

USO DE MÁQUINAS VIRTUAIS PARA O DESENVOLVIMENTO DE AMBIENTES DE PROJETOS

Felipe Terror Lima¹, Marcelo Marques Gomes², Paulo Guilherme Seifer³,
Roberto Scalco⁴

Abstract — *This work presents a proposal for the use of virtual machines as environment of development of projects and as environment of learning for disciplines Operational Systems and Networks I and II of the Escola de Engenharia Mauá. During the creation of a project of software it's common that diverse configurations in the computer are made, installing software and run services, until reaching an ideal situation for the conduction of the project. For each computer that the developer will have to use, these configurations have to be made again. With the use of virtual machine all the work will be made one time only and its image can be copied for many others computers of a network. Moreover, also it's possible to use these virtual machines to demonstrate the behavior of diverse operational systems as well as applying configurations in networks and visualizing the packages transmitted in virtual networks between machines.*

Index Terms — *Development Environment, Operational Systems, Virtual Machine, Virtualization.*

INTRODUÇÃO

Embora as máquinas virtuais sejam um conceito relativamente antigo, em relação à história da computação, suas aplicações em situações didáticas têm sido pouco exploradas.

Um problema existente na metodologia das aulas de um laboratório para apresentar e estudar as características de diferentes sistemas operacionais consiste no fato de que para fazer a troca do sistema operacional em uso é necessário reiniciar o computador.

Além disso, durante o desenvolvimento de *softwares*, tornam-se necessários alguns ajustes e configurações no sistema operacional que nem sempre respondem da mesma maneira em diferentes computadores.

Este trabalho resume como um laboratório de Sistemas Operacionais pode utilizar máquinas virtuais para o estudo de seus componentes, bem como a criação de um ambiente para desenvolvimento de *softwares* controlado e replicável.

CONCEITOS UTILIZADOS

Máquinas virtuais, apesar de largamente popularizadas nos últimos anos, são um conceito antigo, dos primórdios da computação. Já na década de 1960 a IBM utilizava este conceito nos *mainframes* IBM7044, onde cada sistema executado trabalhava em uma imagem fiel do sistema físico. Existem muitas definições para máquina virtual, como em [1] que a define como “uma duplicata eficiente e isolada de uma máquina real”, ou ainda [2], [3] e [4] mostram a definição da IBM que é a de uma cópia totalmente protegida e isolada de um sistema físico.

Essencialmente uma máquina virtual provê um ambiente parcial ou totalmente simulado em *software* para a execução de um aplicativo. Normalmente são constituídas por uma camada executando em um sistema operacional *host*, instalado diretamente no computador, que provê os recursos necessários para um sistema operacional *guest*, instalado na máquina virtual. Essa camada é tipicamente chamada de *Virtual Machine Monitor* - VMM.

É na VMM que se encontram as maiores diferenças entre máquinas virtuais. Uma VMM pode simular totalmente em *software* um ambiente físico, em um processo chamado de emulação. Máquinas virtuais como o QEMU [5] realizam este processo. Aqui cada instrução do sistema operacional *guest* é interpretada e executada no sistema operacional *host*. Com esse processo é possível simular ambientes totalmente diferentes do ambiente *host* permitindo, por exemplo, a simulação de um sistema baseado em processadores Sparc[®] em um ambiente baseado em processadores Intel[®].

Uma aplicação muito popular para emuladores são os jogos. Podem ser encontrados emuladores para sistemas como Atari, Super Nintendo e outros para utilização de jogos destes em um ambiente PC típico.

Outra forma de trabalho de uma VMM é a de virtualização, onde um ambiente, o mais próximo possível do *host*, é fornecido para o *guest*. Nesta situação parte das instruções do sistema operacional *guest* é executada de forma direta no sistema *host*. Devido a limitações de segurança da arquitetura Intel[®], nem toda instrução pode ser

1 Felipe Terror Lima, Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá, Praça Mauá, 1, 09.580-900, São Caetano do Sul, SP, Brasil, felipeterror@yahoo.com.br

2 Marcelo Marques Gomes, Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá, Praça Mauá, 1, sala G-02, 09.580-900, São Caetano do Sul, SP, Brasil, marques.gomes@maua.br

3 Paulo Guilherme Seifer, Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá, Praça Mauá, 1, sala G-02, 09.580-900, São Caetano do Sul, SP, Brasil, paulo.seifer@maua.br

4 Roberto Scalco, Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá, Praça Mauá, 1, sala G-02, 09.580-900, São Caetano do Sul, SP, Brasil, roberto.scalco@maua.br

executada de forma direta, exigindo ainda a interferência direta da VMM. Essa forma de trabalho é mais rápida que a emulação. Tecnologias recentes como a Intel[®] VT e a AMD[®] Pacifica provêm recursos de *hardware* para a virtualização, tornando-a mais eficiente e diminuindo a interferência da VMM.

Por fim, outra forma de trabalho que vem se tornando popular para virtualização é a paravirtualização. Aqui existe uma integração maior entre a VMM e o sistema *guest*, o que significa que a VMM não é mais transparente. Nesse caso são necessárias modificações no sistema operacional *guest* para que esta integração ocorra. Essas modificações visam diminuir as barreiras de limitação de segurança da plataforma Intel e otimizar o trabalho do *guest* de modo a se integrar melhor com a VMM e tornar a máquina virtual mais rápida e eficiente.

PROPOSTA DE UM AMBIENTE DE TRABALHO

Ambiente de Desenvolvimento de Projetos

Durante o desenvolvimento de projetos computacionais é comum o uso de várias ferramentas, sejam elas para criar a interface do *software*, modelar várias etapas ou até outros sistemas onde o *software* final será executado. Desta maneira, a instalação do ambiente de desenvolvimento tem se tornado uma tarefa cada vez mais complexa, empregando-se horas e até dias para configurar todos os aplicativos e serviços que serão utilizados.

Esta situação se agrava quando o projeto exige a atuação de vários programadores em paralelo, desenvolvendo em ambientes que dificilmente são idênticos. Isto possibilita várias situações de incompatibilidades, como uma parte do *software* que funciona apenas no equipamento onde foi criada ou não funciona em certos computadores.

Entre os grupos de ferramentas mais usuais em um projeto, destacam-se:

- ambiente integrado de desenvolvimento (Delphi, Eclipse, Netbeans, Visual Studio .NET, etc.)
- modelagem UML (Eclipse, StarUML, NetBeans, ERWin, Rational Rose, etc)
- sistemas gerenciadores de bando de dados (Oracle, Firebird, SQL Server, MySQL, etc.)
- servidores de página *web* (Apache, IIS, Tomcat, etc.)

Algumas destas ferramentas procuram integrar diversas funções, como o Eclipse, que aceita *plugins* que permitem a modelagem de objetos. Mas em geral é preciso instalar cada ferramenta separadamente e configurar a interface entre elas manualmente, para cada computador.

No caso de projetos que visam escrever no registro do sistema operacional ou manipular diretamente a memória ocorre outro problema: o ambiente de testes se torna instável. Durante o desenvolvimento pode se tornar comum o mal funcionamento do sistema operacional, ou até a perda de parte dele.

A proposta estudada é de utilizar um ambiente virtualizado para o desenvolvimento de *softwares*, agilizando a configuração, garantindo a padronização entre os desenvolvedores e também permitindo que o ambiente (inclusive os registros de memória) seja gravado para recuperação posterior.

Ambiente Didático

Os cursos Sistemas Operacionais e Redes I e Sistemas Operacionais e Redes II, oferecidos aos alunos do curso da Engenharia Elétrica (ênfase Computação) da Escola de Engenharia Mauá, tem como parte de sua metodologia didática o uso de laboratório, fazendo com que os alunos desses cursos realizem atividades práticas.

Para o suporte destes cursos é utilizado um laboratório composto de computadores (plataforma Intel[®]) com os sistemas operacionais Linux (Slackware 12.0) e Microsoft Windows[®] 2000 Server instalados na forma “*dual boot*”.

Dentro da proposta de utilização do laboratório e dada a natureza dos cursos, é muito comum a reconfiguração (e por vezes a modificação completa) de partes importantes dos sistemas operacionais instalados. A consequência disto, muitas vezes, são danos irreversíveis nesses sistemas, obrigando a sua reinstalação.

Outro problema motivado pela atual forma de organização dos computadores do laboratório é o de demora na troca de sistema operacional. Como os cursos trabalham com ambos os sistemas operacionais, há a necessidade de, durante a aula, ocorrer a troca do sistema em uso, e esta troca consome tempo, visto que envolve a finalização de um sistema e a inicialização de outro, processos que não podem ocorrer de forma concorrente e que são demorados.

Para o ano letivo de 2008 são propostas mudanças organizacionais que visam otimizar o processo de utilização de mais de um sistema operacional e melhorar a manutenção do laboratório. Essas mudanças se baseiam na utilização de máquinas virtuais.

Serão realizadas:

- instalação de um sistema operacional único nos computadores. Esse sistema não será utilizado de forma direta pelo aluno como ferramenta para as atividades, mas sim como sistema base para gerenciamento do computador.
- instalação de uma máquina virtual. O ambiente provido pela máquina virtual é quem fornecerá ao aluno as plataformas onde serão realizadas de fato as atividades de aula.

Será utilizado como sistema base o Slackware 12.0 e como máquina virtual o VirtualBox 1.5.2.

Na máquina virtual serão instalados os sistemas operacionais a serem estudados, como os próprios Slackware 12.0 e Microsoft Windows[®] 2000 Server, além de outros como Microsoft Windows[®] 2003 Server e outras distribuições Linux.

Como a máquina virtual provê um ambiente independente de trabalho, é esperado que qualquer dano promovido no sistema *guest* em decorrência de um experimento não afete o *host*, garantindo a integridade do sistema base.

Também será amplamente utilizado o recurso de *snapshot* (cópia de segurança) visto que, preservando uma imagem íntegra de cada sistema *guest* instalado a manutenção dos sistemas para experiência fica simplificada.

Outro problema citado, o da troca do sistema em uso também fica simplificado, uma vez que é possível a utilização de várias instâncias de máquinas virtuais simultaneamente.

Existe a expectativa de perda de desempenho dos sistemas operacionais citados na máquina virtual, principalmente na utilização simultânea de mais de um sistema *guest*, mas é importante observar que não há a necessidade de grande desempenho computacional por parte dos sistemas *guest*, mas sim de controle e dinamismo na sua utilização. Os reais efeitos de perda de desempenho só poderão ser observados no transcorrer do ano letivo de 2008, quando esta proposta de organização será implementada e utilizada de fato pelos alunos.

RESULTADOS

Comparação entre Máquinas Virtuais

Para obter uma estimativa do desempenho de diversos sistemas operacionais instalados em máquinas virtuais, foram selecionados três aplicativos disponíveis na internet: QEMU 0.9.0, VirtualBox 1.5.0 [6] e VMware Server 1.0.4 [7]. A escolha desses aplicativos foi feita com base no custo de cada um e na disponibilidade de versões para Microsoft Windows® e Linux.

Como principal computador, foi utilizado um *notebook* Toshiba Satellite A135-S4467, com microprocessador Intel® Core™ 2 Duo 1,6 GHz T5200 (sem suporte a virtualização), 1 GB RAM, com sistema operacional Microsoft Windows Vista® Home Premium 32 bits.

Na tabela I estão apresentadas os sistemas operacionais que foram instalados nas máquinas virtuais selecionadas.

TABELA I
SISTEMAS OPERACIONAIS INSTALADOS NAS MÁQUINAS VIRTUAIS

QEMU	VirtualBox	VMWare Server
MS-DOS 6.22	Windows 95	MS-DOS 6.22
Windows 3.1	Windows 98	Windows 2.0
Windows 95	Windows XP	Windows 3.1
Windows 98	Windows 2003 Server	Windows 3.11
Windows XP	Linux SuSE 10.2	Windows 95
Linux SuSE 10.2	Linux Debian Etch	Windows 98
	Arch Linux	Windows XP
	Linux Slackware 12	Windows 2003 Server
	FreeBSD	Linux SuSE 10.2
	Solaris 10	Linux SuSE 10.3
		Ubuntu 7.4 64 bits
		Solaris 10

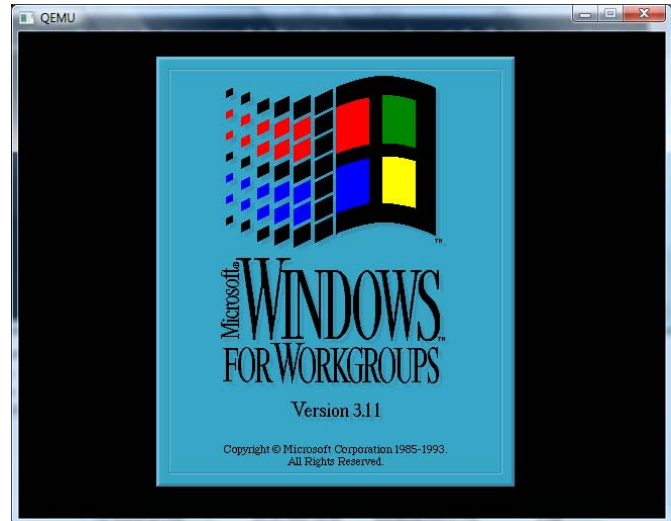


FIGURA. 1

TELA DE BOAS-VINDAS DO MICROSOFT WINDOWS 3.11 SENDO EXECUTADO NO QEMU.

Das máquinas virtuais testadas, o QEMU foi a que apresentou pior desempenho, mesmo com o pacote de aceleração KQEMU instalado. Outra desvantagem dessa máquina virtual, executado no sistema Windows, é a necessidade de se criar uma Rede Privada Virtual (VPN) para permitir a comunicação entre *host* e *guest*. Na documentação do QEMU, é sugerido o uso do aplicativo OpenVPN [8]. Tal inconveniente não aconteceu no *host* Linux testado.



FIGURA. 2

BSOD NO WINDOWS 95 SENDO EXECUTADO NO QEMU.

Como vantagem, o QEMU pode ser executado a partir de uma mídia removível, sem a necessidade da sua instalação no *host*.

O VirtualBox apresenta uma interface gráfica amigável (podendo ser configurada para a língua portuguesa) para a criação e gerenciamento de máquinas e *drives* virtuais

Dentre os sistemas operacionais instalados, deve-se salientar que o Solaris 10, instalado no VirtualBox não funcionou corretamente durante os testes.

- Firebird 1.5;
- *Internet Information Server* integrado ao Microsoft Windows 2003 Enterprise Server;
- VPN integrada ao Microsoft Windows® 2003 Enterprise Server.

As máquinas virtuais apresentaram bom desempenho durante o desenvolvimento do projeto, embora foi sensivelmente abaixo do obtido com o uso do sistema operacional diretamente. Este ambiente foi utilizado para a codificação e testes finais do sistema, sendo que a máquina virtual que executava o Microsoft Windows® XP Professional foi copiada para os três desenvolvedores, não ocorrendo, portanto, problemas de compatibilidade ou necessidade de reinstalação de algum *software* durante o projeto.

Como teste final, o sistema foi executado ao longo de três dias, durante aproximadamente oito horas diárias. Neste período, o programa teve seu banco de dados alimentado seis vezes e o servidor foi atualizado duas vezes, sem que houvesse necessidade de parada total do programa. Desta maneira, o projeto apresentou uma boa resistência às modificações e uma tolerância a erros expressiva.

A utilização de máquina virtual permitiu que o sistema fosse levado do computador de desenvolvimento ao computador de produção sem a necessidade de reinstalação do IIS e reconfiguração da conexão de VPN. Isto representou um grande ganho na velocidade de desenvolvimento e implantação. Mesmo assim, foi sentido uma perda de desempenho na execução dos *softwares* em comparação ao uso direto do sistema operacional.

CONSIDERAÇÕES FINAIS

Máquinas virtuais, atualmente, mostram-se viáveis em aplicações comerciais e acadêmicas. São confiáveis, flexíveis e apresentam desempenho bastante satisfatório.

Durante os testes, muitos usuários não foram capazes de diferenciar um sistema virtualizado (sendo executado em tela cheia) de um sistema real.

Em disciplinas como as citadas neste trabalho, as vantagens não terminam por aí: uma vez criado um ambiente padrão, ele pode ser replicado para diversos computadores, quantas vezes forem necessárias, principalmente no início de cada aula, restaurando as características originais do ambiente padrão. Isso dinamiza a manutenção dos sistemas operacionais.

Dada o isolamento entre os sistemas *host* e *guest*, o sistema base se mantém íntegro em caso de dano no sistema *guest*. Esse fato foi amplamente verificado no projeto “Terminal de Vendas de Arquivos Multimídia” o que torna sua aplicação em sala de aula uma prática viável e interessante.

Durante a execução do projeto, notou-se uma perda de performance, o que não foi significativo para a aplicação.

Uma possibilidade observada durante os testes é compartilhar um ambiente padronizado entre programadores, permitindo mobilidade no desenvolvimento do projeto.

AGRADECIMENTOS

Agradecemos aos alunos da sexta série da ênfase Computação da Engenharia Elétrica da Escola de Engenharia Mauá que participaram do projeto “Terminal de Vendas de Arquivos Multimídia” utilizando a máquina virtual e compartilhando duas experiências com os autores deste trabalho.

REFERÊNCIAS

- [1] POPEK, G. J., GOLDBERG, R. P. "Formal Requirements for Virtualizable Third Generation Architectures". *Communications of the ACM*, Vol 17, No 7., 1974, 412–421.
- [2] GOLDBERG, R. "Survey of Virtual Machine Research". *IEEE Computer Magazine*, Vol 7, 1974, 34 – 45.
- [3] GOLDBERG, R. MAGER, P. "Virtual Machine Technology: A bridge from Large Mainframes to Networks of Small Computers". *IEEE Proceedings Compton Fall 79*, 1979. 210 – 213.
- [4] SUGERMAN, J., GANESH, V., B. LIM. "Virtualizing I/O Devices on Vmware Workstation's Hosted Virtual Machine Monitor". *Proceedings of the 2001, USENIX Annual Technical Conference*. 1 – 14, 2001.
- [5] BELLARD, F. "QEMU Open source processor emulator". Disponível em: <<http://www.fabrice.bellard.free.fr/qemu>>. 2005. Acesso em 28 nov. 2007.
- [6] INNOTEK AG. "VirtualBox". 2007. Disponível em: <<http://www.virtualbox.org>>. Acesso em 28 nov. 2007.
- [7] VMWARE, INC. "VMware: Virtualization, Virtual Machine & Virtual Server Consolidation", 2007. Disponível em: <<http://www.vmware.com>>. Acesso em 28 nov. 2007.
- [8] OPENVPN. "OpenVPN - An open source SSL VPN Solution by James Yonan", 2006. Disponível em: <<http://openvpn.net>>. Acesso em 30 out. 2007.
- [9] ANDRADE, M. T. de. "Um estudo comparativo sobre as principais ferramentas de virtualização". 2006. 53p. Trabalho de Graduação – Universidade Federal de Pernambuco. Pernambuco. 2006.
- [10] LIMA, F. T., RODRIGUEZ, P. F. B., SABINO, R. D. "Terminal de Venda de Arquivos Multimídia", 2007. 50p. Trabalho de Graduação – Escola de Engenharia Mauá. São Caetano do Sul, 2007.