

# Prova de Conceito de Gerenciadores de Infraestruturas Virtualizadas como Serviço Explorando o Sistema de Alocação de Recursos e Experimentação do FIBRE

Vinícius Gonçalves Braga<sup>1</sup>, João Paulo Esper<sup>1</sup>, Murillo Silva e Nunes<sup>1</sup>  
Elton Vivot Dias<sup>1</sup>, Sand Luz Corrêa<sup>1</sup>, Kleber Vieira Cardoso<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)

{viniciusbraga, eltondias, joaopauloesper, murillonunes}@inf.ufg.br

{sand, kleber}@inf.ufg.br

**Abstract.** *This paper presents an assessment of the feasibility of deploying on-demand Virtualized Infrastructure Managers (VIMs), or VIM as a service, in the context of network slicing, where more than one VIM may be instantiated simultaneously. To perform this validation, we use the FIBRE's resource allocation and experimentation system. Experimental results show that it is feasible to instantiate VIMs on demand in these scenarios, even when dealing with more complex VIMs such as OpenStack.*

**Resumo.** *Este artigo apresenta uma avaliação da viabilidade de se implantar VIMs sob demanda (VIMs como um serviço) no contexto de fatiamento de redes, onde mais de um VIM pode ser instanciado simultaneamente. Para realizar essa validação, utilizamos o sistema de alocação de recursos e experimentação do FIBRE. Os resultados experimentais mostram que é viável instanciar VIMs sob demanda nestes cenários, mesmo quando lidando com VIMs mais complexos, como o OpenStack.*

## 1. Introdução

Um dos maiores desafios das redes 5G é permitir que diferentes tipos de provedores de serviço, por exemplo, indústria automotiva, entretenimento e manufatura, com necessidades muito distintas, operem sobre a mesma infraestrutura física de rede [Contreras and López 2018]. Para tratar esse problema, nos últimos anos, o conceito de fatiamento de rede (*network slicing*) [Third Generation Partnership Project (3GPP) 2017] foi proposto como um meio de criar partições lógicas e funcionais de infraestruturas de rede, computação e armazenamento, adaptadas ao tipo de serviço a ser fornecido. Cada fatia de rede representa um serviço fim-a-fim independente, permitindo que provedores de rede implementem diferentes arquiteturas simultaneamente. Assim, o conceito de fatiamento de rede é uma abstração natural para separar vários inquilinos (*tenants*) ou provedores de serviço que operam sobre os mesmos recursos físicos. Para realizar o conceito de fatiamento de rede, tecnologias como Computação em Nuvem, Redes Definidas por Software (SDN) e Virtualização de Funções de Rede (NFV) têm sido amplamente exploradas.

Uma visão complementar à abstração de fatias de redes instanciadas dinamicamente e otimizadas para atender a um determinado serviço ou categoria de serviços, é a

ideia de prover Gerenciadores de Infraestruturas Virtualizadas (*Virtualized Infrastructure Managers* - VIMs) como um serviço, ou seja, *VIM-as-a-service*. Nessa visão, introduzida em [Stuart Clayman 2017], um VIM é instanciado sob demanda e implantado automaticamente para controlar e gerenciar os recursos de uma determinada fatia de rede. Diferentemente das configurações tradicionais de computação em nuvem, em que um único VIM é geralmente instanciado para todo o *data center*, um VIM sob demanda é implantado dinamicamente para cada fatia de rede. Como consequência, o próprio VIM se torna um parâmetro de configuração da fatia e pode ser escolhido de acordo com as necessidades do serviço. Por exemplo, alguns serviços podem ser melhor atendidos por ferramentas de virtualização de baixo nível, como Xen ou KVM, enquanto para outros, plataformas de contêiner, como o Docker ou o Kubernetes, são mais apropriadas. Outra vantagem da criação de VIMs sob demanda é que eles podem ser instanciados no topo da fatia de rede e, como tal, não precisam estar cientes dessa abstração. Como consequência, soluções como OpenStack, Kubertenes e OpenVIM podem ser usadas sem modificações.

Em [Stuart Clayman 2017] são apresentados os elementos arquiteturais necessários para implementar um VIM como um serviço no contexto de fatiamento de rede. Entretanto, o autor não apresenta nenhuma prova de conceito que possa validar a viabilidade dessa solução. Aproveitando essa lacuna, neste trabalho, exploramos o novo sistema de alocação de recursos e experimentação do FIBRE para validar essa proposta. Para isto, implantamos, através de *templates* genéricos e outros recursos providos pelo sistema, três VIMs distintos sob demanda, a saber: OpenStack, Kubernetes e *Very Lightweight Network & Service Platform* (VLSP). Os resultados experimentais mostram que é viável instanciar VIMs sob demanda para gerenciar os recursos de uma fatia de rede mesmo quando lidamos com VIMs sofisticados, como o OpenStack. Para ilustrar, conseguimos entregar um OpenStack completamente funcional em quatro nós em menos de 10 minutos. Os resultados também mostram que o sistema de alocação de recursos do FIBRE forneceu uma plataforma eficiente, de fácil reprodutibilidade e altamente escalável durante a realização dos experimentos.

O restante deste trabalho está organizado da seguinte forma. Na Seção 2, introduzimos o conceito de VIM como um serviço e revisamos o modelo de fatiamento de redes e VIMs sob demanda introduzido em [Stuart Clayman 2017]. A Seção 3 descreve os principais elementos do sistema de alocação de recursos e experimentação do FIBRE que foram usados neste trabalho. Na Seção 4 apresentamos a avaliação experimental que conduzimos a fim de validar a ideia de VIMs instanciados por demanda. Finalmente, a Seção 5 apresenta as conclusões e direções para trabalhos futuros.

## 2. VIM como um Serviço

Do ponto de vista de provedores de infraestrutura de computação em nuvem e de provedores de redes, automatizar o gerenciamento dos recursos oferecidos é crucial, uma vez que essa atividade é responsável por grande parte dos custos envolvidos. Gerenciadores de Infraestruturas Virtualizadas (VIMs) são plataformas de gerenciamento desenvolvidas com esse propósito. Dentre as principais funções de um VIM podemos destacar:

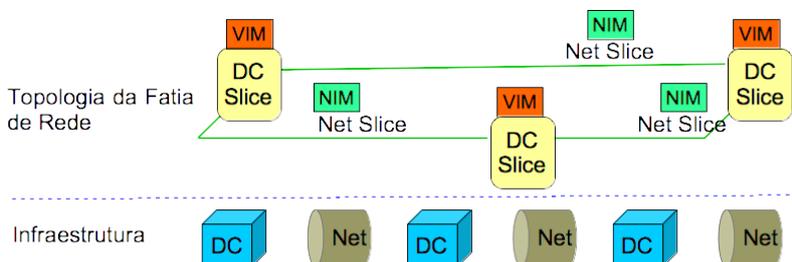
- fornecimento de uma visão homogênea dos recursos virtualizados, independente da tecnologia de virtualização subjacente;
- gerenciamento do ciclo de vida dos recursos virtualizados (máquina virtual ou

- contêiner), incluindo a configuração dinâmica de uma rede de comunicação para grupos de recursos, bem como seus requisitos de armazenamento;
- possibilidade de configurar diferentes políticas de alocação de recursos para atender as necessidades dos diferentes clientes;
  - possibilidade de aumentar ou diminuir dinamicamente a quantidade de recursos virtualizados para acompanhar a demanda da carga de trabalho dos clientes;
  - possibilidade de migrar serviços ou aplicações na presença de uma falha.

OpenStack e Kubernetes são exemplos de VIMs com grandes comunidades de usuários atualmente. O primeiro consiste em um conjunto de ferramentas usado para gerenciamento de máquinas virtuais, enquanto o segundo é uma plataforma para gerenciamento e orquestração de contêineres.

Em [Stuart Clayman 2017] é apresentada uma visão geral dos mecanismos, componentes e abstrações que podem ser utilizados para fornecer um VIM como um serviço no contexto de fatiamento de redes. Nessa visão, um provedor de rede possui e opera a infraestrutura física, que inclui recursos de computação, comunicação e armazenamento. Provedores de serviço, também conhecidos como inquilinos, pagam para o provedor de rede para fornecer seus serviços sobre a infraestrutura física, a qual é compartilhada entre essas entidades. Como cada inquilino tem seus próprios requisitos de serviço, um subconjunto dos recursos físicos é disponibilizado para ele como uma fatia de rede, e cada fatia é isolada das demais.

Uma fatia de rede é uma coleção de recursos físicos espalhados por um ou mais *data centers* (incluindo *data centers* de nuvem e de borda). Esses recursos são conectados dinamicamente em tempo de execução para criar um sistema em rede fim-a-fim, como mostrado na Figura 1. Dentro de um *data center*, o conjunto de recursos alocados a uma fatia é chamado de *DC slice* e geralmente abrange recursos de computação e armazenamento. No entanto, diferente de outras abordagens desenvolvidas para federação de nuvens, em [Stuart Clayman 2017], cada *DC slice* tem seu próprio VIM que é implantado sob demanda como um serviço. Como consequência, a escolha do VIM não é um recurso pré-determinado uma vez pelo provedor de rede, mas uma opção para o provedor de serviço. Isto também significa que o provedor de serviço tem todo o controle sobre seus recursos e seu VIM.



**Figura 1. Componentes de uma fatia de rede [Stuart Clayman 2017].**

Nesse modelo, uma abstração semelhante também é definida para recursos de rede, conhecida como *Network slice*. Um *Network slice* é criado sob demanda para conectar dois *DC slices*. Além disso, um *Network slice* é gerenciado e configurado pelo seu próprio *Network Infrastructure Manager* (NIM), o qual é implantado sob demanda quando o *Network slice* é instanciado.

Usando por exemplo um portal, um provedor de serviço pode solicitar uma fatia de rede com determinadas propriedades e detalhes de nível de serviço. Essa solicitação é enviada ao sistema de alocação de recursos do provedor de rede que mapeia a especificação da fatia para recursos de computação, armazenamento e rede e decide quais partes da infraestrutura devem alocá-las. Em seguida, cada *data center* é contactado para que criem o respectivo *DC slice* em seu domínio. Cada *data center* aloca os recursos necessários, instancia o VIM sob demanda para gerenciar os recursos daquela parte da fatia e retorna ao sistema de alocação os detalhes necessários para acessar o VIM. O sistema de alocação então cria os *Network slices* necessários para conectar os recursos dos diferentes *data centers* e instancia os respectivos NIMs que irão gerenciar os *Network slices*. Por fim, o provedor de serviço recebe suas fatias de recursos e seus VIM e NIMs necessários para gerenciá-las. Quando o tempo de vida da fatia de rede expira, seus recursos são liberados.

Em resumo, em [Stuart Clayman 2017], uma fatia de rede é uma coleção de *DC slices* conectados por *Network slices*, cada um com seus próprios VIMs e NIMs instanciados dinamicamente e sob demanda. No nosso trabalho, entretanto, consideraremos que as fatias de rede são criadas sobre recursos de um único *data center*, uma vez que esse cenário é suficiente para validar a ideia de VIM como um serviço. A prova de conceito de NIM como um serviço, bem como cenários mais complexos envolvendo vários VIMs e NIMs serão abordados em trabalhos futuros.

### 3. Sistema de Alocação de Recursos e Experimentação do FIBRE

A *testbed* FIBRE é uma infraestrutura de pesquisa voltada para a experimentação em redes, operada por instituições acadêmicas brasileiras e liderada pela Rede Nacional de Ensino e Pesquisa (RNP) [RNP 2018]. Atualmente, a *testbed* consiste em uma federação de 14 ilhas de experimentação, cada uma contendo um núcleo comum de switches Open-Flow, servidores para computação e armazenamento e um *cluster* de nós sem fio. Essa infraestrutura é gerenciada por um sistema de alocação de recursos e experimentação fortemente baseado no *framework cOntrol and Management Framework* versão 6.0, ou OMF6 [Stavropoulos et al. 2015]. A Figura 2 ilustra os componentes arquiteturais principais de uma ilha do FIBRE.

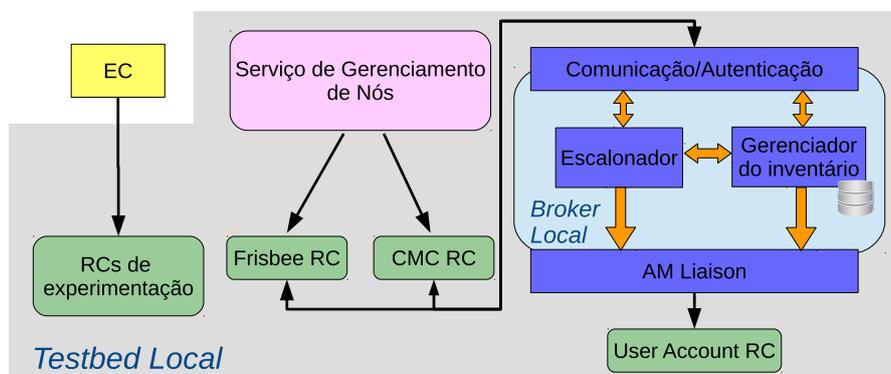


Figura 2. *Testbed* local de uma ilha FIBRE.

O ponto de entrada para o experimentador para todo o ambiente de experimentação é o *Experiment Controller* (EC). Essa entidade faz parte dos componentes

de controle que executam na camada de gerência, sendo responsável por executar os comandos que instruem a execução do experimento. Esses comandos são passados através de scripts escritos na linguagem OEDL (*OMF Experiment Description Language*). Como o EC faz parte da camada de gerência, na Figura 2 ele é apresentado de forma isolada dos componentes locais da *testbed*.

A entidade central na arquitetura do OMF6 é o recurso e diferentes elementos da *testbed* gerenciada são modelados através dessa entidade. Para cada tipo de recurso, existe um *Resource Controller* (RC) correspondente, ou seja uma entidade que age como um mediador entre o EC e os recurso local, gerenciando o ciclo de vida do recurso e enviando mensagens de controle para ele. Atualmente, existem RCs para representar máquinas virtuais, nós habilitados para conexão sem fio, sensores especializados, recursos OpenFlow, contas de usuários e geradores de imagens (como por exemplo o Frisbee).

O *Broker* é o módulo responsável por gerenciar o inventário, através do Gerenciador de Inventário, e a reserva de recursos, através do Escalonador. Além disso, ele ainda fornece uma interface, o *AM Liaison*, para a implementação de ações a eventos do Gerenciador de Invetário, como, por exemplo, a implementação do provisionamento automático de recursos ao atingir o horário de uma reserva. Um outro exemplo de uso do *AM Liaison* é mostrado na Figura 2, no qual o *Broker* se comunica com o *User Account RC* após a criação de uma conta no inventário, para criar uma conta Linux e configurar uma chave para acesso SSH.

O Serviço de Gerenciamento de Nós possibilita controlar os nós sem fio através do CMC RC, permitindo ligar, desligar, reiniciar e verificar o estado dos nós. Além disso, o Serviço de Gerenciamento de Nós permite também salvar imagens de disco dos nós e carregá-las posteriormente nos nós, através do Frisbee RC, de maneira eficiente, como mostraremos na Seção 4.

Além do EC, outros componentes da arquitetura executam na camada de gerência. Grande parte desses componentes são necessários em experimentos federados. Contudo, como o escopo deste trabalho está limitado a um único *data center*, representado em nossos experimentos por uma *testbed* local, não comentaremos sobre esses componentes. Uma descrição completa dos componentes de gerência e dos componentes locais de uma *testbed* FIBRE pode ser encontrada em [Ciuffo et al. 2017].

## 4. Avaliação Experimental

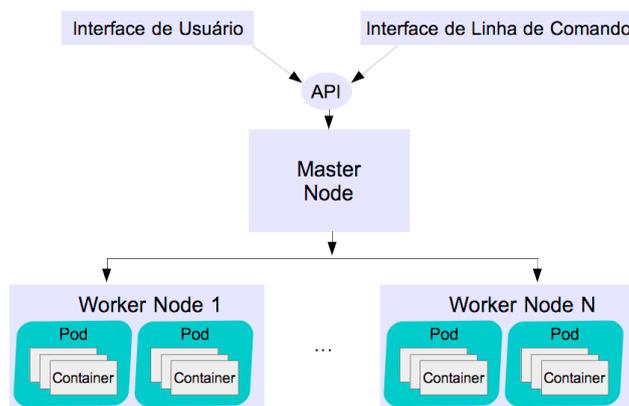
Nesta seção inicialmente apresentamos os VIMs utilizados em nossa avaliação experimental. Estes VIMs foram escolhidos por apresentarem características bastante diferentes, como discutiremos na Seção 4.1. Na Seção 4.2, descrevemos como a *testbed* FIBRE foi configurada para a realização dos experimentos, bem como os resultados obtidos.

### 4.1. VIMs usados na Avaliação

#### Kubernetes

O Kubernetes [Kubernetes 2018] é uma plataforma que permite a orquestração de contêineres espalhados em múltiplas máquinas de um *cluster*. Desenvolvido pela Google, tomando como base sua experiência de vários anos trabalhando com contêineres, o Kubernetes se tornou *open source* em 2014. Como mostrado na Figura 3, o Kubernetes possui dois componentes principais: *Master Node* e *Worker Node*. O *Master Node*

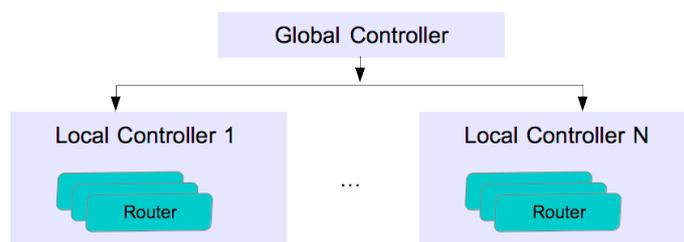
é responsável pelo gerenciamento do *cluster*. Todas as funcionalidades administrativas são executadas através dele, como por exemplo, tratamento de requisições *REST*, monitoramento do *cluster*, agendamento de tarefas e implantação de contêineres. Os *Worker Nodes*, por sua vez, são os componentes responsáveis pela execução dos contêineres.



**Figura 3. Arquitetura básica do Kubernetes.**

## VLSP

O VLSP [UCL 2018] é uma plataforma desenvolvida pela *University College London* com o propósito de simular vários aspectos no domínio SDN e ambientes virtuais dinâmicos. A plataforma é formada por três componentes principais: *Global Controller*, *Local Controller* e *Virtual Router*, como mostrado na Figura 4. O *Global Controller* atua como ponto de controle da plataforma, gerenciando todos os *Local Controllers*. Os *Local Controllers* agem como um hipervisor em uma máquina física, instanciando *Virtual Routers*. Cada *Virtual Router*, por sua vez, age como um recurso virtualizado, onde programas e serviços podem ser instanciados. Adicionalmente, *Virtual Routers* podem ser conectados entre si, formando uma topologia de rede. Contudo, diferentemente do Kubernetes e do OpenStack, o VLSP não é um VIM real, mas sim, uma plataforma de simulação de vários aspectos que envolvem VIMs reais em ambientes SDN.



**Figura 4. Arquitetura básica do VLSP.**

## Openstack

O OpenStack [OpenStack 2018] é uma das plataformas de gerenciamento de infraestruturas virtualizadas mais maduras atualmente. Lançada como uma plataforma *open source* em 2010 pela NASA e Rackspace, o OpenStack atualmente é constituído por vários componentes de software altamente modulares que permitem várias configurações diferentes

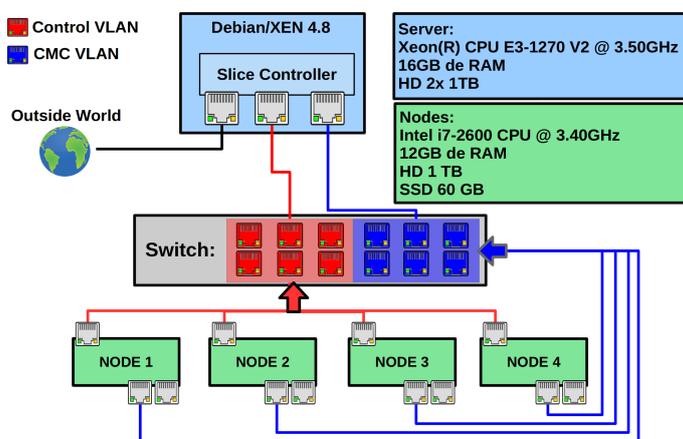
da plataforma. Os principais módulos são listados a seguir: 1) *Identity Service*, responsável pela autenticação dos usuários e de outros serviços que acessam a plataforma; 2) *Image Service*, armazena e recupera imagem das máquinas virtuais; 3) *Compute Service*, responsável pelo ciclo de vida das instâncias das máquinas virtuais; e 4) *Networking Service*, responsável pela configuração de rede dentro do Openstack. Além desses serviços principais, a plataforma oferece também um sofisticado serviço de orquestração de máquinas virtuais, monitoramento da infraestrutura e gerência de alocação de recursos *bare metal*.

Uma implantação básica do OpenStack envolve três módulos independentes: um módulo *Compute Service* para criar as instâncias virtualizadas, um módulo *Networking Service* para prover um serviço de rede para as instâncias e um serviço de mensagens que permite a comunicação entre os outros dois módulos.

De maneira resumida, os três VIMs usados nesta avaliação apresentam características muito distintas. Enquanto o OpenStack está centrado na gerência de ambientes baseados em tecnologias de virtualização de hardware, como hipervisores tradicionais, o Kubernetes e o VLSP são focados em gerência de ambientes baseados em virtualização leve. Adicionalmente, enquanto o OpenStack é uma plataforma madura, oferecendo vários serviços de gerência sofisticados, o Kubernetes é focado em *data centers* menores, e o VLSP é uma plataforma voltada para simulação em larga escala. Essa diversidade, por sua vez, impõe requisitos diferentes ao sistema de alocação e experimentação do FIBRE, como veremos a seguir.

## 4.2. Resultados

Para avaliar a viabilidade de se implantar VIMs sob demanda, utilizamos recursos da *testbed* FIBRE conforme configuração ilustrada na Figura 5. A infraestrutura de testes é composta de um servidor (em azul) e 4 nós (em verde). Esses recursos pertencem à ilha localizada na Universidade Federal de Goiás (UFG). No servidor, são armazenadas as imagens de cada um dos VIMs utilizados nesta avaliação. Essas imagens foram geradas a partir do Serviço de Gerenciamento de Nós do sistema de alocação de recursos da *testbed*. Nos nós, os VIMs serão implantados dinamicamente através de scripts OEDL.



**Figura 5. Testbed FIBRE configurada para nossos experimentos.**

Todos os nós possuem a mesma configuração de hardware, o que permite que imagens geradas em um nó sejam compatíveis com os demais. Cada nó possui um arduino,

chamado *Chassi Management Controller* (CMC), ligado à rede CMC. O CMC é utilizado para ligar e desligar os nós, além de permitir a verificação do estado (ligado/desligado) do nó. A outra interface do nó é ligada à rede de controle, a partir da qual é feita a instalação das imagens dos VIMs e a comunicação com eles e entre eles.

Para os experimentos, criamos imagens pré-configuradas para o VLSP, Kubernetes e Openstack. Para o VLSP criamos uma única imagem com todos os componentes instalados no Debian 9, resultando em uma imagem de 1,8 GB. Para o Kubernetes, criamos uma imagem com o *Master Node* e outra com o *Worker Node*, com tamanhos de 2,1 GB e 1,8 GB, respectivamente. Em ambas as imagens os componentes foram instalados no Ubuntu Server 16.04. Para o OpenStack, assumimos uma configuração básica e criamos uma imagem para o *Controller Node* (*Networking Service* juntamente com o serviço de mensagem) com 2,6 GB e outra para o *Compute Node* com 2,6 GB. Em ambas as imagens, os componentes foram instalados no CentOS 7 Minimal.

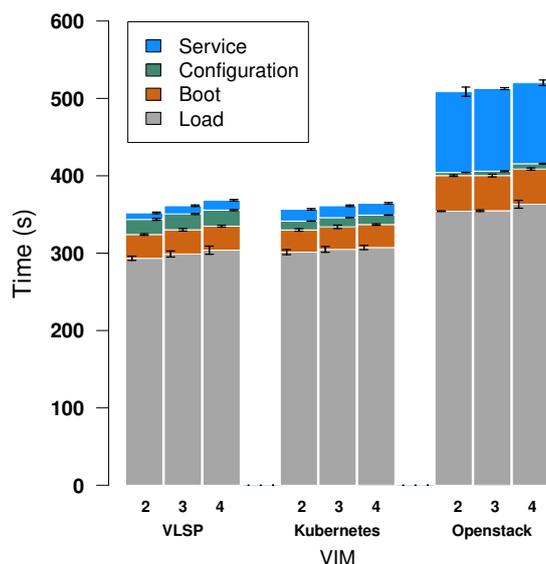
A fim de verificar o tempo necessário para instanciar uma infraestrutura com um VIM operacional, realizamos testes com cada um dos VIMs citados. Coletamos, então, quatro tempos: tempo de carga (*load*), necessário para carregar a imagem do VIM nos nós; tempo de inicialização (*boot*), necessário para iniciar o sistema operacional após a imagem ter sido aplicada; tempo de configuração (*configuration*), no qual as configurações necessárias para o VIM iniciar são realizadas; e o tempo de inicialização do serviço (*service*), o qual representa o tempo para o VIM ser colocado em execução após a configuração.

Para cada VIM, executamos 15 testes com um número variado de nós, sendo o mínimo com 2 nós e o máximo com 4 nós. A Figura 6, mostra os tempos médios, com o intervalo de confiança de 95%, coletados em nossos testes. Como ilustrado na Figura 6, o tempo para carregar as imagens (*load*) é o maior, e depende do tamanho da imagem carregada. Contudo, podemos observar que o aumento do tempo em relação ao aumento da quantidade de nós é relativamente pequeno, indicando a escalabilidade do sistema de alocação de recursos e experimentação do FIBRE. O tempo de *boot* também está relacionado à imagem carregada, e é semelhante independente da quantidade de nós utilizada nos testes.

O processo de configuração dos nós com VLSP e Kubernetes considera os seguintes passos em cada um dos nós: configurar a rede e reiniciar o serviço de rede. Em nossos testes, o tempo para reiniciar o serviço de rede foi maior no Debian 9 que no Ubuntu Server 16.04 e, por esse motivo, o tempo de configuração do VLSP foi maior que o do Kubernetes. Isso pode ser observado na Figura 6.

A configuração do OpenStack é mais complexa e envolve as seguintes etapas: configuração de IP estático e hostname dos nós; alteração dos arquivos de configuração com os novos IPs; atualização da lista de Compute Nodes e; alteração da senha do usuário admin do OpenStack. O tempo para que o OpenStack fique completamente operacional a partir de sua inicialização é maior que o tempo dos demais VIMs. Este resultado era esperado pois o OpenStack é composto de vários módulos diferentes, o que faz com que o serviço completo demore mais para iniciar. O VLSP é o VIM mais simples e, por este motivo, seu tempo de inicialização é o menor.

Por fim, realizamos testes carregando dois VIMs diferentes ao mesmo tempo para



**Figura 6. Tempos discriminados por VIM e por número de nós usados.**

mostrar a viabilidade de criar várias fatias de rede simultâneas, cada uma instanciando dinamicamente seu próprio VIM. Cada um dos VIMs foi carregado em dois nós (por exemplo, VLSP nos nós 1 e 2 e Kubernetes nos nós 3 e 4). Na Figura 7, mostramos os tempos de *load*, *boot*, *configuration* e *service* de cada conjunto de testes, discriminados por VIM. Como podemos observar, os tempos necessários para implantar os VIMs em conjunto é semelhante ao tempo necessário para implantar cada VIM separadamente. Isso mostra que, usando mecanismos de gerenciamento e implantação de imagens eficientes, a criação de uma fatia não influencia de maneira significativa a criação de outra. Isso evidencia também a eficiência e efetividade do sistema de alocação de recursos e experimentação do FIBRE.

## 5. Conclusão

Este artigo apresentou uma avaliação da viabilidade de se implantar VIMs sob demanda no contexto de fatiamento de redes, onde mais de um VIM pode ser instanciado simultaneamente. Para realizar essa validação utilizamos o sistema de alocação de recursos e experimentação do FIBRE. Os resultados mostraram que mesmo um VIM complexo como o OpenStack pode ser implantado em menos de 10 minutos em um conjunto de nós. Dado o tempo necessário para implantar um OpenStack completamente funcional, esse resultado é altamente satisfatório. Mostramos assim, que a ideia de VIM como um serviço no contexto de fatiamento de recursos é viável. Os resultados também mostram que o sistema de alocação de recursos e experimentação do FIBRE se comportou de maneira eficiente e escalável. Como trabalhos futuros, pretendemos validar a ideia de NIM como um serviço bem como cenários mais complexos envolvendo NIMs e VIMs. Estes experimentos serão interessantes também do ponto de vista do desenvolvimento e evolução do sistema de alocação de recursos do FIBRE, uma vez que diferentes RCs serão necessários nesta validação e exigirão mais do sistema.

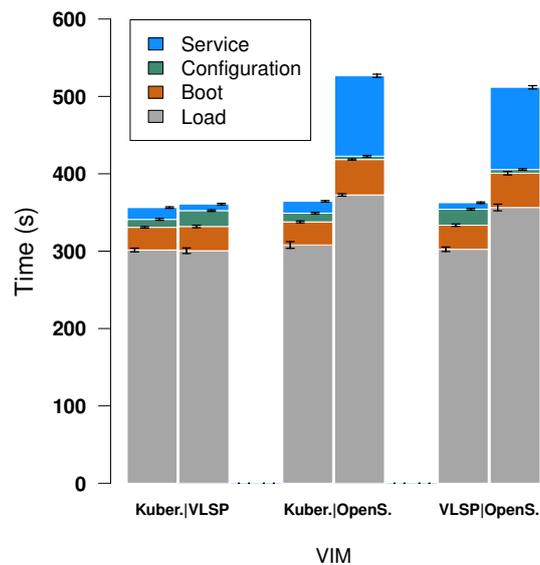


Figura 7. Tempos discriminados por conjuntos de VIMs.

## Referências

Ciuffo, L., Caled, D., Santos, J., de Rezende, J., and Machado, I. (2017). The next generation of the fibre software architecture. In *VII Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF)*.

Contreras, L. M. and López, D. R. (2018). A network service provider perspective on network slicing. *IEEE Softwarization*.

Kubernetes (2018). Production-Grade Container Orchestration. <https://kubernetes.io/>. Acessado em 15/03/2018.

OpenStack (2018). Open source software for creating private and public clouds. <https://www.openstack.org>. Acessado em 15/03/2018.

RNP (2018). Future Internet Brazilian Environment for Experimentation. <https://fibre.org.br/>.

Stavropoulos, D., Dadoukis, A., Rakotoarivelo, T., Ott, M., Korakis, T., and Tassiulas, L. (2015). Design, architecture and implementation of a resource discovery, reservation and provisioning framework for testbeds. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*, pages 48–53. IEEE.

Stuart Clayman (2017). Network Slicing Supported by Dynamic VIM Instantiation. <https://datatracker.ietf.org/meeting/100/materials/slides-100-nfvrg-3-network-slicing-support-by-dynamic-vim-instantiation>. Acessado em 15/03/2018.

Third Generation Partnership Project (3GPP) (2017). System Architecture for the 5G System. [http://www.3gpp.org/ftp//Specs/archive/23\\_series/23.501/](http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/). Acessado em 01/03/2018.

UCL (2018). Very Lightweight Network and Service Platform for SDN Environments. <http://clayfour.ee.ucl.ac.uk/usr/>. Acessado em 15/03/2018.