

## Proposta e Implementação de um *Framework* de Controle para *Testbeds* Federados que Integram Nuvem e SDN

Isabella de A. Ceravolo<sup>1</sup>, Diego G. Cardoso<sup>1</sup>, Cristina K. Dominicini<sup>1</sup>  
Rodolfo da S. Villaça<sup>1</sup>, Moisés R. N. Ribeiro<sup>2</sup>, Magnos Martinello<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Informática (PPGI)

<sup>2</sup>Programa de Pós-Graduação em Engenharia Elétrica (PPGEE)

Universidade Federal do Espírito Santo (UFES) – Vitória/ES

isabella.ceravolo@aluno.ufes.br, dgiacomellic@gmail.com

crisrina.dominicini@ifes.edu.br, rodolfo.villaca@ufes.br

moises@ele.ufes.br, magnos@inf.ufes.br

**Abstract.** *With the growth of Future Internet research, the demand for platforms for experimentation has grown. The FUTEBOL project aims to support research, education and innovation through the development of testbeds that allow an integrated experience between cloud computing and packet-switched, optical and wireless networks. To this end, this paper proposes a control framework architecture for testbeds that incorporates programmability and flexibility to standard technologies, such as OpenStack and OpenFlow, reducing the gap between proof of concept and production environments.*

**Resumo.** *Com o crescimento das pesquisas em Internet do Futuro cresceu a demanda por plataformas para experimentação. O projeto FUTEBOL objetiva apoiar pesquisa, ensino e inovação através do desenvolvimento de testbeds que possibilitam a experimentação integrada entre computação em nuvem e redes comutadas por pacote, redes ópticas e sem fio. Para esse fim, este trabalho propõe uma arquitetura de framework de controle para testbeds que incorpora programabilidade e flexibilidade à tecnologias e padrões adotados comercialmente, tais como OpenStack e OpenFlow, reduzindo a lacuna entre provas de conceito e os ambientes de produção.*

### 1. Introdução

Para contribuir com o avanço da pesquisa e inovação em redes e telecomunicações, pesquisadores e empresas desenvolvem novos protocolos, arquiteturas e aplicações. Uma etapa importante nesse processo é a realização de provas de conceito, o que necessita de um ambiente onde se possa reproduzir com fidelidade as condições desejadas de latência, largura de banda e escala. Os *testbeds* para experimentação de projetos de Internet do Futuro visam suprir essa necessidade por meio de plataformas que oferecem programabilidade e flexibilidade que, geralmente, não são encontradas em ambientes comerciais de computação em nuvem ou na Internet [Berman et al. 2015].

Com a crescente demanda por processamento e rede que deve ser atendida pela nuvem, pesquisadores e indústria têm se voltado para o desenvolvimento de novas soluções relacionadas à orquestração, virtualização de funções de rede e convergência entre redes móveis e ópticas. Para possibilitar a exploração dessas novas soluções, é necessário o desenvolvimento de tecnologias para apoiar a condução de experimentos em provas de conceito.

Nesse contexto surge o projeto FUTEBOL<sup>1</sup>, que visa fomentar a pesquisa, ensino e inovação em redes e telecomunicações, habilitando a experimentação integrada em redes ópticas e sem fio. Dentre os objetivos do FUTEBOL estão a implantação de *testbeds* e o desenvolvimento de um *Control and Management Framework* (CMF) federado para controle dos *testbeds*. A federação do projeto adota uma arquitetura heterogênea, em que cada *testbed* tem liberdade para selecionar seu CMF.

Neste trabalho, apresentamos uma proposta de arquitetura para o CMF que controlará recursos que serão oferecidos pelo *testbed* FUTEBOL na Universidade Federal do Espírito Santo (UFES). A proposta visa preencher a lacuna existente entre as plataformas de mercado e as plataformas de experimentação. De um lado, grande parte dos CMFs existentes não utilizam plataformas comerciais de nuvem<sup>2</sup>. Do outro lado, nuvens comerciais não oferecem o nível de programabilidade da infraestrutura que é desejado por experimentadores. Assim, o CMF proposto integra a flexibilidade exigida pela federação com a robustez de um ambiente similar ao encontrado em produção.

Trabalhos similares à proposta aqui apresentada já foram desenvolvidos pelo projetos OpenGENI *rack*<sup>3</sup> e KOREN<sup>4</sup>. Entretanto, a solução OpenGENI possui alto custo e baixa flexibilidade, dificultando sua adoção no contexto apresentado. O KOREN não disponibiliza documentação suficiente para a reprodução de seu CMF.

## 2. Análise de Requisitos

Como membro da federação FUTEBOL, o *testbed* na UFES precisa respeitar decisões tomadas em conjunto pelos participantes do projeto. Uma das decisões é que os *testbeds* do projeto sejam integrados à federação Fed4FIRE. Assim, o CMF deve ser compatível com a arquitetura *Slice-based Federation Architecture* (SFA). Como trata-se de uma federação em que cada *testbed* é operado por uma instituição diferente e tem autonomia para controlar os recursos, o SFA é essencial para promover a consistência e interoperabilidade na descoberta, reserva e provisionamento de recursos [Sitaram et al. 2016]. Outros requisitos advindos da integração com o Fed4FIRE são: i) Suporte ao OML como ferramenta de instrumentação de experimentos; ii) Dispor de ferramenta para monitoramento do *testbed*; e iii) Suporte a ferramenta de controle de experimento.

Além disso, o CMF aqui apresentado segue as premissas de software aberto, com a intenção de proporcionar uma solução que pode ser replicada com baixo custo. Por isso, evitamos o uso de soluções proprietárias para controle dos recursos físicos. No *testbed* da UFES, serão oferecidos recursos de computação e equipamentos de redes comutadas por pacote, ópticas e sem fio. Para controlar esses recursos, utilizaremos tecnologias abertas, porém padrão de mercado (OpenFlow, por exemplo), pois acreditamos que isso favorece a consistência dos resultados obtidos pelos experimentos com as condições encontradas em ambientes de produção.

## 3. Projeto e Implementação do CMF do FUTEBOL

Com base nos requisitos mencionados, a Figura 1 mostra a arquitetura proposta para o CMF desenvolvido na UFES. A *Camada de Federação* atua como mediadora entre as

<sup>1</sup><http://www.ict-futebol.org.br/>

<sup>2</sup>Por exemplo, o OMF (<http://mytestbed.net/>) e o OCF (<http://fp7-ofelia.github.io/ocf/>).

<sup>3</sup><http://www.opengeni.net/>

<sup>4</sup><http://www.koren.kr/koren/eng/>

requisições vindas da federação e o componente *Orquestrador*, que interage com os controladores dos recursos físicos. Essa camada acessa diretamente a base de dados que armazena informações sobre os recursos reservados. Essas informações, além de possibilitarem o acesso externo via SSH (*Secure SHell*) às máquinas virtuais, também são utilizadas pelo *Orquestrador*. Como o *testbed* oferecerá uma plataforma de computação em nuvem, com equipamentos de rede programáveis seguindo o conceito de *Software Defined Networks* (SDN), teremos dois *Aggregate Managers* (AMs): um para a nuvem e outro para recursos SDN. Para apoiar o desenvolvimento dos AMs, foi selecionada a implementação de referência das interfaces SFA desenvolvida pelo projeto GENI, o *GENI Control Framework* (GCF) [Kim and Lee 2012]. Dessa forma, cada AM, além de utilizar o GCF, também possui um componente, chamado de *Despachante*, responsável por traduzir as chamadas SFA em chamadas para a API *RESTful* do *Orquestrador* do *testbed*.

O *Orquestrador* do *testbed* tem a função de definir quais recursos devem ser empregados no atendimento das requisições, por meio de otimizações realizadas internamente. A *Camada de Orquestração* é fundamental, pois permite a manipulação integrada e eficiente de recursos e promove o desacoplamento entre os mecanismos que habilitam a federação e as tecnologias que gerenciam os recursos.

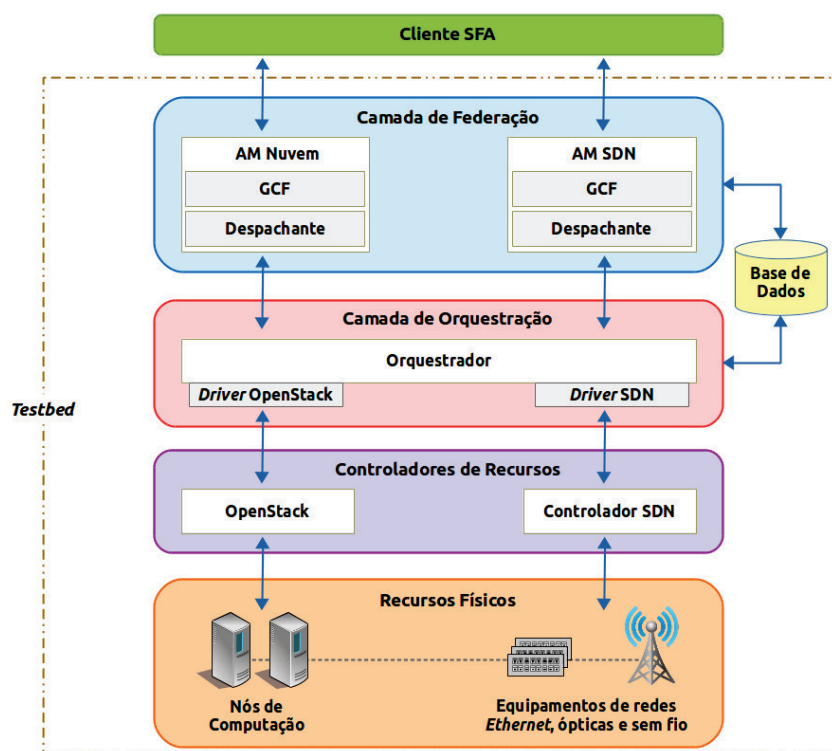


Figura 1. Arquitetura do *framework* de controle do *testbed*.

A camada de *Controladores de Recursos* contém as plataformas utilizadas na gestão dos recursos físicos. Para gerir os recursos de computação em nuvem, selecionamos o OpenStack, definido como um sistema operacional de nuvem [Sitaram et al. 2016]. Ele foi escolhido por seguir o modelo de infraestrutura como serviço e, assim, permitir o controle integrado dos recursos físicos de computação, armazenamento e rede. Trata-se de um projeto aberto, desenvolvido por uma comunidade global e que conta com o apoio de organizações consolidadas no ramo de computação em nuvem. Essas caracte-

terísticas colaboram para que seja uma plataforma em constante evolução que agregue as tendências do mercado. Além disso, é possível realizar instalações customizadas, selecionando os módulos que oferecem os serviços desejados [del Castillo et al. 2013]. As aplicações podem gerir os serviços da nuvem através das APIs do OpenStack, o que proporciona flexibilidade. Como plataforma para o controle dos recursos de rede, selecionamos o controlador Ryu, que pode receber comandos via API *RESTful*. Na camada física, serão utilizados hardware *commodity*, graças à flexibilidade oferecida pelos controladores de recursos escolhidos. Para atender ao requisito de oferecer uma ferramenta para controle de experimento, selecionamos o NEPI por ser compatível com a linguagem Python (usada no desenvolvimento do CMF). Assim, qualquer usuário com conhecimento de lógica de programação e da linguagem Python poderá criar *scripts* para a execução de experimentos com os recursos do *testbed*. Para atender ao requisito de monitoramento do *testbed*, selecionamos o Zabbix, por sua robustez.

#### 4. Conclusão

A proposta apresentada diminui a lacuna entre o ambiente onde são feitas as provas de conceito e onde as soluções serão implantadas. A arquitetura proposta preza pela flexibilidade e baixo custo, adotando tecnologias abertas. O *Orquestrador* permite que o CMF suporte com eficiência experimentos que necessitam de migração e/ou escalada de recursos. O desenvolvimento do CMF do *testbed* FUTEBOL na UFES é um trabalho em andamento. Até a escrita desse trabalho, uma versão inicial do AM da nuvem e do *driver* para OpenStack já haviam sido desenvolvidos. Ambos estão disponíveis<sup>5</sup> e contam com instruções para instalação. Há uma demonstração em vídeo<sup>6</sup> desses componentes trabalhando na alocação de máquinas virtuais. O processo de definição dos requisitos do *testbed* se mostrou incremental e centrado nos aspectos de interoperabilidade, escalabilidade e código aberto.

#### Agradecimentos

Este trabalho tem recebido financiamento do projeto Horizon 2020 (União Européia) sob no. 688941 (FUTEBOL), assim como do MCTI por meio da RNP e do CTIC. Além disso, gostaríamos de agradecer o financiamento do CNPq/CAPES e FAPES.

#### Referências

- Berman, M. et al. (2015). Future internets escape the simulator. *Communications of the ACM*, 58(6):78–89.
- del Castillo, J. A. L., Mallichan, K., and Al-Hazmi, Y. (2013). Openstack federation in experimentation multi-cloud testbeds. In *Cloud Computing Technology and Science (CloudCom), 5th International Conference on*, volume 2, pages 51–56. IEEE.
- Kim, H. and Lee, S. (2012). First cloud aggregate manager development over first: Future internet testbed. In *Information Networking (ICOIN), 2012 International Conference on*, pages 539–544. IEEE.
- Sitaram, D., Harwalkar, S., and Kumar, K. S. (2016). Standards based integration of intercloud for federation with openstack. In *Cloud Computing in Emerging Markets (CCEM), 2016 IEEE International Conference on*, pages 113–118. IEEE.

<sup>5</sup><https://github.com/nerds-ufes/cm-futebol>

<sup>6</sup><https://goo.gl/D9OLn6>