XI CONNEPI
CONGRESSO NORTE NORDESTE DE PESQUISA E INOVAÇÃO
MACEIÓ · ALAGOAS · 2016

Troca de contexto de CPU e sua relação com a degradação de desempenho em ambientes virtualizados no o Xen Server

André Henrique Sousa de Menezes[1], Kelvin Romero Meira de Oliveira Corderio[2], Leandro Cavalcanti de Almeida[3], Paulo Ditarso Maciel Junior [4]

[1]Discente de graduação em redes de computadores - IFPB. Bolsista de iniciação científica do PIBIT/CNPq do IFPB. e-mail: menezescode@gmail.com; [2]Discente de graduação em redes de computadores - IFPB. Voluntário de iniciação científica do PIBIT/CNPq do IFPB. e-mail: kelvin.romero@academico.ifpb.edu.br; [3]Professor do curso de redes de computadores - IFPB. e-mail: leandro.almeida@ifpb.edu.br; [4]Professor do curso de redes de computadores - IFPB. e-mail: paulo.maciel@ifpb.edu.br

RESUMO: Virtualização de servidores é uma tecnologia indispensável na infraestrutura de TI hoje em dia, ela permite que várias máquinas virtuais compartilhem o mesmo *hardware*, promovendo economia de recursos, como espaço físico, refrigeração e energia, também evitando que os recursos de *hardware* sejam subutilizados. Técnicas como a paravirtualização, investigada neste artigo, podem criar uma pequena sobrecarga no sistema enquanto realizam as rotinas do *hypervisor*, que é responsável por receber e gerenciar as chamadas de sistema das máquinas virtuais. Neste artigo é analisado a relação da troca de contextos da CPU com a degradação de desempenho de máquinas virtuais que residem num mesmo hospedeiro físico durante um ataque de negação de serviço distribuído. No cenário construído, duas máquinas virtuais são hospedadas utilizando o *hypervisor* Xen Server versão 7.2.1511. Foram coletados dados relacionados as trocas de contexto e interrupções de CPU diretamente no *hypervisor*. A partir destes dados é possível conjecturar que durante um ataque de negação de serviço, há um grande volume de interrupções de alta prioridade que são convertidas em trocas de contexto.

Palavras–chave: hypervisor, interrupções, contexto, DDoS.

CPU context switching and its relationship with the degradation of performance in virtualized environments in Xen Server

ABSTRACT: Server virtualization is an indispensable technology in IT infrastructure today, it allows multiple virtual machines to share the same hardware, promoting resource savings, such as physical space, cooling and power, also avoiding that hardware resources are underutilized. Techniques such as paravirtualization, investigated in this article, can create a small overhead in the system while performing the routines hypervisor, which is responsible for receiving and managing the system calls the virtual machines. This paper analyzed the relationship of exchange of CPU settings with the degradation of performance of virtual machines that reside on the same physical host for a distributed denial of service attack. In constructed scenario, two virtual machines are hosted using the hypervisor Xen Server version 2.7.1511. Was collected data regarding the context switches and CPU interrupts directly in the hypervisor. From these data it is possible to conjecture that for a denial of service attack, there is a large volume of high-priority interrupts that are converted into context switches.

KEYWORDS: hypervisor, interruptions, context, DDoS.

INTRODUÇÃO

Atualmente uma das mais importantes tecnologias utilizadas no setor de tecnologia da informação (TI) é a virtualização de servidores. Que consiste em máquinas virtuais (VMs) compartilhando simultaneamente o mesmo *hardware* (*Host*). Nesta técnica o compartilhamento de recursos físicos permite uma melhor utilização do *hardware* ao mesmo tempo que mantém o isolamento dos sistemas hospedados. Neste artigo o hospedeiro utilizado foi o Xen Server 7.2.1511. O Xen Server possui um *hypervisor* responsável por receber chamadas do sistema (*System calls*) e de acesso ao *hardware*, provenientes das máquinas hospedadas (*Guests*). O *hypervisor* recebe essas requisições e repassa as instruções para o *hardware*. Esse cenário é chamado de paravirtualização, no qual o núcleo do sistema operacional hospedado não interage diretamente com o *hardware* onde precisar de modificações para funcionar adequadamente. Por outro lado, a paravirtualização não exige *hardware* modificado para suportar máquinas virtuais.

Diversas empresas utilizam essas técnicas para gerar economia, facilitar a manutenção e atender demandas internas, além disso é comum que a virtualização seja utilizada como a base para serviços de infraestrutura. No cenário de infraestrutura como serviço cada usuário ou locatário gerencia sua própria infraestrutura de TI, quando e conforme necessário. Um exemplo seria de uma empresa que precisa montar um servidor Web e no lugar de construir localmente a infraestrutura ela pode simplesmente alugar algumas VMs em um *datacenter* externo.

Neste caso é indispensável atestar os níveis de isolamento de máquinas virtuais (NIKOUNIA e MOHAMMADI 2015), em um experimento realizado anteriormente foi observado que durante um ataque de negação de serviço distribuído (DDoS – *Distributed Denial of Service*) à uma máquina virtual em um servidor Xen, outras VMs que compartilham o *hardware* sofrem degradação no desempenho. Através dos dados coletados foram identificados maior tempo de resposta dos servidores Web e picos de processamento nas VMs vizinhas à máquina virtual atacada.

Anteriormente foi observado (SHEA e LYU, 2012) que a degradação de desempenho em ambientes virtualizadas tem uma relação direta, entre outros fatores, com o aumento não

linear da troca de contextos da CPU. No trabalho desenvolvido identificaram que durante um ataque DDoS o sistema não virtualizado um número menor de trocas de contexto do que os ambientes virtualizados. Neste caso os pesquisadores propuseram uma mudança no *kernel* do KVM (outra ferramenta de virtualização) que reduz a quantidade de trocas de contexto na CPU, resultando em um melhor desempenho do sistema.

As trocas de contexto da CPU e as interrupções, alvo da análise desse artigo são de extrema relevância para o desempenho da infraestrutura. A troca de contexto é a transferência de controle da CPU de um processo para outro. Eles ocorrem quando o processo conclui seu ciclo ou o *hardware* ou o *software* enviam uma interrupção e o sistema operacional recupera forçadamente o controle da CPU (REMZI e ANDREA, 2016).

Este trabalho teve como objetivo realizar uma análise do comportamento do Xen Server durante um ataque de negação de serviço distribuído, mais especificamente informações coletadas no Dom0[1] e a relação de troca de contexto e interrupções da CPU sob o ataque. O

MATERIAL E MÉTODOS

Esta seção descreve o experimento e todos os dispositivos de *hardware* e *software* utilizados para execução, captura e processamento das informações.

O experimento foi executado em um ambiente controlado. Conforme pode ser visualizado na Figura 1, é elaborado para maior controle sobre as variáveis, inviável em um cenário de provedor de IaaS (*Infrastructure as a Service*), no qual não se poderia instalar as ferramentas necessárias para execução dos experimentos e a captura de informações, outro problema seria a instabilidade conexão para um servidor Web externo.

O servidor utilizado possui um processador Intel i7 de quatro núcleos, 32 GB de memória RAM, um disco rígido de 1 TB e duas interfaces de rede *Gigabit Ethernet*. O sistema operacional utilizado como *hypervisor* foi o XenServer da Citrix Systems Inc., versão 7.2.1511. Neste servidor, foram criadas duas máquinas virtuais idênticas, contendo cada uma a seguinte configuração: 1 núcleo de processador virtual, 1 GB de memória RAM, 10GB de disco rígido e uma interface de rede virtual. O sistema operacional utilizado nas máquinas

[1] Dom0 é uma máquina virtual que possui permissões para controlar o *hypervisor*, e consequentemente o *hardware*.

virtuais foi o Debian GNU/Linux versão 8.2. As duas máquinas virtuais executaram o servidor Web Apache versão 2.

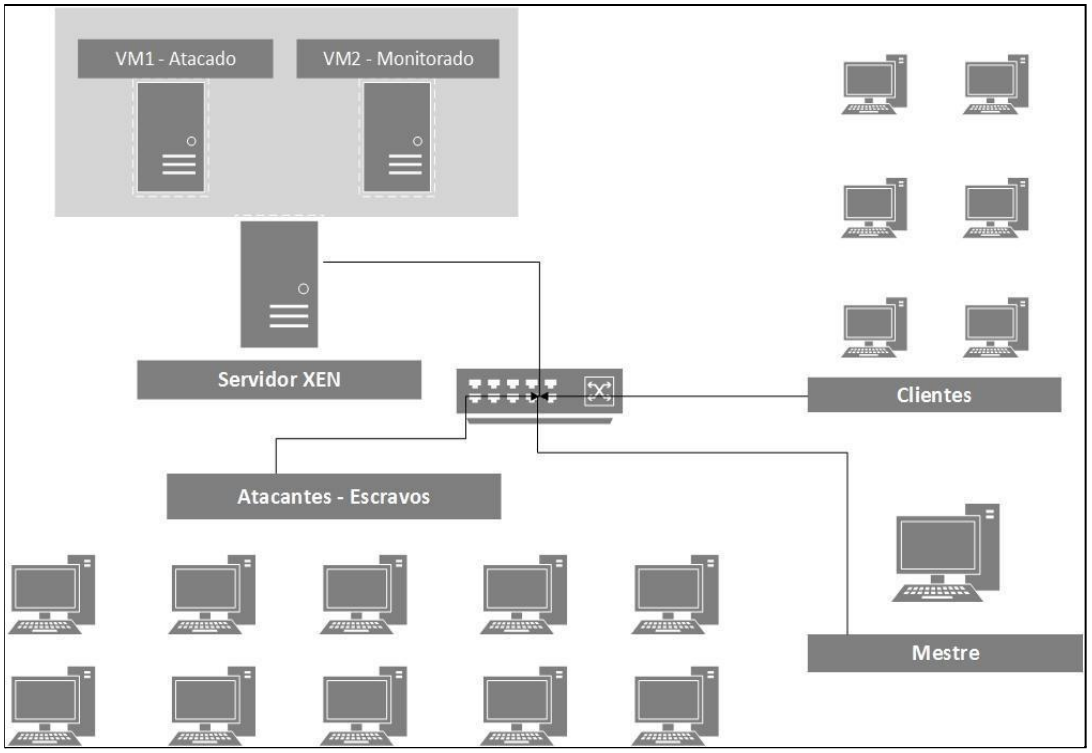

Figura 1. Cenário de avaliação utilizado nos experimentos. IFPB, 2016

As demais máquinas físicas utilizadas no experimento possuem cada uma as seguintes configurações: um processador Intel I5, 8 GB de memória RAM, um disco rígido de 500 GB e uma interface de rede *Fast Ethernet*. Utilizando o sistema operacional Windows 7. Nesse sistema foram instalados o VirtualBox e o Vagrant, sendo o segundo usado somente para automatizar o processo de criação das máquinas virtuais hospedadas pelo VirtualBox, uma VM em cada computador. As máquinas virtuais utilizadas para os atacantes (*slaves*), clientes e mestre contavam com os seguintes recursos: 256 MB de memória RAM, 40 GB de disco rígido, 2 interfaces de rede virtuais no

O switch utilizado foi um Cisco Catalyst 2960, com 24 portas *Fast Ethernet* e 2 portas *Gigabit Ethernet*. O sistema operacional utilizado no ativo de rede foi o IOS (*Internetwork Operating System*) versão 12.2. É importante ressaltar que, apesar de todo o tráfego dos experimentos utilizarem um único switch físico, as máquinas VM1 e VM2 foram

configuradas em redes virtuais diferentes, estabelecidas internamente no *hypervisor*, no master e nos atacantes (*slaves*) e apenas 1 para o cliente, pois este último realiza apenas solicitações "genuínas" ou seja, simula o acesso normal ao servidor Web.

O experimento foi executado utilizando as duas interfaces *Gigabit Ethernet* ao mesmo tempo em um esquema de agregação de link, somando as velocidades das interfaces. Ou seja, neste caso, as VMs compartilhavam uma interface com capacidade de 2 Gbps de tráfego. Para o funcionamento da agregação de tráfego, foi necessário configurar as duas portas *Gigabit Ethernet* no switch para executarem o protocolo LACP (*Link Aggregation Control Protocol*).

Para realizar o experimento, um conjunto de scripts escritos em linguagem Shell foram construídos para automatizar os passos necessários de cada rodada. O esquema de funcionamento dos scripts desenvolvidos e as ferramentas utilizadas podem ser vistos na Figura 2.

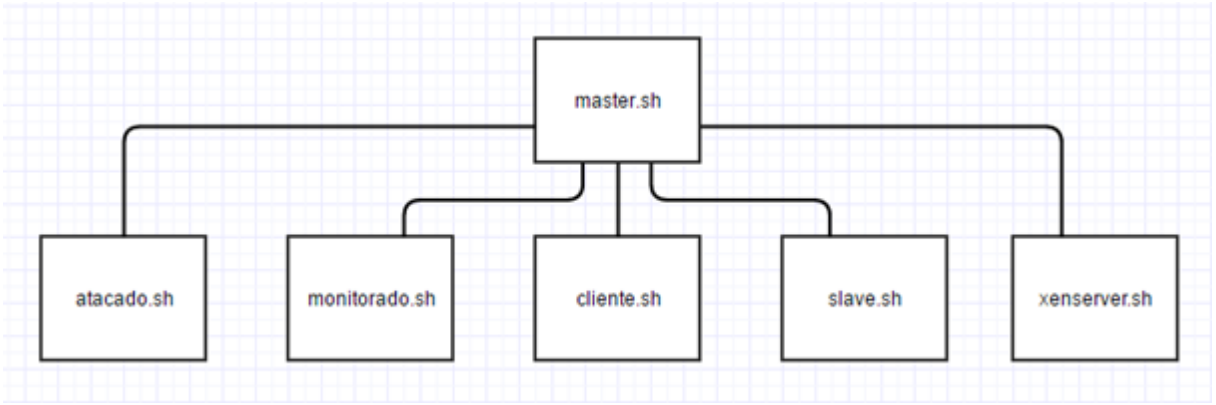

Figura 2. Esquema dos Scripts. IFPB, 2016

O experimento foi composto por 60 rodadas. Cada rodada do experimento passou por quatro fases distintas, são elas: (i) Inicialização, (ii) Execução, (iii) Coleta e (iv) Limpeza. O script master.sh, localizado no Mestre, basicamente é responsável por inicializar todos os outros scripts localizados nos demais dispositivos, utilizando uma sessão com o protocolo SSH (*Secure Shell*).

Nas máquinas virtuais, VM1 (Atacado) e VM2 (Monitorado), as ferramentas Sysbench e Stress-ng, que avaliam o desempenho de um sistema realizando uma sobrecarga intencional

e igual em todas as rodadas. Foram utilizadas para simular uma carga de trabalho sintética, buscando aproximar o experimento de um ambiente real, no qual os servidores estariam em constante trabalho.

No servidor Xen foi utilizado o *software* Vmstat, disponível nativamente em distribuições baseadas em Linux, para coletar dados relativos as interrupções e as trocas de contexto realizadas da CPU do *hypervisor*.

Foram realizadas 30 rodadas sem onde o comportamento simulado é de 6 clientes fazendo acesso Web a VM1, e as VM 1 e 2 hospedadas pelo *hypervisor* realizando uma carga padrão de trabalho através do Sysbench. Nas 30 rodadas com ataque o cenário se repete com o acréscimo de 10 computadores *slaves*, que disparam o ataque para a VM1.

RESULTADOS E DISCUSSÃO

Foi observado que durante o ataque DDoS existe um aumento expressivo no número de interrupções. Enquanto realiza o processamento de controle, o número médio de interrupções é de 25.295, porém durante ataque esse valor salta para 141.157. Também foi constatado um aumento da média de Trocas de contexto da CPU, de 13.500 para 95.369. Inicialmente o aumento é considerado normal, pois um ataque irá aumentar o número de processos. Porem esse o crescimento de interrupções não é linear, enquanto as interrupções cresceram 82,08% as trocas de contexto aumentaram 85,84% na média. Analisando mais cuidadosamente os dados extraídos, foi observado que durante o ataque a CPU converte 14,23% mais interrupções em trocas de contexto da CPU do que no cenário sem ataque. Isso significa que o ataque DDoS produz um pico de interrupções de alta prioridade pra CPU, conforme visualizado na Tabela 1 que possui a taxa de conversão com e sem ataque.

Tabela 1. Taxa de conversão de interrupções em trocas de contexto no Dom0

Métrica (médias)	Sem ataque (DoS)	Com ataque (DoS)
Interrupções	25.295	141.157
Trocas de contexto	13.504	95.369
Taxa de conversão	53,39%	67,63%

Para uma melhor compreensão do que ocorre, ocorre, a Figura 3 representa duas situações, onde a linha de cor vermelho representa o crescimento esperado nas trocas de contexto em função das interrupções e a linha cor azul representa a relação verificada através do experimento de ataque DDoS.

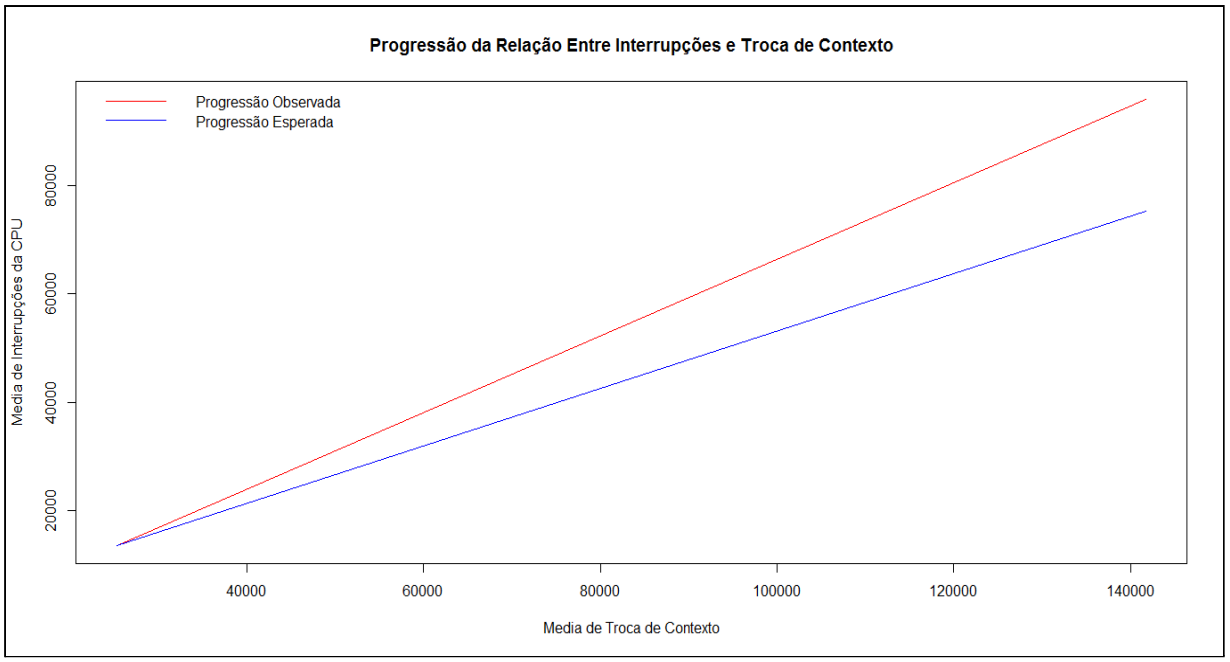

Figura3. Gráfico da progressão da relação entre Interrupções e Troca de Contexto, utilizando a média de interrupções de CPU em função da média de troca de contexto, 2016

CONCLUSÕES

O aumento da troca de contextos durante um ataque de DDoS torna clara que são criados um maior número de interrupções de alta prioridade na CPU. Acarretando em um maior número de trocas de contexto na CPU. O cenário ainda é limitado para determinar o tipo de interrupção de alta prioridade gerada e a proporção do número de interrupções para o volume do ataque de negação de serviço.

Pode-se inferir que a interrupção de alta prioridade seja de hardware oriunda da placa de rede, porém não há dados que possam garantir essa hipótese, ficando então como sugestões de trabalhos futuros a identificação da origem das interrupções de alta prioridade gerada em

grande volume pelo ataque DDoS e também uma solução (verificada viabilidade) para o tratamento deste tipo de interrupção.

AGRADECIMENTOS

Este trabalho foi realizado com o suporte do Programa Institucional de Voluntários de Iniciação Científica – PIVICT/PRPIPG/IFPB, Edital nº 01/2016.

REFERÊNCIAS

Shea, R. and Liu, J. (2012). **Understanding the Impact of Denial of Service Attacks on Virtual Machines**. In Quality of Service (IWQoS), 2012 IEEE 20th International Workshop on, pages 1–9.

Nikounia, S. and Mohammadi, S. (2015). **Hypervisor and Neighbors' Noise: Performance Degradation in Virtualized Environments**. Services Computing, IEEE Transactions on, PP(99):1–1.

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. **Operating Systems: Three Easy Pieces**. 0.91. ed. Arpaci-Dusseau Books, LLC, 2016. 666 p.