

Avaliação de desempenho em Switches OpenFlow

Gustavo de Araújo¹, Cristiano Bonato Both²

¹Departamento de Informática – Curso de Engenharia de Computação
Universidade de Santa Cruz do Sul (UNISC)
Santa Cruz do Sul – RS – Brazil

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

inf.araujo@gmail.com, cbboth@inf.ufrgs.br

Abstract. *Software-Defined Networking (SDN) is a paradigm of computer networks that allows flexibility and agility in developing new technologies. This is achieved by separating the control of network devices that route the user packages. This paradigm has gained prominence with the specification of the OpenFlow protocol, which allows a communication interface between the control plane and the data plane. Despite the rapid spread of this protocol, the literature does not go deep performance analysis of data routers devices (switches). This article proposes a methodology for evaluating performance OpenFlow switches focused on analyzing the data plane hardware. As a result it is obtained a comparison of four OpenFlow switches, and two virtual-two reais.*

Resumo. *Software-Defined Networking (SDN) é um paradigma de redes de computadores que permite a flexibilidade e agilidade no desenvolvimento de novas tecnologias. Isto é alcançado, separando o controle da rede dos dispositivos que encaminham os pacotes do usuário. Este paradigma ganhou destaque com a especificação do protocolo OpenFlow, que permite uma interface de comunicação entre o plano de controle e o plano de dados. Apesar da rápida disseminação deste protocolo, a literatura não aprofunda na análise de desempenho dos dispositivos encaminhadores de dados (switches). Este artigo, propõem uma metodologia para a avaliação de desempenho em switches Openflow focada em analisar o hardware do dispositivo de encaminhamento. Como resultados é obtido uma comparação entre quatro switches OpenFlow, sendo, dois virtuais e dois reais.*

1. Introdução

Software-Defined Networking é um paradigma emergente em redes de computadores que propõe uma plataforma flexível para o desenvolvimento de novas soluções para infraestruturas de redes. Uma de suas principais características está na separação da lógica de controle da rede (controlador) dos equipamentos que encaminham os pacotes através de um plano de dados (*switches*). Com essa separação, os *switches* se tornam apenas equipamentos de encaminhamento de pacotes [Kreutz et al. 2014]. Para que essa separação ocorra, é necessário um protocolo de comunicação entre o plano de controle e o plano de dados. Com este propósito, o protocolo Openflow foi desenvolvido, sendo

atualmente, a mais relevante implementação do paradigma SDN [McKeown et al. 2008]. Um *switch* Openflow possui uma ou mais tabelas com regras para manipulação dos pacotes de dados (tabela de fluxos). Cada regra corresponde a um subconjunto do tráfego e uma sequência de ações de como lidar com os pacotes, como por exemplo, descartar, encaminhar ou modificar cabeçalho [ONF 2013].

Com esse novo paradigma, as métricas tradicionais de avaliação de desempenho como latência, *jitter*, perda de pacotes e largura de banda se tornam insuficientes para analisar o desempenho de equipamentos de redes baseadas em SDN. O desempenho de um *switch* OpenFlow não está ligado somente à vazão do tráfego de dados, mas também em como é feito o processamento do tráfego de controle. Por exemplo, é importante avaliar a latência da comunicação entre *switch* e controlador. Pois, a latência deste canal impacta diretamente no tráfego de dados, uma vez que, o *switch* não toma decisões autônomas, sendo de responsabilidade do controlador informar o que deve ser feito com determinado fluxo. Deste modo, uma metodologia de avaliação de desempenho deve ser elaborada para esse novo paradigma de rede.

Atualmente, a literatura sobre SDN apresenta poucas soluções para a avaliação de desempenho. Cbench [Tootoonchian et al. 2012] avalia a latência de atualização da tabela de fluxos o atraso de pacotes que necessitam de modificação do cabeçalho para serem encaminhados. OFCBenchmark [Jarschel et al. 2012] também realiza uma análise de controladores, verificando métricas como os pacotes por segundos enviados do controlador para o *switch*. E Oflops [Rotsos et al. 2012] permite avaliar algumas características do *switch* como tempo de instalação de uma nova regra na tabela de fluxos, latência do canal de controle e impacto das mensagens de coleta de estatística no tráfego de controle e tráfego de dados.

O restante deste trabalho é organizado da seguinte forma: na seção 2 são apresentadas as bases para a realização deste trabalho. Na Seção 3 é apresentado a metodologia para a validação de desempenho. No Seção 4, são exibidos os resultados obtidos com a implementação do protótipo. E finalmente, na Seção 5 são exibidas as considerações finais e trabalhos futuros.

2. Fundamentação Teórica

Esta Seção aborda os conceitos básicos sobre SDN e Openflow, assim como, alguns dos trabalhos relacionados a avaliação de desempenho neste paradigma. Na Seção 2.1 são apresentados os principais conceitos do paradigma SDN e do protocolo Openflow. E na Seção 2.2 são apresentados os trabalhos relacionados a avaliação de desempenho em SDN.

2.1. SDN/Openflow

Em uma rede IP tradicional, os planos de dados e controle são acoplados e embarcados no mesmo equipamento de rede, com controle sendo descentralizado, fazendo com que os equipamentos tomem decisões sobre o encaminhamento de pacotes de maneira autônoma. Este tipo de abordagem torna a arquitetura de uma rede muito complexa. Como consequência, tem-se uma rede muito estática que dificulta o desenvolvimento de novos serviços. Visando permitir que soluções de rede sejam facilmente projetadas e desenvolvidas o paradigma SDN foi baseado nos seguintes pilares:

- Desacoplar plano de dados do plano de controle. Desta maneira, os equipamentos de rede se tornam apenas encaminhadores de pacotes, não tomando decisões autônomas.
- Lógica de controle é movida para uma entidade externa, centralizada, implementada em software. Esta entidade chama-se controlador e tem a funcionalidade de implementar toda a lógica de controle da rede, isto faz com que, o desenvolvimento de novos serviços de rede sejam facilmente implementáveis.
- O encaminhamento de dados torna-se baseado em fluxos ao invés de ser baseado em destino. Um fluxo pode ser definido como um conjunto de campos de cabeçalhos utilizados para identificação (*match*) e um conjunto de ações (*action*) [Kreutz et al. 2014].

Na arquitetura SDN, o plano de dados é representado pelos equipamentos de rede como *switches* e roteadores, porém, nenhuma decisão de controle é tomada de maneira embarcada nestes equipamentos. Sendo assim, o plano de dados apenas implementa as regras definidas pelo plano de controle [ONF 2013]. O plano de controle, no paradigma SDN, é representado pelo controlador que é responsável pela inteligência da rede. Através de uma interface de comunicação, o plano de controle “programa” os equipamentos do plano de dados, de acordo com a necessidade da rede. Desta maneira os controladores são responsáveis por decidir como serão encaminhados os pacotes do plano de dados.

O paradigma SDN foi amplamente difundido, após a especificação do protocolo OpenFlow. Este protocolo define uma interface entre o plano de dados e o plano de controle. O protocolo OpenFlow é implementado em ambos os planos, de forma que, o equipamento do plano de dados possui uma tabela que armazena regras de encaminhamento de dados, chamada tabela de fluxos. Este protocolo também prevê um canal seguro para comunicação entre *switch* e controlador, como por exemplo *Security Socket Layer*, assim como um conjunto de mensagens para manipulação da tabela de fluxos.

2.2. Trabalhos Relacionados

As pesquisas referentes a análise de desempenho em SDN, em sua maioria, focam-se em analisar controladores. Por exemplo, na metodologia apresentada pelo *benchmark Cbench* [Tootoonchian et al. 2012], avalia a latência de processamento do controlador para requisições do tipo *Packet-in*, bem como quantidade máxima de mensagens *Packet-in* que o controlador consegue suportar por segundo.

O *framework* Oflops possui uma grande quantidade de ferramentas para avaliação de desempenho em *switches* OpenFlow [Rotsos et al. 2012]. Com esta ferramenta é possível verificar a capacidade dos *switches* de encaminhar os pacotes de dados, o tráfego do canal de controle, latência para instalação de um novo fluxo, além de monitorar o tráfego. Essa ferramenta também possibilita a geração de tráfego de dados e tráfego de controle. Oflops possui uma arquitetura modular, permitindo que novos testes sejam adicionados ao *framework* de maneira rápida [Rotsos et al. 2012].

É possível perceber que apenas a metodologia Oflops fornece testes para a avaliação de desempenho de *switches* OpenFlow, enquanto, Cbench e OFBenchmark focam-se em testes para controladores. E mesmo assim, o *framework* Oflops oferece poucas ferramentas para avaliação do hardware dos *switches*.

3. Metodologia de Avaliação

Esta Seção descreve a metodologia utilizada para avaliação de desempenho, os tipos de testes realizados e o cenário onde os testes foram feitos. Na Seção 3.1 são descritos detalhes da implementação dos testes. Na Seção 3.2 são apresentados os tipos de testes utilizados. E na Seção 3.3 é apresentado o cenário de teste utilizado para obter os resultados.

3.1. Implementação

Para alcançar esses objetivos foi escolhido o controlador Floodlight [Network 2014]. É um controlador muito difundido dentro da comunidade SDN possuindo uma boa documentação sendo de fácil utilização. Além disso, este controlador possui uma API JSON permitindo uma fácil interação de maneira remota. Neste trabalho, esta API foi utilizada para a passagem de parâmetros, inicialização e coleta de resultados.

Para gerar novos fluxos é utilizado um algoritmo que incrementa o MAC destino e o MAC origem. Este mecanismo é utilizado, pois, o endereço MAC possui 48 bits de endereçamento, isto permite uma grande combinação de fluxos diferentes sejam instalados como por exemplo, um fluxo com MAC origem: 00:00:11:11:22:22 e vlan: 100. A *action* utilizada é o direcionamento do tráfego para uma determinada porta.

3.2. Tipos de Teste

Foram definidos seis tipos de *match* para a realização dos experimentos, Podem ser visualizados na Tabela 1.

Tabela 1. Tipos de Match utilizados para teste.

Tipo de Match	Campos de Match
1	MAC Destino
2	MAC Destino e MAC Origem
3	MAC Destino, MAC Origem, VLAN e VLAN Priority
4	L2, IP Origem e IP Destino (sem Type)
5	L2, IP Origem, IP Destino e Type
6	Todos os campos

3.3. Cenário de Testes

Para coleta e análise dos resultados, foram escolhidos quatro *switches*, onde os testes foram efetuados. Estes *switches* foram divididos em dois grupos: (i) *switches* virtuais e (ii) *switches* reais. O primeiro grupo, diz respeito a uma implementação em software que emula um *switch* real. Os *switches* utilizados para este grupo foram *OpenvSwitch* [OpenvSwitch 2015] e a implementação do *switch* de Stanford [Naous et al. 2008] utilizando NetFPGA [NetFPGA 2015]. A Tabela 2 apresenta a comparação entre os *switches* virtuais utilizados, assim como, algumas informações das máquinas hospedeiras de cada um deles.

<i>Características</i>	OpenvSwitch	Stanford
Capacidade da Tabela de Fluxos	100KB	32KB
CPU	Core Intel i5-3337U	Intel Core 2 Duo
Memória	6GB	3GB
OS	Ubuntu 12.04	CentOs 4
Interfaces	4 x 10/100BASE-T RJ45 (todas Virtuais)	4 x 1000BASE-T RJ45 (NetFPGA)
Capacidade de encaminhamento	Não especificado	1000M

Tabela 2. Comparativo entre *switches* virtuais

O segundo grupo, diz respeito a *switches* reais que são comercializados. A Tabela 3 mostra um comparativo entre os *switches* escolhidos, as informações foram retiradas da especificação disponibilizada pelo fabricante de cada *switch*. Para o *Switch A* foi utilizado o *firmware* Indigo [Indigo 2015] e para o *Switch B* foi utilizado o *firmware* original de fábrica.

<i>Feature</i>	Switch A	Switch B
Capacidade da Tabela de Fluxos	Não especificado pelo fabricante	2.5KB
CPU	P2020	Cavium CN5010
Memória	2GB	256MB
OS	Debian	Indigo
Interfaces	48 x 10/100/1000BASE-T RJ45 4 x 1 GbE (SFP)	48 x 100/1000BASE-T RJ45 4x 1 GbE (SFP)
Capacidade de encaminhamento	132GB	176GB

Tabela 3. Comparativo entre *switches* reais

4. Resultados

Esta Seção apresenta os resultados obtidos com os testes implementados. Na Seção 4.1 apresenta os resultados para a capacidade máxima da tabela de fluxos. Na Seção 4.2 apresenta os resultados para a utilização da memória. E finalmente, na Seção 4.3 são apresentados os resultados para a utilização de CPU.

4.1. Capacidade da Tabela de Fluxos

Com os testes realizados no *Switch A*, pode-se notar que a quantidade de campos utilizados para *match* influencia na capacidade total de armazenamento de regras da tabela de fluxos. Na Figura 1 são exibidos os resultados obtidos. Com a análise do gráfico percebe-se que existe uma relação entre a quantidade de fluxos que podem ser instalados e a quantidade de campos utilizados para *matches*. Quanto maior a quantidade de campos necessários para realizar um *match*, menor a quantidade máxima de fluxos que podem ser instalados. Percebe-se também que para *matches* utilizando cabeçalho L2 e IP origem e

destino (Tipo 4) há uma grande queda na quantidade de fluxos que podem ser instalados. Entretanto, ao utilizar o campo *type* do cabeçalho Ethernet a quantidade máxima de fluxos instalados aumenta. Isso acontece devido a algum mecanismo de otimização para o armazenamento de fluxos na tabela de acordo com os campos utilizados para *match*.

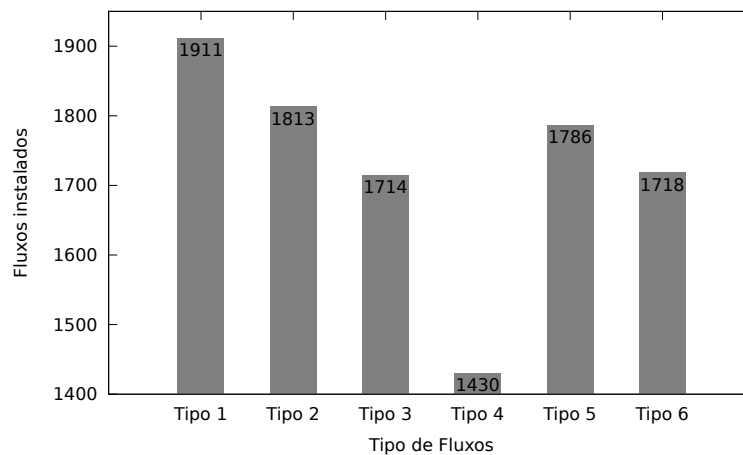


Figura 1. Resultados obtidos para o Switch A.

Os outros *switches* não apresentaram variações de acordo com o tipo de *match* realizado, além de, permitirem uma capacidade de armazenamento de regras igual ou maior do que a anunciada pelo fabricante. A tabela 4 são exibidos os resultados para o Switch B, OpenvSwitch e Stanford.

Switch	Capacidade Máxima da Tabela de Fluxos
Switch B	2560
Stanford	32792
OpenvSwitch	1000000

Tabela 4. Quantidade máxima de fluxos instalados em cada switch.

4.2. Utilização da Memória

Para cada tipo de *match* foi realizada a medição da quantidade de memória utilizada conforme os fluxos fossem instalados. A Figura 2 mostra a comparação da utilização da memória entre os Switch A e o Switch B. O Switch A variou a quantidade de memória utilizada de acordo com a quantidade de fluxos instalados. O Switch B mostrou uma diminuição da utilização de memória de acordo com a quantidade de campos utilizados para o *match*.

Os Switches virtuais mostraram um comportamento constante, a quantidade de campos utilizados para o *match* não variou a quantidade de memória utilizada. O OpenvSwitch utilizou 500MB de memória RAM para instalação de 100K regras, enquanto, o Switch de Stanford utilizou 16MB para a instalação de aproximadamente 32K regras.

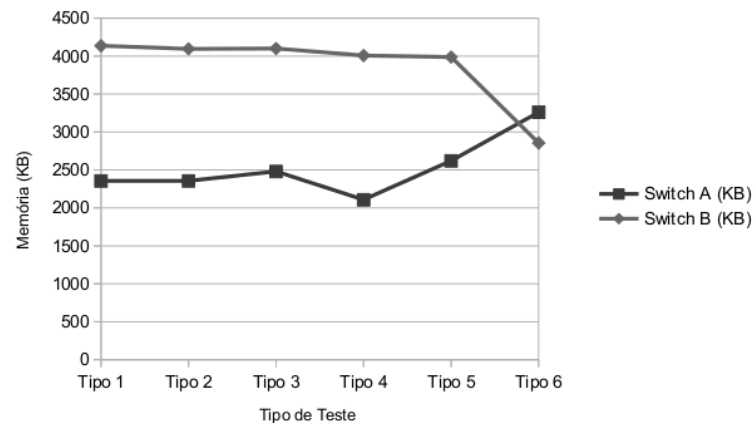


Figura 2. Resultados de utilização de memória entre os *switches* reais.

4.3. Utilização de CPU

Para realização deste teste foi enviado para cada *switch* a quantidade máxima de fluxos que podem ser instalados em suas tabela de fluxos. Como resultado, foi obtido a sobrecarga que a instalação seguida de vários fluxos possui em cada *switch*. A Tabela 5 apresenta os resultados obtidos. É possível notar que a instalação de seguida de várias fluxos nos *switches* reais não apresentam um sobrecarga significativa na CPU. Porém, nos *switches* virtuais a utilização da CPU apresenta um acréscimo significativo. Isso ocorre, pois os *switches* virtuais não possuem processadores otimizados para este tipo de funcionalidade. Além de que, os *switches* reais possuem um hardware específico para o processamento de mensagens OpenFlow.

	Switch A	Switch B	OpenvSwitch Memória RAM	Stanford Memória RAM
Sobrecarga de Utilização da CPU	2%	3%	27%	14%

Tabela 5. Sobrecarga da utilização da CPU no momento da instalação de fluxos

5. Conclusão e Trabalhos Futuros

Nesse artigo foi proposto uma metodologia de testes de desempenho para avaliar o hardware de *switches* Openflow. Para a implementação dessa metodologia, foi estudado as métricas que influenciam no desempenho destes dispositivos. Além disso, foi estudado como as atuais metodologias de avaliação de desempenho são implementadas. Após a implementação, a metodologia de testes foi aplicada em quatro *switches* OpenFlow, sendo dois *switches* reais e dois *switches* virtuais. Os resultados obtidos foram apresentados comparando os *switches* utilizados, onde, pode-se verificar qual *switch* possui o melhor desempenho para uma determinada métrica.

Este trabalho, abre a possibilidade para a realização de projetos como otimizações na metodologia desenvolvida, aumentar a quantidade de tipos testes de testes, criação de uma base de dados com os resultados obtidos por cada *switch*. Dessa forma, pode-se aumentar o escopo da metodologia de avaliação, possibilitando a fácil comparação dos resultado entre diferentes *switches*.

Referências

- Indigo (2015). Indigo. Disponível em: <<http://www.projectfloodlight.org/indigo/>>. Acesso em: Set. 2015.
- Jarschel, M., Lehrieder, F., Magyari, Z., and Pries, R. (2012). A flexible openflow-controller benchmark. In *Proceedings of the 2012 European Workshop on Software Defined Networking*, EWSDN '12, pages 48–53, Washington, DC, USA. IEEE Computer Society.
- Kreutz, D., Ramos, F. M. V., Veríssimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Computing Research Repository*, abs/1406.0440.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, pages 69–74.
- Naous, J., Erickson, D., Covington, G. A., Appenzeller, G., and McKeown, N. (2008). Implementing an openflow switch on the netfpga platform. In *Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '08, pages 1–9, New York, NY, USA. ACM.
- NetFPGA (2015). Netfpga. Disponível em: <<http://netfpga.org/site/>>. Acesso em: Ago. 2015.
- Network, B. S. (2014). Floodlight. Disponível em: <<https://projectfloodlight.org/floodlight/>>. Acesso em: Set. 2014.
- ONF (2013). Sdn architecture. Disponível em: <<https://www.opennetworking.org/>>. Acesso em: Mar. 2014.
- OpenvSwitch (2015). Openvswitch. Disponível em: <<http://openvswitch.org/>>. Acesso em: Ago. 2015.
- Rotsos, C., Sarrar, N., Uhlig, S., Sherwood, R., and Moore, A. W. (2012). Oflops: An open framework for openflow switch evaluation. In *Proceedings of the 13th International Conference on Passive and Active Measurement*, PAM'12, pages 85–95, Berlin, Heidelberg. Springer-Verlag.
- Tootoonchian, A., Gorbunov, S., Ganjali, Y., Casado, M., and Sherwood, R. (2012). On controller performance in software-defined networks. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, Hot-ICE'12, pages 10–10, Berkeley, CA, USA. USENIX Association.