

Análise de Utilização de Recursos Computacionais pelos Controladores SDN

Igor Morais¹, Marcelo Santos¹, Petrônio Junior¹, Carlos Kamienski², Stenio Fernandes¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Av. Prof. Luís Freire s/n, Cidade Universitária – 50740-540 – Recife – PE, Brasil

²Universidade Federal do ABC (UFABC)
Av. dos Estados 5001, Bangu – 09210-580 – Santo André – SP – Brasil

¹{imlm,mabs,pglj,sflf}@cin.ufpe.br

²cak@ufabc.edu.br

Abstract. *There are several studies of SDN controllers are overloaded with traffic in different test scenarios and their responses are computed in order analyze how such variations impact their throughput. Differently, this paper presents performance analyses of several SDN controllers looking at computing resources utilization on their host machines.*

Resumo. *Existem diversos estudos em que os controladores SDN são submetidos à sobrecarga de tráfego em diferentes cenários e suas respostas são analisadas em relação apenas à vazão da rede. Diferentemente, este artigo apresenta uma análise de desempenho de diversos controladores SDN observando o consumo de recursos das máquinas em que estão sendo executados.*

1. Introdução

O paradigma SDN (*Software Defined Networking*) tem se tornado mais difundido nos meios acadêmicos e na indústria [Nunes et al. 2014], tornando as redes programáveis. Isso difere o SDN dos paradigmas de redes anteriores, em que era preciso configurar os elementos de maneira isolada, demandando acesso a cada um dos equipamentos [Feldmann et al. 2002]. Com o SDN, a rede é configurada em um único ponto da rede. Para o seu adequado funcionamento, faz-se necessário o uso dos controladores SDN, que podem estar em qualquer ponto. Eles utilizam algumas linguagens para programar a rede como Python, C++ e Java [Monaco et al. 2013]. Além disso, por serem *softwares*, podem sofrer as consequências do *software aging* [Alencar et al. 2014]. Neste trabalho, foi realizado um estudo sobre o impacto do uso dos controladores no consumo de recursos na máquina hospedeira. Para isso, foram realizadas análises quantitativas e qualitativas desta relação. O restante deste trabalho é organizado como segue. A Seção 2 apresenta os trabalhos relacionados a esta pesquisa. A Seção 3 apresenta a metodologia que foi utilizada no decorrer do desenvolvimento deste artigo. Logo em seguida, na Seção 4 são apresentados os resultados obtidos. Por fim, na Seção 5 são resumidas as conclusões obtidas e os trabalhos futuros.

2. Trabalhos Relacionados

Nos últimos anos, diversos estudos foram realizados com foco no uso das redes SDN, principalmente relacionados ao desempenho dos controladores SDN. Em [Shalimov et al. 2013], os autores abordaram aspectos como desempenho, confiabilidade e

escalabilidade dos principais controladores SDN do mercado. Parâmetros como a quantidade de *threads* do controlador, de *switches* simulados e de MACs utilizados foram observados. Nos resultados, apenas a vazão dos dados na rede como indicador. Em [Wang et al.2014], foi desenvolvido um controlador SDN com características que o torna mais eficiente em relação a outros controladores, como: (i) a alocação estática da memória, (ii) técnicas de *multithread* e (iii) a forma de agregação de pacotes. Por fim, são comparados os resultados em relação à vazão e latência. Este trabalho é uma extensão da pesquisa realizada em [Alencar et al. 2014], onde o consumo de recursos de controladores implementados em Java foram analisados. Foram realizadas comparações dos controladores com menos recursos e apenas uma linguagem de programação.

3. Metodologia

Neste trabalho, o consumo dos recursos da máquina hospedeira no uso dos controladores SDN foi analisado. Os controladores utilizados nos testes foram: Floodlight¹ (Java), POX² (Python) e Ryu³ (Python). O cenário desenvolvido para os testes é descrito a seguir. Duas máquinas são conectadas em uma rede local, sem interferência da rede externa no tráfego de dados. Uma das máquinas contém a ferramenta Cbench a fim de gerar carga para o controlador. E a outra possui os controladores a serem testados instalados. Assim, ao iniciar o experimento, apenas o controlador que será testado naquele momento estará sendo executado. O Collectl, instalado na mesma máquina em que se encontram os controladores, realiza a coleta de toda a informação, incluindo aspectos relacionados à memória, disco, rede, entre outros. Foram utilizados conjuntos de dados gerados pelo Cbench com as seguintes parametrizações: 16 *switches* emulados e 1000 *hosts* por cada *switch* emulado. Cada iteração acontecia com ciclos de 30 minutos de tráfego do CBench, parando por 10 minutos, e em seguida retornava ao funcionamento, com tempo total de 12 horas. A Tabela 1 mostra um resumo desse cenário, exibindo as especificações de cada controlador juntamente com as configurações do experimento.

Tabela 1 – Fluxos gerados para cada controlador.

	Linguagem	MultiThread	Openflow	Switches	Hosts	Tempo	Fluxos/s
Floodlight	Java	Sim	1.0	16	1000	12h	111.639,66
POX	Python	Não	1.0	16	1000	12h	6.675,51
Ryu	Python	Não	1.0	16	1000	12h	10.636,40

O Cbench gerou cargas diferentes para os controladores dificultando uma análise comparativa com os resultados brutos do experimento. A fim de que todos os controladores possuíssem a mesma base para comparação dos resultados foi adotada uma forma de tornar a comparação dos resultados mais justa. Isso se deu através da relação entre o consumo dos recursos da máquina hospedeira e os fluxos gerados por segundo pelo gerador de tráfego. Em suma, é possível quantificar os recursos consumidos para que o controlador pudesse tratar um único fluxo. Neste trabalho, as máquinas em que foram realizados os testes têm as seguintes configurações: processador Intel (R) Xeon (R) CPU E3-1240 V2 3.40 GHz, sistema operacional debian 7.8, 8 GB de memória RAM, 16 GB de espaço para *swap* e paginação de 4 GB.

¹ <http://www.projectfloodlight.org/floodlight/>. Acessado em Abril de 2015.

² <http://www.noxrepo.org/pox/about-pox/>. Acessado em Abril de 2015.

³ <http://osrg.github.io/ryu/>. Acessado em Abril de 2015.

4. Resultados

Os resultados foram obtidos através dos experimentos utilizando os diversos controladores e serão apresentados a seguir.

4.1 Utilização de Memória

Como pode ser visto na Figura 1, existem diferenças na forma como cada um dos controladores lida com a memória. Na Figura 1 (a), pode-se notar que o POX armazena uma quantidade maior de informações no buffer comparando-o com os outros controladores. O Floodlight, por sua vez, quase não armazena dados no buffer. Enquanto isso, o Ryu apresentou um consumo médio, sendo 33% menor que o Floodlight. Já na Figura 1 (b) são apresentadas as informações relativas ao cache, exibindo que tanto o POX quanto o Floodlight praticamente não armazenaram informações nessa memória. Por outro lado, o Ryu possuiu um consumo alto desse tipo de memória. Da mesma forma que a Figura 1 (b), a Figura 1 (c) mostrou o Ryu se sobressaindo em relação aos demais. Desta vez, foi relacionado à quantidade de memória RAM usada. Por fim, conforme apresentada na Figura 1 (d), o POX utilizou uma maior área de memória para swap. O Ryu, por sua vez, utilizou 38% a menos que o POX. Enquanto isso, o Floodlight possuiu pouco uso dessa memória.

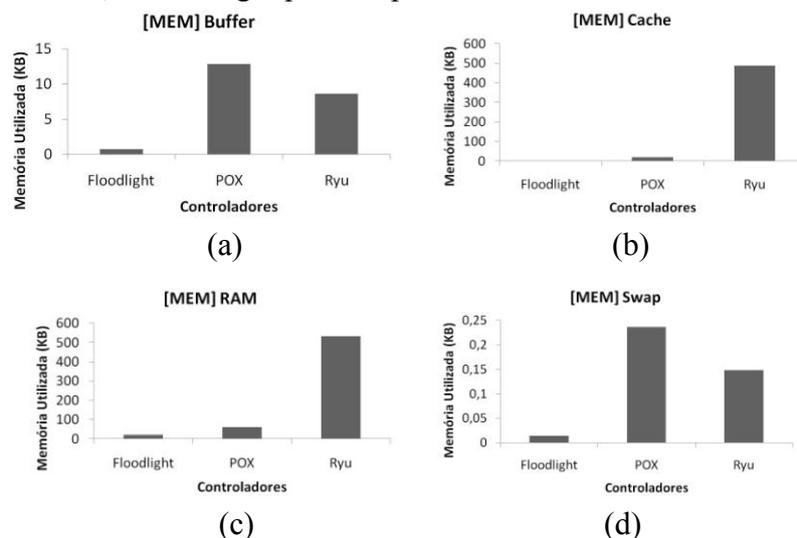


Figura 1 – Consumo de memória no hospedeiro.

4.2 Paginação

A paginação realizada pelo hospedeiro tem impacto na forma como a máquina armazena suas informações e demanda processamento. Para que fosse mais fácil a visualização dos dados, foram criados dois gráficos relativos à paginação, conforme apresentados na Figura 2.

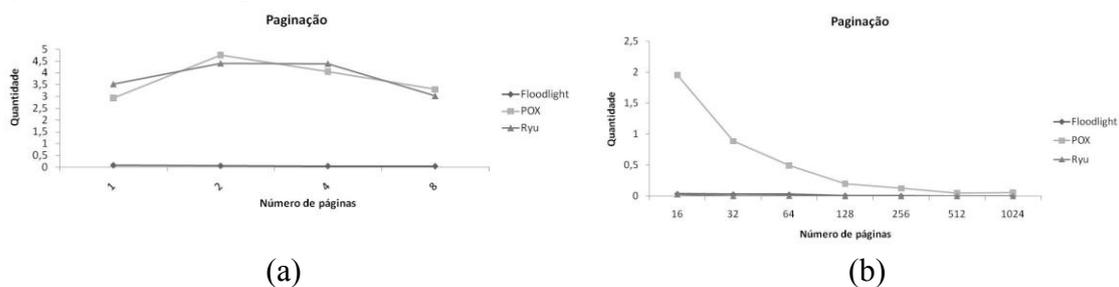


Figura 2 - Paginação do Hospedeiro.

Na Figura 2 (a), pode-se observar a paginação realizada com 1 até 8 páginas. Tanto o POX quanto o Ryu possuem uma maior quantidade de paginação em relação a poucas páginas, enquanto que o Floodlight praticamente não pagina. Já para um número de páginas entre 16 e 1024, como apresentado na Figura 2 (b), o comportamento dos controladores mostra-se diferente. O POX continua realizando muitas paginações, enquanto que os outros dois possuem valores desprezíveis.

4.3 Consumo de CPU

A Figura 3 exibe o comportamento dos recursos da CPU nos experimentos.

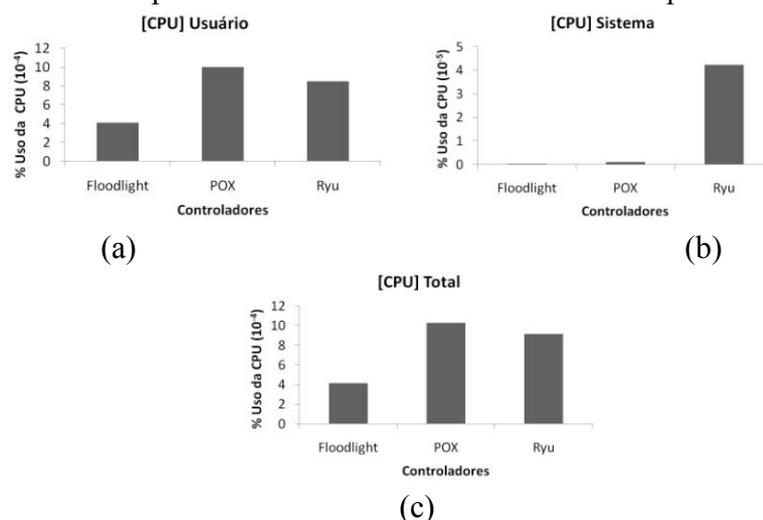


Figura 3 - Consumo de CPU no Hospedeiro.

Assim, como se pode observar na Figura 3 (a), o consumo da CPU para as requisições oriundas do usuário são maiores para o Ryu e o POX. Este último controlador apresentou valores um pouco maiores desse tipo de requisição, em torno de 17%. Enquanto isso, o Floodlight quase não fez requisições de usuário quando comparados aos outros controladores. Já na Figura 3 (b), são exibidos os resultados para as requisições provenientes do sistema. Nesse gráfico, tanto o Floodlight quanto o POX praticamente não apresentam valores expressivos. Por outro lado, o Ryu possui grandes quantidades de requisições do sistema. Por fim, na Figura 3 (c) é mostrado o consumo de CPU demandado por cada um dos controladores, considerando todos os tipos de requisições. O POX apresenta-se como o que mais CPU é exigido por cada fluxo de tráfego, seguido pelo Ryu, com valores em torno de 12% maiores. Já o Floodlight não demanda muito processamento para os fluxos que recebe.

4.4 Utilização de Disco

A Figura 4 mostra o consumo de disco no hospedeiro tanto na parte de escrita quanto na de leitura.

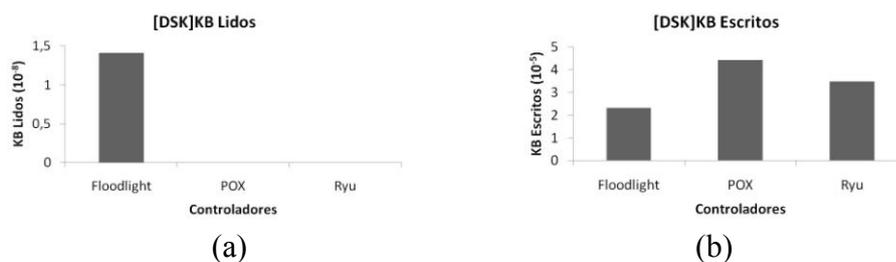


Figura 4 - Consumo de Disco no Hospedeiro.

Assim, na Figura 4 (a) são apresentados os dados referentes às leituras realizadas do disco. O POX e o Ryu não realizam leitura de dados do disco durante o seu uso, apenas o Floodlight. Em relação à Figura 4 (b), pode ser visto que todos os controladores realizaram escritas no disco. O controlador POX foi o que realizou mais escrita, seguido pelo Ryu, e depois pelo Floodlight. Os resultados do Ryu são 22% menores que o POX, enquanto que o Floodlight é 48% menor relacionando-o ao mesmo controlador. A Tabela 2 mostra de forma resumida o consumo dos recursos demandados por cada controlador. Assim, podem ser feitas comparações entre os controladores analisando essas informações qualitativas.

Tabela 2 – Resumo dos recursos consumidos pelos controladores.

	Memória	Paginação	CPU	Disco
Floodlight	Baixo	Baixo	Baixo	Alto
POX	Médio	Alto	Alto	Baixo
Ryu	Alto	Alto	Alto	Baixo

5. Conclusão

Neste trabalho, foi realizada uma análise dos recursos consumidos na máquina hospedeira quando da utilização dos controladores SDN. Como foi possível observar nos resultados apresentados, cada um dos controladores possui comportamento diferente em relação aos recursos da máquina. Em resumo, o Floodlight pelo *multithread* mostrou ser o que possui o consumo mais eficiente dos recursos do hospedeiro para cada fluxo recebido, demandando pouco poder de processamento, espaço de armazenamento e uso do disco. Além do mais, ele não desperdiça o espaço utilizado em disco. O POX foi o que apresentou a segunda melhor gestão dos recursos, com pouco consumo de memória, porém com um alto consumo de processamento, explicado pelo fato desse controlador realizar bastante paginação enquanto que organiza melhor a memória. Já o Ryu teve o pior desempenho dos controladores estudados, devido ao alto consumo de todos os tipos de memória, alto processamento e pouca utilização dos recursos do disco. Como trabalho futuro, pretende-se aumentar a quantidade de controladores utilizados e também o tempo do experimento para avaliar aspectos de *software aging*.

Referências

- Alencar, A. et al., (2014), “How Software Aging Affects SDN: A View on the Controllers”, Global Information Infrastructure and Networking Symposium (GIIS), 2014, pp. 1-6.
- Feldmann, A. et al., (2002), “IP network configuration for intradomain traffic engineering”, Network, IEEE, vol. 15, no. 5, pp. 46-57.
- Monaco, M. et al., (2013), “Applying operating system principles to SDN controller design”, HotNets-XII Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks, no.2.
- Nunes, B. A. A. et al., (2014) “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks”, Communications Surveys & Tutorials, IEEE, vol. 16, no. 3, pp. 1617-1634.
- Shalimov, A. et al., (2013) “Advanced Study of SDN/OpenFlow controllers”, CEE-SECR '13 Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia.
- Wang, F. et al., (2014) “A Research on High-Performance SDN Controller”, Cloud Computing and Big Data (CCBD), 2014 International Conference on, pp. 168-174.