INSTRUMENTAÇÃO E MEDIÇÃO DO DESEMPENHO NOS DRIVERS DE REDE DO KVM

Bruno Gomes Correia Rodrigues[1]; Djamel Sadok[2]

[1]Estudante do Curso de Engenharia da computação - CIn – UFPE; E-mail: bgcr@cin.ufpe.br,
[2]Docente/pesquisador do Centro de Informática – CIn – UFPE. E-mail: jamel@gprt.ufpe.br.

Sumário: A virtualização vem se popularizando cada vez mais nos dias atuais, sendo utilizada por várias empresas. Porém, ainda existem desafios ao se utilizar virtualização, como garantir eficiência, o controle de recursos e a equivalência entre os sistemas. Dentre as formas de virtualização, uma das mais conhecidas e utilizadas é a virtualização utilizando o *KVM*. O *KVM* é uma forma de virtualizar sendo mais leve e fácil de utilizar, pois o código do *KVM* já vem inserido dentro do *Kernel Linux*, desde a versão 2.6.20. Como o simples envio e recebimento de pacotes sofre com os efeitos da virtualização, este trabalho utiliza uma técnica de envio e recebimento de pacotes, combinada com algumas otimizações de virtualização que utilizam o *KVM*, para avaliar qual cenário se aproxima do cenário ideal, ou seja, um cenário que não sofre com os efeitos da virtualização.

Palavras–chave: Elvis; *KVM*; *Kernel*; *PF RING*; *ping*

INTRODUÇÃO

Rede de computadores vem evoluindo desde o seu surgimento, com o objetivo de suprir as necessidades dos usuários. Um dos grandes problemas dessa área é a Ossificação da Internet, ou seja, a dificuldade em modificar o núcleo da rede, pelo fato do mesmo ser composto em sua maior parte por hardware e protocolos proprietários de difíceis acessos e modificação. A virtualização foi desenvolvida na década de 1960 pela *IBM*, originalmente para particionar os *mainframes* em várias instâncias lógicas e executar em um único *mainframe* físico como o hospedeiro. Este recurso foi inventado porque a gestão dos computadores *mainframe* tornou-se complicada. Os cientistas perceberam que esta capacidade de particionamento permite que vários processos e aplicações sejam executados ao mesmo tempo, aumentando assim a eficiência do ambiente e diminuindo a sobrecarga de manutenção. Nos dias atuais, a virtualização de rede é vista como uma solução para o problema de Ossificação da Internet, pois a virtualização de rede oferece flexibilidade, diversidade, segurança e uma capacidade maior para realização de gerenciamentos. Entretanto, esta tecnologia apresenta algumas desvantagens, sobretudo, pelo fato de que cada sistema hospedeiro deve ser virtualizado. Existe a possibilidade de uma sobrecarga ser gerada no ambiente real, algo que pode ocasionar uma perda de desempenho nos ambientes virtualizados. Além disso, existem também alguns desafios ao utilizar virtualização, como garantir eficiência, controle de recursos e equivalência entre os sistemas [1].

MATERIAIS E MÉTODOS

Inicialmente foi realizado o estudo de estado da arte através da leitura de artigos sobre o tema na literatura e alguns experimentos que serviram como base para esta pesquisa, encontrados nas bases disponíveis online, como Google Scholar [2], Microsoft Academic Search [3], biblioteca digital ACM [4] e IEEE [5]. Em seguida, o bolsista realizou as implementações necessárias na ferramenta do *PF RING*[6] e instalou o *Kernel* necessário para utilizar o *ELVIS* [7]. Após ter colocado em funcionamento todas as edições

necessárias nos códigos, foram criados *scripts* com a finalidade de monitorar os resultados que foram gerados pela ferramenta do *PF RING*. Com todos os *scripts* em funcionamentos, os experimentos de *ping* foram realizados bem como a coleta dos dados.

RESULTADOS

Tabela 1 – Resultados das perdas de pacotes de cada cenário

Cenários	Média
Cenário 1	0%
Cenário 2	0%
Cenário 3	0%
Cenário 4	0%
Cenário 5	0%

Tabela 2 – Resultados do *RTT* de cada cenário

Cenários	Média	Desvio Padrão
Cenário 1	$1.94e^{-7}$	$1.40e^{-8}$
Cenário 2	$1.24e^{-7}$	$4.58e^{-8}$
Cenário 3	$2.43e^{-7}$	$1.35e^{-7}$
Cenário 4	$1.28e^{-7}$	$7.30e^{-8}$
Cenário 5	$2.69e^{-7}$	$1.05e^{-7}$

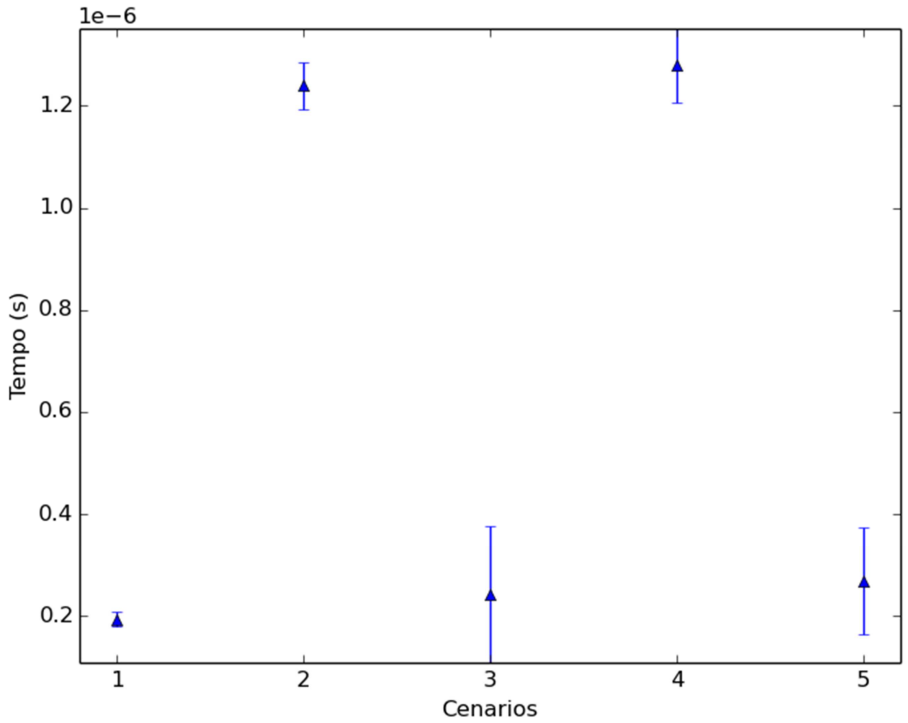

Figura 1 – Comparação entre os 5 cenários

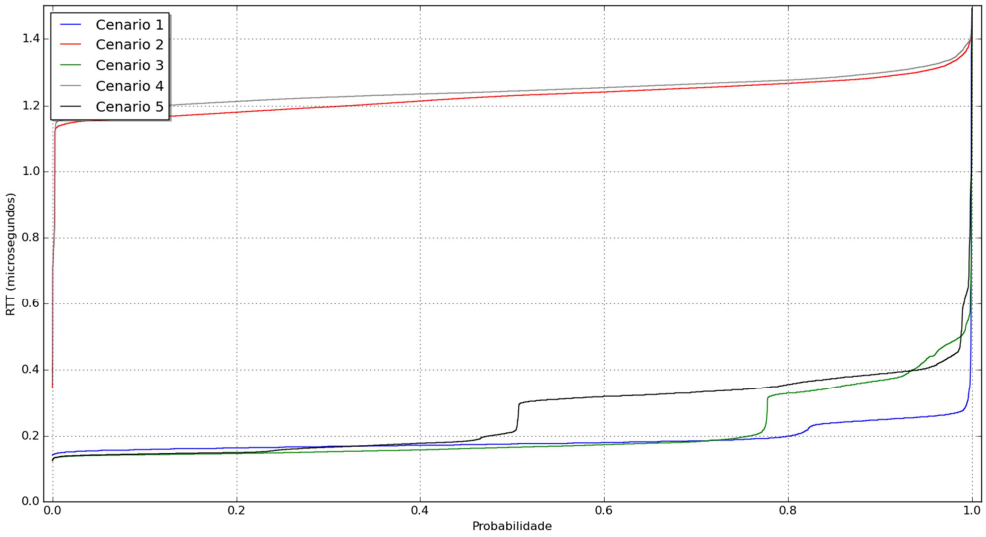

Figura 2 – *CDF* dos cenários

A partir da Tabela 1, verifica-se que não ocorreu perda de pacotes em todos os cenários. As ferramentas utilizadas, *pfsend* e *pfcount*, juntamente com o *framework* do *PF RING*, são fatores que influenciam em uma baixa perda de pacotes. Após realizar uma única média e colocar todos os cinco cenários em uma única figura, como mostrado na Figura 1 e na Tabela 2, observa-se melhor o que foi dito anteriormente, que os cenários 3 e 5, e os cenários 2 e 4 ficam muito próximos um do outro, eles são estatisticamente equivalentes. Dentre estes, os cenários 3 e 5 são os que chegam mais próximo do primeiro cenário (*baseline*), pois se encontram na mesma ordem de magnitude. Na Figura 2, foi calculado o *CDF* a partir dos resultados dos experimentos em cada cenário, e como se pode observar, os cenários 3 e 5 de fato são os que se aproximam mais do cenário 1. Entretanto, o cenário 3 esteve aproximadamente 78% do tempo próximo ao *baseline*, enquanto o cenário 5 esteve apenas 50%.

DISCUSSÃO

Analisando os resultados obtidos, pode-se perceber que o uso do módulo de *Kernel* do *vPF RING* não se faz necessário, por mostrar resultados similares aos cenários em que ele não foi utilizado. Isto talvez seja devido ao fato dos experimentos não levar em conta outros fatores, como por exemplo, múltiplas máquinas virtuais concorrendo por recursos. Por outro lado, a otimização ao realizar a virtualização com o *ELVIS*, mostrou-se indispensável caso o sistema seja virtualizado utilizando *virtio* no *KVM* [8], pois os resultados são próximos ao cenário com ambiente não virtualizado (*baseline*).

CONCLUSÕES

Com base nos resultados obtidos, pode-se concluir que a utilização do módulo *virtual* do *PF RING* não é necessária para realizar a captura de pacotes, quando o módulo do *PF RING* já está sendo utilizado. Por outro lado, a otimização feita no módulo do *virtio* no *KVM*, mostrou-se indispensável. Com isto, o melhor cenário ao utilizar virtualização no *KVM* para captura de pacotes dentre os que foram estudados, seria utilizar a otimização do *ELVIS*. Uma proposta para trabalhos futuros seria realizar os mesmos testes com múltiplas máquinas virtuais executando na mesma máquina física, onde seria interessante avaliar o consumo de processamento da máquina física, além de comprovar se os resultados obtidos

atualmente se repetem em um ambiente que possui máquinas virtuais disputando recursos. Outra proposta seria utilizar uma bateria de teste similar a esta, porém utilizando conteinerização, que pode ser uma alternativa mais leve para realizar a virtualização completa.

AGRADECIMENTOS

Em agradecimento a PROPESQ por disponibilizar o auxílio financeiro e ao grupo do GPRT pelo fornecimento do espaço e do equipamento necessário para que a pesquisa fosse realizada sem problemas. Agradeço também ao professor orientador pelo suporte durante toda a pesquisa.

REFERÊNCIAS

[1] Sahoo, J., Mohapatra, S., Lath, R., Abril 2010. Virtualization: A survey on concepts, taxonomy and associated security issues. Em: Computer and Network Technology (ICCNT), 2010 Second International Conference on. Pp. 222-226.

[2] Google Scholar. http://scholar.google.com.br. Acesso em: 30 setembro 2014.

[3] Microsoft Academic Search. http://academic.research.microsoft.com. Acesso em: 26 julho 2014.

[4] Biblioteca digital ACM . http://dl.acm.org. Acesso em: 30 setembro 2014.

[5] IEEE . http://ieeexplore.ieee.org. Acesso em: 30 setembro 2014.

[6] Cardigliano, A., Deri, L., Gasparakis, J., Fusco, F., 2011. vpf ring: Towards wire-speed network monitoring using virtual machines. Em: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference. IMC'11. ACM, New York, NY, USA, pp. 533-548.

[7] Har'El, N., Gordon, A., Landau, A., Ben-Yehuda, M., Traeger, A., Ladelsky, R., 2013. Efficient and scalable paravirtual i/o system. Em: USENIX Annual Technical Conference. Pp. 231-242.

[8] KVM. http://www.linux-kvm.org/page/Main_Page. Acesso em: 30 setembro 2014