

# Fatias (Leves) como um Serviço: Uma Solução para Dispositivos Móveis na Borda da Rede

Lucas B. Oliveira<sup>1</sup>, Alan Veloso<sup>1</sup>, Denis Rosário<sup>1</sup>, Antônio Abelém<sup>1</sup>

<sup>1</sup> Programa de Pós-Graduação em Ciência da Computação (PPGCC)  
Universidade Federal do Pará (UFPA) – Belém – PA – Brasil

lucas.borges@itec.ufpa.br, alan.veloso@icen.ufpa.br  
{denis, abelem}@ufpa.br

**Abstract.** *Cloud Network Slicing (CNS) is the process of isolating end-to-end, on-demand, independently controlled, managed, and/or orchestrated cloud and network resources, which are called slices. In this context, one can consider devices with low computational power, such as mobile devices, to provide resources for slices, thus bringing services even closer to the end users. However, platforms that provide Slice as a Service (SlaaS) do not allocate mobile devices as part of the slice. In this sense, this paper presents a SlaaS platform that allocates mobile devices as part of the slice. For this, lightweight virtualization is used instead of full virtualization to do the slice instantiation. The results show it is possible to allocate mobile devices as part of the slice. The results obtained show that the solution is promising because of better performance in lightweight virtualization compared to full virtualization.*

**Resumo.** *Fatiamento de Rede e Nuvem (Cloud Network Slicing - CNS) é o processo capaz de isolar fim-a-fim e sob-demanda recursos de nuvem e rede controlados, gerenciados e/ou orquestrados independentemente, os quais são denominados de fatias. Nesse contexto, pode-se considerar dispositivos com baixo poder computacional, como os dispositivos móveis, para prover recursos para as fatias, e assim deixando os serviços ainda mais próximos dos usuários finais. Contudo, as plataformas que fornecem Fatia como um Serviço (Slice as a Service - SlaaS) não aloca dispositivos móveis como parte da fatia. Nesse sentido, este trabalho apresenta uma plataforma de SlaaS que aloca dispositivos móveis como parte da fatia. Para isso, utiliza-se a virtualização leve ao invés da virtualização completa para fazer a instanciação da fatia. Os resultados mostram ser possível alocar dispositivos móveis como parte da fatia. Os resultados obtidos mostram que a solução é promissora por conta do desempenho melhor na virtualização leve em comparação com a virtualização total.*

## 1. Introdução

Tecnologias como Redes Definidas por Software (do inglês, *Software Defined Networking* - SDN) e Virtualização das Funções da Rede (do inglês, *Network Functions Virtualization* - NFV) permitem construir uma arquitetura de rede de computadores programável e dinâmica sob demanda. Aliado ao conceito de computação em nuvem, surgiu o conceito de Fatiamento de Rede e Nuvem (*Cloud Network Slicing* - CNS) que engloba as tecnologias de rede e nuvem, viabilizando novos serviços [de Almeida et al. 2020]. Por exemplo, um serviço de rede de fornecimento de conteúdo turístico que entrega conteúdo para o usuário

com base na sua localização, ou um serviço que possibilita o monitoramento da carga em contêiner utilizando também dispositivos de Internet das Coisas [NECOS project 2019b].

CNS trata-se do processo de alocar recursos de rede e nuvem de ponta a ponta, criando fatias isoladas com conjuntos de recursos de infraestrutura de nuvem (computação e armazenamento) e funções de rede de provedores (conectividade) [Maciel et al. 2019]. Desta forma, os aplicativos de serviço que apresentam operações e atributos especialmente concebidos podem ser atendidos [Clayman et al. 2021].

O projeto NECOS<sup>1</sup> (Novel Enablers for Cloud Slicing) cunhou o termo CNS oferecendo o modelo de Fatia como um Serviço (do inglês, *Slice as a Service* - SaaS) [Silva et al. 2018]. O modelo SaaS aborda o ciclo de vida completo (da criação até o descomissionamento) de instâncias de fatia de rede e nuvem sob demanda em tempo de execução, permitindo o provisionamento de serviços altamente isolados de multilocação de ponta a ponta [Clayman et al. 2021]. A alocação desses recursos é realizada em organizações especializadas em fornecer esses recursos de rede, de nuvem ou ambos.

Contar com o processamento e armazenamento pessoal do usuário na borda da rede para serviços a terceiros é comum. Exemplos disso são: o roteador dos dispositivos móveis, que possibilita o compartilhamento de acesso à rede móvel, projetos de computação distribuída, onde os doadores oferecem voluntariamente tempo de computação de computadores pessoais para uma causa específica [Mengistu and Che 2019], até mesmo *blockchains* podem ser consideradas como computação voluntária, Bitcoin e Ethereum, pois utilizam os computadores dos membros da rede para realizar validações [Casino et al. 2019]. Portanto, é possível contar com a colaboração consciente do usuário para realizar computação e armazenamento [Mengistu and Che 2019], no caso deste trabalho, para alocar uma fatia de rede e nuvem. Dado o crescimento da capacidade de processamento apresentada pelos dispositivos móveis (do inglês, *Mobile Devices* - MDs), eles podem ser recursos disponíveis para uma fatia. Isso deixará os serviços ainda mais próximos dos usuários finais. A proposta pode ser aplicada em um cenário onde um motorista de aplicativo aloca um serviço de terceiros no seu MD para fornecer aos seus passageiros, como, por exemplo, um serviço de propaganda baseado na localidade.

SaaS utiliza virtualização para construir uma fatia podendo utilizar diferentes dispositivos computacionais, tais como dispositivos de rede e servidores [Silva et al. 2018]. Comumente, para realizar essa virtualização vem sendo utilizada a virtualização total e assim, faz-se necessário que os dispositivos pertencentes a fatia consigam realizar a virtualização total. Especificamente, a virtualização total consiste em emular os recursos físicos possibilitando a execução do seu próprio sistema operacional [Potdar et al. 2020]. Contudo, nem todos os dispositivos conseguem realizar a virtualização total, como no caso da maioria dos dispositivos móveis. Isso deve-se principalmente ao fato de que as máquinas virtuais são grandes necessitando de um poder computacional [Potdar et al. 2020] que nem todos os MDs podem disponibilizar, sem prejudicar seu uso. Então, visto que os MDs não têm essa capacidade, surge a seguinte questão de pesquisa: Como MDs podem ser inseridos em uma fatia?

Dado o problema apresentado acima, este trabalho propõe uma solução para uma plataforma de SaaS com base na arquitetura proposta na plataforma NECOS, utilizando

---

<sup>1</sup><http://www.h2020-necos.eu/>

virtualização leve para alocar fatias evitando a criação de uma pilha completa dedicado ao sistema operacional, como a virtualização total. Assim, dispositivos que comumente não podem realizar a virtualização total, como os móveis, passam a fazer parte da fatia. Então, um componente da arquitetura da plataforma foi alterado e para avaliar a participação de MDs na fatia, foram realizados dois processos avaliativos, viabilidade e desempenho do novo componente, comparando os métodos de instanciação. Resultados de implementação mostraram ser possível alocar MDs como parte das fatias utilizando virtualização leve, além disso, os resultados contribuem com a concepção de que a virtualização leve consome menos recursos que a virtualização total.

O restante deste trabalho está organizado da seguinte forma: A proposta de uma plataforma de SaaS responsável por alocar fatias de recursos é apresentada na Seção 2. Posteriormente são apresentados os detalhes do cenário de avaliação e os resultados na Seção 3. Por fim, a Seção 4 traz as considerações finais e trabalhos futuros.

## 2. Fatia como um Serviço para Dispositivos Móveis

Este trabalho propõe alterações na arquitetura do NECOS para incluir os MDs como parte da fatia, disponibilizando seus recursos de computação e redes. Desta forma, a seguir serão apresentados os componentes da plataforma NECOS. Em seguida, é descrita a proposta em si, destacando as mudanças realizadas na plataforma NECOS para incluir MDs nas fatias.

### 2.1. Novel Enablers for Cloud Slicing (NECOS)

Este trabalho é desenvolvido baseado em virtualização de recursos e estendendo a plataforma NECOS, a qual é apresentada na Figura 1. A plataforma NECOS é composta pelos componentes Tenant's Domain, NECOS Slice Provider, Resource Marketplace, e Resource Domain. A seguir são apresentados os principais módulos, onde mais detalhes podem ser encontrados em [NECOS project 2019a].

O componente **Tenant's Domain** é uma organização que exige fatias de rede em nuvem para que seus próprios serviços sejam executados, alguns módulos próprios que são específicos de acordo com o serviço ofertado: o *Service Orchestrator* responsável por orquestrar o serviço na fatia disponível e o *Service Activator*, que manda a descrição da fatia para o NECOS Slice Provider.

O componente **NECOS Slice Provider** é responsável por orquestrar o conjunto de recursos disponíveis e construir a fatia, o qual possui os módulos: *Slice Builder*, para criar a fatia a partir do conjunto de recursos disponíveis; *Slice Resource Orchestrator (SRO)*, para unir os recursos alocados e orquestrar e gerenciar todo o ciclo de vida da fatia; *Infrastructure & Monitoring Abstraction (IMA)* para prover uma visão única do monitoramento de recursos em domínios distintos.

O **Resource Marketplace** é o local de negociação dos recursos, o qual funciona de forma distribuída e possui os módulos *Slice Broker* e *Slice Agent*. O *Slice Broker* recebe as demandas e descobre o local que possui recursos disponíveis para serem alocados. Essa descoberta é efetuada através da interação com o *Slice Agent* que responde informando quais provedores e recursos estão disponíveis para fazer parte da fatia.

O **Resource Domain** é quem disponibiliza os seus recursos para a construção da fatia. Como os recursos podem ser de processamento, armazenamento e rede, dois

módulos são necessários para isso: *DC Slice Controller* responsável por criar fatias de processamento e armazenamento de dados; e *WAN Slice Controller* que cria as fatias de rede.

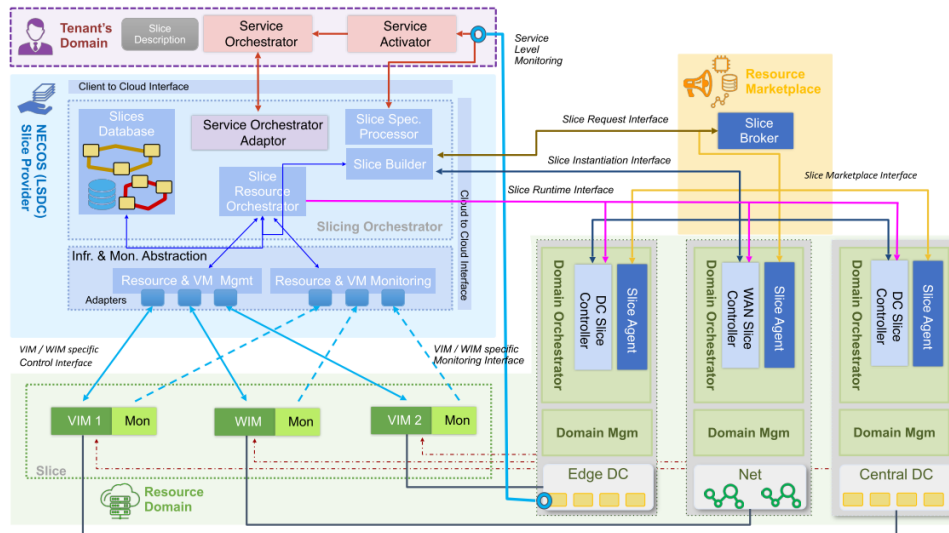


Figura 1. Arquitetura NECOS [NECOS project 2019a]

## 2.2. Arquitetura do NECOS aplicada à MD

Originalmente, a plataforma NECOS utiliza virtualização total para criação das fatias. Entretanto, esse modelo pode tornar inviável que dispositivos com menor poder computacional façam parte do repertório de recursos para a construção da fatia. Portanto, o módulo *DC Slice Controller* foi alterado para instanciar fatias mais leves, com foco no modelo de virtualização baseado em contêiner, dando suporte a construção de fatias em dispositivos mais modestos.

Então, dispositivos com menor poder computacional, como os MDs, compõem a plataforma NECOS, disponibilizando seus recursos para hospedar microsserviços sob demanda. Os MDs possuem incorporado processador, roteador e bateria elétrica. Portanto, esses componentes são suficientes para atuar como um centro móvel de processamentos de dados de baixa capacidade. Na arquitetura do NECOS, o MD atua como um *Resource Provider*, englobando os módulos necessários para o provisionamento de fatias. Assim, o MD é responsável por desempenhar os serviços como centro de dados da nuvem e provedor de serviços de rede.

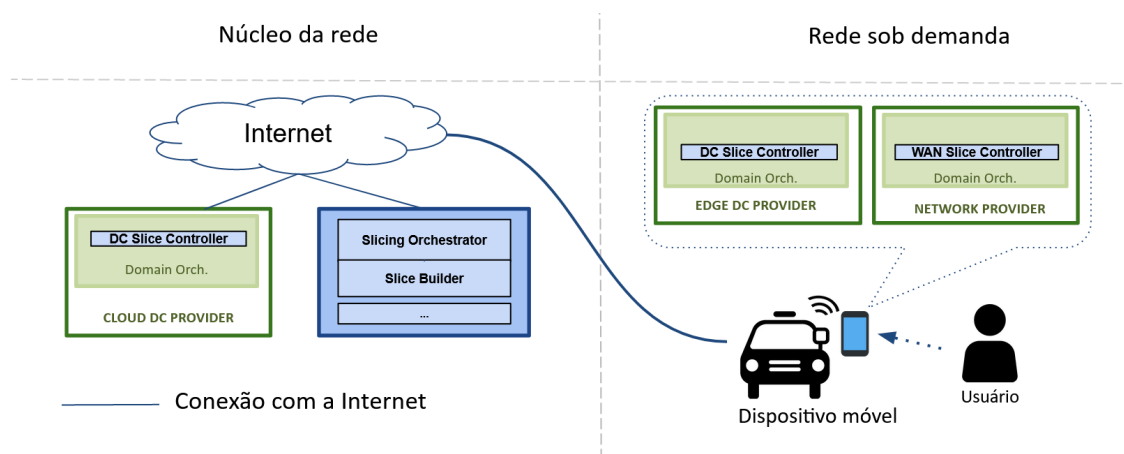
A plataforma NECOS possui diversos componentes, todos com suas respectivas funções e importância. Neste caso, a Figura 2 apresenta de maneira simplificada a arquitetura do NECOS adaptado ao cenário do MD, destacando os módulos mais importantes para a proposta. É possível identificar os *Resource Providers* em verde, distribuídos geograficamente, o *Slice Provider* em azul que possui o SRO e o *Slice Builder*. É possível observar momentos importantes no cenário. No primeiro momento, o MD recebe o serviço migrado do centro de dados principal. No segundo momento, o serviço encontra-se mais próximo do usuário, visto que está hospedado no MD e disponível através do seu ponto de acesso sem fio.

No cenário, o MD e o servidor no núcleo são os *Resource Providers*. O *DC Slice Controller*, presente neles, é responsável por criar fatias, alocando os recursos necessários de computação e armazenamento para uma determinada fatia. Além disso, ele lida com solicitações de fatias e verifica a possibilidade de criar uma, baseado na disponibilidade de recursos locais. Se a criação da fatia for possível, serão selecionados recursos do conjunto para serem alocados.

O roteador integrado no MD, atua como *WAN Slice Provider*. Nele está presente o *WAN Slice Controller*, que irá manipular as solicitações de fatias de rede e determinar se é possível criar uma no domínio com base na disponibilidade de recursos locais (por exemplo, largura de banda). Uma fatia de rede é um conjunto de enlaces virtuais que conecta duas fatias, então diante da necessidade de conectar as fatias instanciadas, o *WAN Slice Controller* fornece o conjunto de enlaces virtuais necessários.

O *Marketplace*, como dito na seção anterior, é um sistema distribuído com dois componentes. O registro dos recursos disponíveis para alocação é responsabilidade do *Slice Broker*, que se comunica constantemente com o *Slice Agent* presente nos provedores de recurso. Desse modo, ao receber as requisições do *Slice Provider*, a criação da fatia é realizada obedecendo aos requisitos dos *Tenants*.

O *Slice Provider* é inserido como o responsável por realizar a manipulação das fatias, da criação à gerência. Encarregado de solicitar ao *Marketplace* as diferentes partes da fatia. O *Slice Builder* executa uma orquestração inicial para elaborar um subconjunto de domínios que possam ser utilizados para criação da fatia. Então, após esse processo o *Slice Orchestrator* combina as partes alocadas em uma única fatia agregada, em que mantém a visão geral de fim a fim e gerência o ciclo de vida de cada parte da fatia individual.



**Figura 2. Arquitetura do NECOS aplicada a MDs**

Os *Resource Providers* ao ingressarem no ambiente, cedem seus recursos de rede, armazenamento e processamento físicos para a criação de entidades virtuais, através do *WAN Slice Controller* e *DC Slice Controller*. Gerindo e disponibilizando ao ambiente os recursos de maneira otimizada, eles podem utilizar máquinas virtuais ou contêineres para prover a virtualização a um nível desejado e assim permitir a instanciação de fatias considerando a diferença entre as técnicas e o cenário.

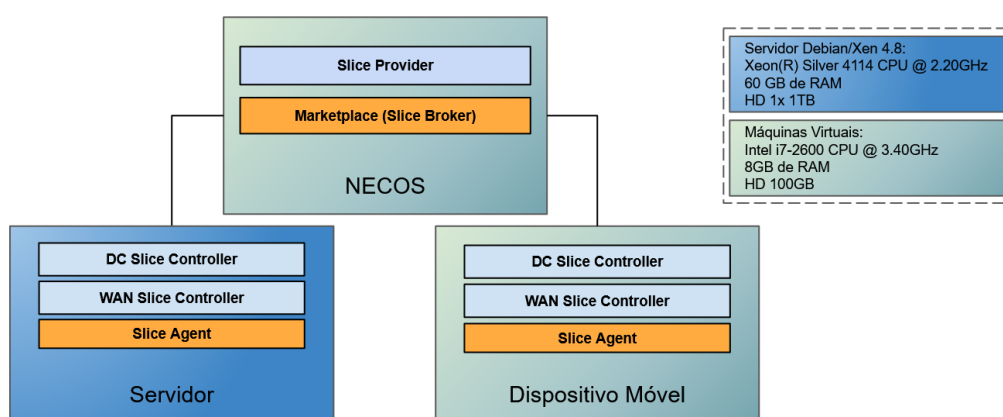
### 3. Avaliação

O processo de avaliação da proposta foi realizado em 2 etapas: a primeira foi a avaliação de viabilidade, para constatar se seria possível um dispositivo móvel instanciar um contêiner da fatia. A segunda etapa foi responsável por realizar uma comparação entre os diferentes métodos de virtualização para comparar o desempenho e observar possíveis limitações da virtualização leve.

#### 3.1. Cenário de avaliação

Para realizar a avaliação de viabilidade foi considerado um MD da marca Xiaomi, modelo Redmi Note 7, com 3 GB de RAM, CPU Octa-core Max 2.2GHz, sistema operacional Android versão 10 QKQ1.190910.002. Nesse dispositivo foi possível realizar a virtualização leve, assim o contêiner com um serviço foi instanciado com êxito. Por outro lado, devido às limitações do MD, não foi possível realizar a avaliação comparativa das técnicas de virtualização total e leve. Portanto, de modo a reproduzir um ambiente similar ao cenário da Figura 2 foi criado um ambiente de testes utilizando máquinas físicas e virtuais para representar os recursos do cenário. Sobre o método de avaliação, foram utilizados os critérios fornecidos pela ISO 25023, ela propõe métricas para avaliação de características dos produtos de software.

Os componentes do cenário de teste podem ser observados na Figura 3. Nela encontra-se uma máquina física atuando como servidor, que disponibiliza seus recursos para a criação de fatias, através dos componentes, *DC Slice Controller* e o *Slice Agent* da plataforma NECOS. Outro nó é o “NECOS”, ele é uma VM que hospeda os componentes, *Slice Provider* e *Marketplace (Slice Broker)*, responsáveis por gerir e implementar os serviços aos dispositivos conectados no ambiente. Semelhantemente, a VM que representa o MD tem funções semelhantes ao nó “Servidor”, ou seja, atuando como *Resource Provider* e disponibilizando-os através dos componentes, *Slice Agent*, *WAN Slice Controller* e *DC Slice Controller*, responsáveis pela instanciação de fatias e criação de enlaces virtuais entre elas.



**Figura 3. Infraestrutura utilizada na avaliação.**

Os resultados para análise foram obtidos através da realização do processo completo de provisionamento de fatias, ou seja, seguindo todos os passos para a instanciação de uma fatia, da requisição até a sua remoção, utilizando primeiramente VMs e posteriormente contêineres. Assim, foi possível obter o tempo total e os recursos de memória

alocados para a inicialização de uma fatia, permitindo conhecer as características de todo ciclo de vida de uma fatia em um ambiente controlado, sem tráfego de fundo.

Além disso, frente a possibilidade de dividir a instanciação da fatia em subprocessos, os resultados puderam ser apresentados separadamente em ordem cronológica dos acontecimentos. Dessa forma, pode-se realizar a análise a cada etapa. Então, a partir dos resultados obtidos em cada subprocesso, foram gerados gráficos correspondentes à quantidade de testes realizados em relação ao tempo decorrido para conclusão do subprocesso.

### 3.2. Resultados

A inicialização do processo se dá através da requisição de uma fatia pelo *tenant*, esse é o processo mais rápido e simples executado na plataforma. Posteriormente, é realizada a extração dos requisitos necessários e verificado se existem recursos disponíveis para serem alocados no instante de tempo que o *tenant* deseja. Ao verificar o banco de dados, é retornado ao *tenant* a resposta da solicitação, que em caso positivo, os recursos são reservados para aquela fatia.

O tempo de requisição é composto pelo tempo de comunicação entre os componentes e pelo tempo de leitura dos bancos de dados presentes no *Marketplace* e *DC Slice Controller*. Contudo, mesmo não existindo uma relação direta no modo em como as tecnologias são desenvolvidas, a Figura 4(a) apresenta uma diferença relevante no tempo de resposta às requisições, provavelmente decorrente das adaptações realizadas no ambiente para utilização de contêineres.

No cenário utilizando VMs, é possível observar o tempo de resposta ficando próximo a um segundo, abaixo do tempo em que os contêineres foram utilizados para prover os serviços. Além disso, vale destacar a constante variação no tempo de resposta no cenário com os contêineres e uma janela de tempo muito ampla, frente à utilização das VMs que apresentou constância e previsibilidade.

Após os recursos serem reservados, a próxima etapa no processo é a criação e inicialização da fatia. A inicialização se dá através do clone de um modelo. Este modelo pode ser a imagem de uma VM ou de um contêiner. Assim, após o clone, as VMs ou os contêineres são iniciados e disponibilizados ao *tenant* que então pode dar início ao serviço requerido.

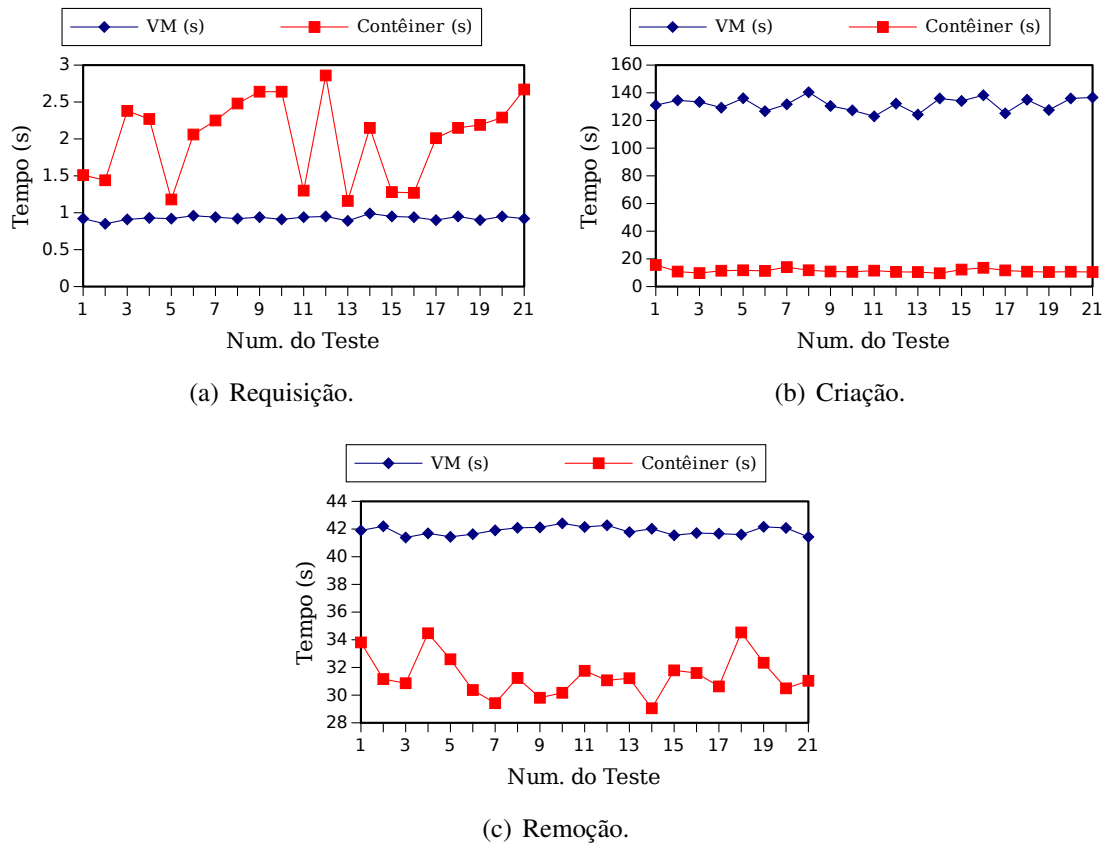
Para o processo de criação, o tempo de execução é composto pelo tempo de resposta e comunicação entre os componentes, clone das imagens e inicialização do ambiente clonado. Assim, lembrando dos resultados do processo anterior, é possível inferir que o tempo de comunicação entre os componentes é baixo. Diante disso, a maioria do tempo é consumida durante o processo de clone e inicialização dos sistemas.

Fica evidente na Figura 4(b) a diferença entre as tecnologias envolvidas, as fatias instanciadas utilizando VMs, consumiram próximo do quádruplo do tempo se comparadas às fatias construídas com contêineres. Diante do gráfico mostrado na Figura 4(b) e lembrando da arquitetura das tecnologias, o tempo consumido no processo é decorrente do tamanho da imagem, formato e o intervalo de inicialização para cada uma das técnicas.

Com a fatia iniciada e em execução, o MD pode disponibilizar serviços mais próximos do usuário em outros ambientes. Através do ponto de acesso sem fio, os serviços que estão sendo executados nele podem ser acessados e consumidos. No final, a fatia

pode ser finalizada e liberar os recursos para novas solicitações.

Neste trabalho, a remoção da fatia é um processo simples, não foram implementadas questões de sincronização de conteúdo em caso de desconexão do MD, substituição ou cópia. Sendo assim, o tempo do processo de remoção é composto somente pelo tempo que os sistemas são finalizados. Então, percebe-se na Figura 4(c), a variabilidade no tempo para a remoção da fatia instanciada por contêineres é mais eficiente que a com VMs.



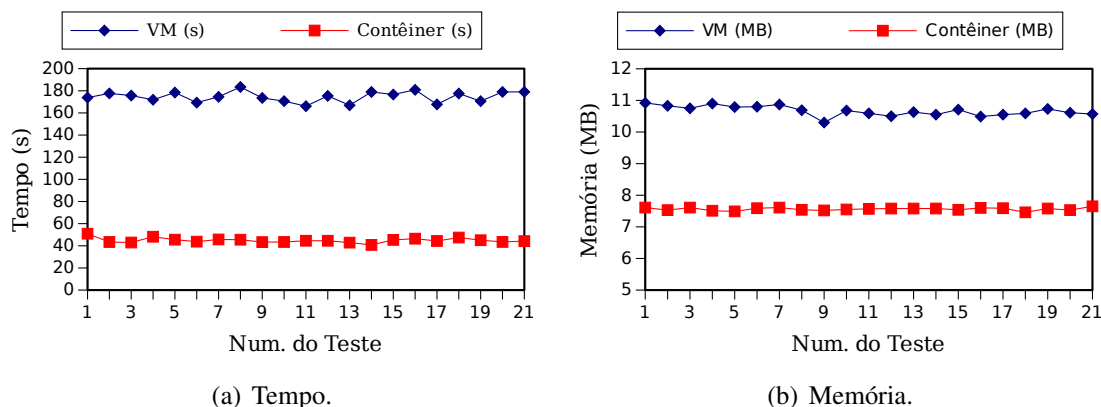
**Figura 4. Tempo para manipulação de fatias.**

Mesmo diante das variações apresentadas pelos contêineres no processo de requisição e remoção das fatias, o tempo total do processo não foi afetado, permanecendo constante e abaixo de um minuto. Por outro lado, a previsibilidade nos processos de requisição e remoção da fatia com VM, não foi suficiente para impedir que a variação ocorrida no processo de criação da fatia influenciasse na variação do tempo total, como mostra o gráfico na Figura 5(a).

A instanciação de fatias que utilizam VMs, além de consumir mais tempo para serem processadas, possuem também maior espaço reservado para alocação na memória. O gráfico da Figura 5(b), apresenta a média de memória alocada pelo sistema operacional do hospedeiro para atender as requisições dos três momentos dos cenários, assim é possível perceber que a utilização de contêineres é menos custosa aos componentes físicos do hospedeiro do que a utilização de VMs.

Reunindo as informações obtidas anteriormente, a Tabela 1 dispõe de informações





**Figura 5. Totalização de gastos.**

estatísticas de memória alocada e dos tempos obtidos durante a execução dos processos de requisição, criação e remoção no ambiente. Através dela, pode-se observar os resultados obtidos, de modo a permitir comparações entre as tecnologias analisadas. As informações relacionadas ao tempo e alocação de recursos, estão formatadas em segundos e megabytes, respectivamente.

**Tabela 1. Informações estatísticas dos resultados. Dados do Intervalo de Confiança estão expressos a 95% de confiança.**

Tecnologia	Estatística	Tempo dos Processos (s)				Alocação de Memória (em MB)
		Requisição	Criação	Remoção	Total	
Máquina Virtual	Média	0.93	131.85	41.87	174.65	10.67
	Variância	0.00	23.44	0.09	23.02	0.02
	Desvio Padrão	0.03	4.73	0.30	4.68	0.15
	Intervalo de Confiança	0.94	133.87	41.99	176.65	10.73
		0.92	129.83	41.74	172.64	10.60
Contêiner	Média	2.01	11.45	31.40	44.86	7.56
	Variância	0.31	2.08	2.21	4.66	0.00
	Desvio Padrão	0.55	1.44	1.49	2.16	0.04
	Intervalo de Confiança	2.24	12.05	32.02	45.76	7.58
		1.78	10.84	30.78	43.95	7.54

A diferença no uso de VMs e contêineres fica evidente ao observar o tempo de criação da fatia e tempo total do processo. Diante da inicialização de um ambiente completo, as fatias instanciadas utilizando VMs consumiram um tempo muito superior ao das fatias construídas com contêineres. Por outro lado, a utilização de contêineres proporcionou uma economia de cerca de 75% ao tempo de instanciação em todo o ambiente.

Portanto, diante dos resultados e características apresentadas de cada tecnologia, é possível inferir que a instanciação de fatias utilizando VMs não é adequada para um cenário com baixo poder computacional. Por outro lado, a utilização de contêineres se mostra uma proposta promissora, de modo que, pode ser implementada como alternativa de virtualização à plataforma NECOS e permitir dispositivos de pequeno porte além dos MDs, como, por exemplo, em Raspberry Pi ou outros dispositivos.

#### 4. Conclusões e Trabalhos Futuros

Este trabalho alcançou o seu objetivo ao inserir novas funcionalidades em um plataforma de SaaS permitindo que MDs façam parte da construção de uma fatia e hospedem serviços sob demanda. Este trabalho realizou toda a instanciação da plataforma NECOS para avaliar comparativamente a técnica de virtualização utilizada pela plataforma e a solução de virtualização baseada em contêiner, buscando a inserção do MD como membro de uma fatia, de forma que disponibilizasse seus recursos. Assim, os tempos de instanciação das fatias formadas por VMs e contêineres foram analisados e comparados visando propor a melhor técnica a ser usada no ambiente.

Como uma forma de incentivar o usuário a disponibilizar os recursos ociosos do seu MD, um sistema de recompensa pode ser implementado. Esse sistema pode recompensar o usuário baseado no tempo de utilização dos seus recursos.

#### Referências

- Casino, F., Dasaklis, T. K., and Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Teleinformatics and Informatics*, 36(May 2018):55–81.
- Clayman, S., Neto, A., Verdi, F., Correa, S., Sampaio, S., Sakelariou, I., Mamatas, L., Pasquini, R., Cardoso, K., Tusa, F., Rothenberg, C., and Serrat, J. (2021). The NECOS Approach to End-to-End Cloud-Network Slicing as a Service. *IEEE Communications Magazine*, 59(3):91–97.
- de Almeida, L. C., Maciel, P. D., and Verdi, F. L. (2020). Cloud Network Slicing: A systematic mapping study from scientific publications.
- Maciel, P. D., Verdi, F. L., Valsamas, P., Sakellariou, I., Mamatas, L., Petridou, S., Papadimitriou, P., Moura, D., Swapna, A. I., Pinheiro, B., and Clayman, S. (2019). A Marketplace-based Approach to Cloud Network Slice Composition Across Multiple Domains. *Proceedings of the 2019 IEEE Conference on Network Softwarization: Unleashing the Power of Network Softwarization, NetSoft 2019*, pages 480–488.
- Mengistu, T. M. and Che, D. (2019). Survey and taxonomy of volunteer computing. *ACM Computing Surveys*, 52(3).
- NECOS project (2019a). D3.2: NECOS System Architecture and Platform Specification. V2.
- NECOS project (2019b). MUlti-Slice/Tenant/Service (MUSTS).
- Potdar, A. M., Narayan, D. G., Kengond, S., and Mulla, M. M. (2020). Performance Evaluation of Docker Container and Virtual Machine. *Procedia Computer Science*, 171(2019):1419–1428.
- Silva, F. S., Lemos, M. O., Medeiros, A., Neto, A. V., Pasquini, R., Moura, D., Rothenberg, C., Mamatas, L., Correa, S. L., Cardoso, K. V., Marcondes, C., Abelem, A., Nascimento, M., Galis, A., Contreras, L., Serrat, J., and Papadimitriou, P. (2018). NECOS Project: Towards Lightweight Slicing of Cloud Federated Infrastructures. *2018 4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018*, pages 283–287.