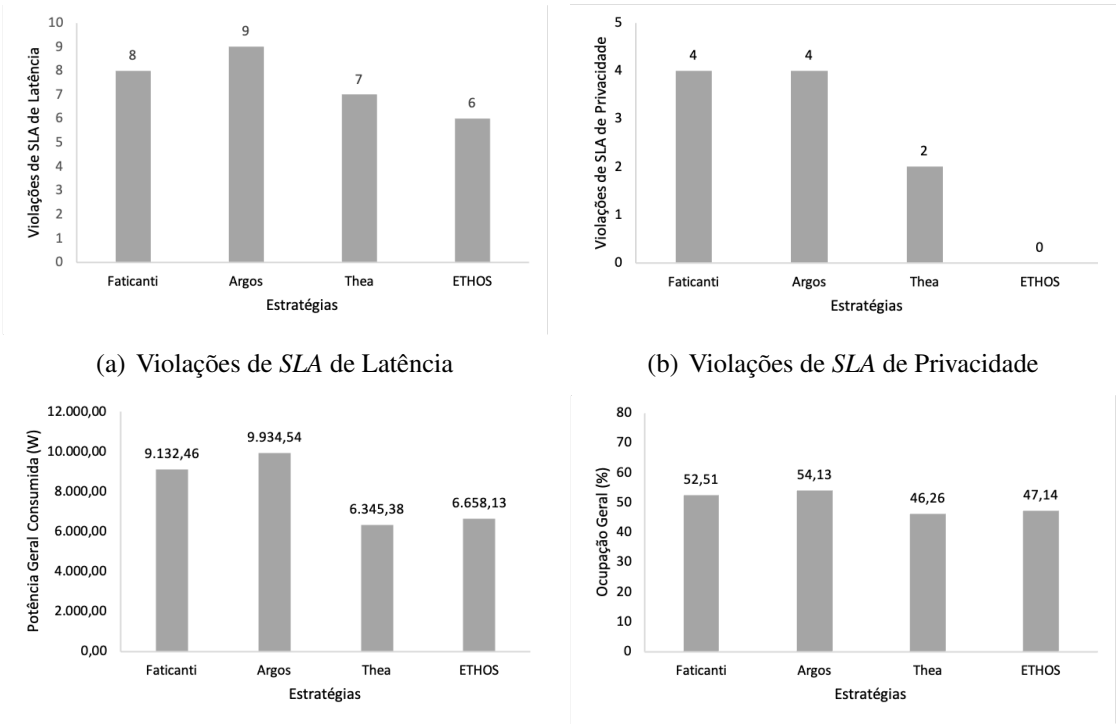
Utilizamos o mesmo cenário proposto em [Souza et al. 2023a] com 16 aplicações compostas de diferentes tamanhos (1, 2, 4 e 8 serviços), com diferentes requisitos de latência (3 e 6). Simulamos 4 aplicações por tamanho com 4 diferentes demandas de recursos de CPU e memória RAM (pequena: 2 *cores* - 2 GB; média: 4 *cores* - 4 GB; grande: 8 *cores* - 8 GB; extra-grande: 16 *cores* - 16 GB), totalizando 60 serviços com demandas heterogêneas. Definimos três níveis de requisitos de privacidade de serviço, nível 0 (sem requisito de privacidade), nível 1 (requisito de privacidade médio) e nível 2 (requisito de privacidade alto). Assumimos que uma violação de *SLA* de privacidade ocorre quando um serviço é hospedado por um servidor de borda que possui nível de confiabilidade menor do que o requisito de privacidade do serviço.

O objetivo da nossa estratégia é alocar os serviços de aplicações dos usuários em uma infraestrutura de borda federada de forma que tanto os requisitos de latência e de privacidade sejam atendidos. Desta forma, comparamos nossa abordagem com outras três estratégias de posicionamento de aplicações em borda federadas, Faticanti [Faticanti et al. 2020], Argos [Souza et al. 2022] e Thea [Souza et al. 2023a]. Assim como nossa proposta, estas estratégias fazem posicionamento de aplicações em bordas fe-

deradas buscando otimizar a latência e a privacidade. No entanto, estes 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.

6. Resultados

Avaliamos as estratégias baseadas nas métricas de violação de *SLA* de latência e violações de privacidade. As violações de *SLA* de latência se referem ao número serviços que tiveram sua latência excedida em relação ao limite exigido pela aplicação. Já as violações de privacidade estão relacionadas ao número de serviços que foram hospedados por servidores de borda que possuiam níveis de confiança abaixo do requisito de privacidade do serviço. Neste trabalho, o algoritmo faz a alocação *offline* das aplicações, ou seja, o estado e a lista de aplicações não se altera durante o ciclo de alocação. O algoritmo finaliza o ciclo de alocação após todas as aplicações daquele ciclo serem provisionadas.



(a) Violações de *SLA* de Latência

(b) Violações de *SLA* de Privacidade

(c) Potência Geral Consumida (W)

(d) Ocupação Geral (CPU e RAM) (%)

Figura 2. Comparação dos resultados das estratégias avaliadas.

6.1. Violações de *SLA* de latência

A Figura 2(a) apresenta os resultados da violação de *SLA* de latência das estratégias avaliadas. As estratégias Argos e de Faticante tiveram os maiores números de violações de *SLA* de latência. Argos utiliza a heurística *First-Fit* como estratégia de posicionamento inicial das aplicações e, com isso, seleciona o primeiro servidor de borda com recursos disponíveis suficiente para atender a demanda do serviço. Desta forma, o Argos

desperdiça recursos dos servidores de borda afetando aplicações com demandas maiores. Por sua vez, a estratégia de Faticanti aloca os serviços de todos os aplicativos de forma simultânea, ordenando os serviços com base em suas ordens de execução dentro de cada aplicação. Diferentemente das estratégias anteriores, tanto ETHOS como o Thea ordenam os serviços de forma a atender as exigências de latência e privacidade, selecionando servidores de borda que entregam recursos e nível de privacidade mais próximos aos requisitos do serviço, evitando, assim, o desperdício de recursos. Já ETHOS obteve vantagem sobre o Thea (14% mais eficiente), e em relação ao Argos e Faticanti, 43,4% e 35% respectivamente. Isso se deve ao fato de que avaliamos o custo de cada servidor de borda, independentemente do nível de confiabilidade dos provedores. Desta forma, ampliamos a quantidade de servidores considerados possíveis candidatos para hospedar cada serviço, não limitando apenas a um único provedor.

6.2. Violações de privacidade

A Figura 2(b) mostra o número de violações de *SLA* de privacidade. As estratégias Argos e de Faticanti apresentaram os piores resultados, pois alocam de forma indiscriminada serviços com baixo requisito de privacidade em servidores de borda com nível de confiança alto. Desta forma, servidores de borda com nível de confiança mais alto começam a ser provisionados com serviços que possuem requisitos de privacidade baixos, e serviços com requisitos de privacidade mais altos precisam ser alocados em servidores de borda que não atendem ao nível mínimo de confiança, acarretando em violações de *SLA* de privacidade.

A estratégia Thea conseguiu diminuir em 50% as violações de privacidade em relação a de Faticanti e o Argos, pois define a ordem de colocação priorizando aplicações maiores compostas por serviços com requisitos de privacidade mais altos. Apesar do Thea evitar o provisionamento de serviços em servidores de borda cuja a confiabilidade excede o requisito de privacidade dos serviços, a estratégia possui uma limitação com relação a granularidade dos níveis de confiabilidade dos servidores de borda, acarretando ainda em violações privacidade. Nos experimentos realizados, ETHOS não viola o *SLA* de privacidade, já que adota uma granularidade maior dos níveis de confiabilidade e extendemos a atribuição do nível de confiabilidade para cada servidor de forma individual. Desta forma, ETHOS consegue ampliar a busca por recursos confiáveis em toda extensão da borda federada.

Apesar de não ser o foco da nossa proposta, comparamos ETHOS com as demais estratégias avaliadas com outras métricas disponíveis na simulação. A Figura 2(c) mostra a potência total consumida ao final do ciclo de posicionamento, mostrando que ETHOS consumiu apenas 5,7% a mais de potência que o Thea, que obteve o melhor resultado. Já a Figura 2(d) mostra o acumulado da ocupação geral dos recursos (CPU e RAM) da infraestrutura. A ocupação geral é dada pela proporção entre a demanda das aplicações, em relação ao consumo de recursos de CPU e memória RAM, e a capacidade geral dos servidores que compõem a infraestrutura de borda. O ETHOS obteve uma taxa de ocupação geral dos recursos em torno de 47%, apenas 1% a mais que o Thea, que também obteve o melhor resultado. Isto quer dizer que ETHOS consegue provisionar os serviços em uma infraestrutura de borda federada de forma a atender os requisitos de privacidade das aplicações com um mínimo de violações de *SLA* de latência, sem que outras métricas sejam sacrificadas, demonstrando a viabilidade da nossa proposta.

7. Conclusões e Trabalhos Futuros

Apesar do potencial da computação de borda, a limitação na escalabilidade vertical de recursos pode prejudicar o provisionamento de aplicações com alta demanda de recursos. Estudos anteriores proporam abordagens baseadas em federação de bordas entre diferentes provedores de infraestrutura para melhorar o *QoS* dos usuários, além de otimizar o uso dos recursos. Alguns estudos recentes implementam estratégias de posicionamento de aplicações em bordas federadas com o objetivo de melhorar o desempenho das aplicações e, ao mesmo tempo, atender aos requisitos de privacidade.

No entanto, estas estratégias possuem algumas limitações relacionadas ao processo de selecionar o melhor servidor de borda para alocar os serviços. Neste trabalho, apresentamos o ETHOS, uma heurística baseada em custo dos servidores de borda para otimizar o posicionamento das aplicações, melhorando o *QoS* dos usuários, além de otimizar a utilização dos recursos.

Experimentos simulados mostram que ETHOS consegue reduzir as violações de *SLA*s de latência entre 14,3% e 43,4% em relação a outras abordagens. Em relação à violação de *SLA*s de privacidade, ETHOS conseguiu diminuir em até 100% a quantidade de violações de privacidade. Além disso, os resultados mostram que ETHOS não sacrifica outras métricas, como a potência consumida e a taxa de ocupação dos recursos, demonstrando a viabilidade da nossa estratégia. Em trabalhos futuros, pretende-se estender ETHOS para fazer o posicionamento *online* das aplicações em uma borda federada, assim como definir métricas para avaliar em tempo real a confiabilidade dos servidores de borda.

Referências

Apat, H. K., Nayak, R., and Sahoo, B. (2023). A comprehensive review on internet of things application placement in fog computing environment. *Internet of Things*, 23:100866.

Aral, A., De Maio, V., and Brandic, I. (2021). Ares: Reliable and sustainable edge provisioning for wireless sensor networks. *IEEE Transactions on Sustainable Computing*, 7(4):761–773.

Brik, B., Frangoudis, P. A., and Ksentini, A. (2020). Service-oriented mec applications placement in a federated edge cloud architecture. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6.

Buyya, R., Srirama, S. N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., Gelenbe, E., Javadi, B., Vaquero, L. M., Netto, M. A. S., Toosi, A. N., Rodriguez, M. A., Llorente, I. M., Vimercati, S. D. C. D., Samarati, P., Milojicic, D., Varela, C., Bahsoon, R., Assuncao, M. D. D., Rana, O., Zhou, W., Jin, H., Gentzsch, W., Zomaya, A. Y., and Shen, H. (2018). A manifesto for future generation cloud computing: Research directions for the next decade. 51(5).

Faticanti, F., Savi, M., Pellegrini, F. D., Kochovski, P., Stankovski, V., and Siracusa, D. (2020). Deployment of application microservices in multi-domain federated fog environments. In *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6.

Feng, C., Han, P., Zhang, X., Yang, B., Liu, Y., and Guo, L. (2022). Computation offloading in mobile edge computing networks: A survey. *Journal of Network and Computer Applications*, 202:103366.

He, X., Jin, R., and Dai, H. (2020). Peace: Privacy-preserving and cost-efficient task offloading for mobile-edge computing. *IEEE Transactions on Wireless Communications*, 19(3):1814–1824.

Institute, E. T. S. (2023). Multi-access edge computing (mec).

Ismail, L. and Materwala, H. (2021). Escove: energy-sla-aware edge–cloud computation offloading in vehicular networks. *Sensors*, 21(15):5233.

Jeong, Y., Maria, E., and Park, S. (2021). Towards energy-efficient service scheduling in federated edge clouds. *Cluster Computing*, pages 1–13.

Kaiser, S., Haq, M. S., Tosun, A. , and Korkmaz, T. (2022). Container technologies for arm architecture: A comprehensive survey of the state-of-the-art. *IEEE Access*, 10:84853–84881.

Kar, B., Yahya, W., Lin, Y.-D., and Ali, A. (2023). Offloading using traditional optimization and machine learning in federated cloud–edge–fog systems: A survey. *IEEE Communications Surveys Tutorials*, 25(2):1199–1226.

Lin, L., Liao, X., Jin, H., and Li, P. (2019). Computation offloading toward edge computing. *Proceedings of the IEEE*, 107(8):1584–1607.

Souza, P., Kayser, C., Roges, L., and Ferreto, T. (2023a). Thea - a qos, privacy, and power-aware algorithm for placing applications on federated edges. In *2023 31st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 136–143.

Souza, P., Vieira, Â. N. C., Rubin, F., Ferreto, T., and Rossi, F. D. (2022). Latency-aware privacy-preserving service migration in federated edges. In *CLOSER*, pages 288–295.

Souza, P. S., Ferreto, T., and Calheiros, R. N. (2023b). Edgesimpy: Python-based modeling and simulation of edge computing resource management policies. *Future Generation Computer Systems*, 148:446–459.

Wang, J., Feng, Z., George, S., Iyengar, R., Pillai, P., and Satyanarayanan, M. (2019a). Towards scalable edge-native applications. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 152–165.

Wang, S., Zhao, Y., Xu, J., Yuan, J., and Hsu, C.-H. (2019b). Edge server placement in mobile edge computing. *Journal of Parallel and Distributed Computing*, 127:160–168.

Xia, J., Cheng, G., Guo, D., and Zhou, X. (2020). A qoe-aware service-enhancement strategy for edge artificial intelligence applications. *IEEE Internet of Things Journal*, 7(10):9494–9506.