

VIRTUALIZAÇÃO DE AMBIENTES DE TRABALHO PARA LABORATÓRIOS DE ENSINO

Lucas F. Silva^{1*}, Rafael G. Trindade¹, João V. F. Lima¹, Rafael Boufleuer¹

1. Núcleo de Ciência da Computação da UFSM.

Resumo:

Este trabalho apresenta uma solução baseada na virtualização de ambientes de trabalho (máquinas virtuais) para laboratórios de ensino dos cursos de Informática atendidos pelo Núcleo de Ciência da Computação (NCC) da UFSM. Cada computador físico pode oferecer máquinas virtuais de diversos sistemas, possibilitando que cada disciplina ou atividade possa ter um ambiente personalizado e seguro. Uma interface desenvolvida pelo NCC permite aos usuários a escolha de qual sistema se deseja utilizar.

Palavras-chave: Virtualização; Laboratórios de ensino; Ambientes de trabalho.

Apoio financeiro: UFSM

Trabalho selecionado para a JNIC pela instituição: UFSM.

Introdução:

O uso de espaços como laboratórios de informática por escolas, universidades e outros ambientes de ensino é um fator importante no desenvolvimento de disciplinas que requeiram um mínimo de envolvimento tecnológico por parte dos alunos.

Diversos níveis de tarefas aplicáveis necessitam de um laboratório bem equipado, não apenas fisicamente mas também no âmbito de softwares. Sendo assim, faz-se necessária a constante atualização das ferramentas a serem disponibilizadas nesses ambientes, e para tal são necessárias instalações e configurações distintas dependendo da finalidade.

Entretanto, é importante frisar a necessidade de manter um sistema que atenda a demanda de uso dos laboratórios para atividades curriculares, variando os tipos de disciplinas e cursos envolvidos. Disciplinas diferentes necessitam de softwares, Sistemas Operacionais e especificações de hardware (memória, espaço em disco, etc.) diferentes, aumentando a complexidade de gerir um sistema único que atenda as necessidades para todas essas especificações.

Uma situação semelhante ocorre no Núcleo de Ciência da Computação (NCC), subunidade do Centro de Tecnologia da UFSM, responsável pelos laboratórios de ensino de dois cursos de graduação na área de Computação. Cada disciplina ou minicurso demanda requisitos específicos de software e sistema operacional, muitas vezes incompatíveis mutuamente. Além disso, a manutenção torna-se impraticável visto que cada alteração no sistema dos laboratórios deve ser propagada a dezenas de computadores, consumindo tempo e recursos.

No presente trabalho é apresentada uma solução baseada na virtualização de ambientes de trabalho (máquinas virtuais), onde cada computador físico pode oferecer diversas configurações de máquina e sistema, possibilitando que cada disciplina ou atividade

possa ter um ambiente personalizado.

Metodologia:

Os sistemas virtuais são disponibilizados no NCC da UFSM de acordo com a demanda dos laboratórios de ensino. Professores ou responsáveis pela demanda podem escolher o sistema operacional (Windows, Linux ou FreeBSD) e descrever os programas que precisam ser instalados no sistema. A virtualização é completa por meio de QEMU [1] e KVM [2] com extensões de hardware habilitadas.

Cada computador do laboratório pode rodar sistemas virtuais, sendo que as permissões dos usuários são distintas entre sistema físico e virtual. No sistema físico, os usuários não possuem permissões de administrador e a eles é permitido somente instanciar máquinas virtuais. O sistema virtual permite qualquer alteração pois a virtualização isola qualquer ação por parte dos usuários.

O computador cliente executa uma aplicação responsável por oferecer ao usuário uma interface para seleção e execução das máquinas virtuais disponíveis, implementado com a linguagem de programação Python.

O programa implementado permite instanciar uma máquina virtual e conectar-se a ela por meio do protocolo VNC (*Virtual Network Computing*) sem perda de desempenho. Caso o usuário feche o cliente VNC, o programa permite operações como reconectar ou desligar a máquina virtual através da biblioteca *libvirt* que oferece métodos em Python para gerenciar máquinas virtuais.

A imagem da máquina física consiste em um sistema Debian GNU/Linux com interface gráfica básica a fim de suportar máquinas virtuais. O computador possui usuário único chamado *ncc* sem quaisquer permissões administrativas, exceto por instanciar máquinas virtuais. Dessa forma, usuários dos laboratórios não podem alterar o sistema físico de cada computador. A imagem do sistema instalado não ultrapassa 1GB.

O espelhamento da imagem física é realizado por meio da ferramenta FOG [3]. Para clonar computadores, primeiramente precisa-se de um computador com o sistema básico instalado e então copia-se uma "imagem" de seu disco rígido local para um servidor na rede gerenciado pelo FOG. Dessa

forma, essa imagem pode ser espalhada em diversos computadores pela rede, criando máquinas "clones" da primeira máquina.

O sistema implementado compõe-se de um modelo cliente-servidor, onde o servidor disponibiliza imagens de máquinas virtuais aos computadores clientes através do protocolo NFS (Network File System) de compartilhamento de arquivos. A fim de economizar o espaço de armazenamento no servidor NFS, utilizou-se o formato de disco QCOW2 [4].

Visando um bom desempenho no âmbito de rede, utilizou-se também o FS-Cache [5] do NFS para a implantação de uma *cache* local de arquivos, que reduz os acessos à rede ao armazenar localmente os arquivos remotos em uso. Cada computador físico dedica uma partição física para *cache* (em geral a partição */dev/sda3* com sistema de arquivos Ext4).

Resultados e Discussão:

Como resultados têm-se o software em pleno uso nos laboratórios atendidos pelo NCC. Para o computador cliente foi desenvolvido, utilizando-se a linguagem de programação Python, uma aplicação responsável por oferecer ao usuário uma interface para seleção e execução das máquinas virtuais disponíveis.

O desempenho desse sistema virtualizado se mostrou bastante satisfatório, comprovando a eficiência do sistema de cache do NFS, pois o mesmo armazena o sistema virtual no computador cliente e assim evita acessos à rede.

Para a garantia da privacidade dos arquivos dos usuários do sistema, implementou-se também o armazenamento *copyonwrite* do QCOW2, que garante que as máquinas virtuais descartem modificações de seus usuários após o uso, ou seja, alterações no sistema são escritas em uma imagem local temporária que posteriormente é excluída.

Conclusões:

Este trabalho apresentou uma solução baseada na virtualização de ambientes de trabalho (máquinas virtuais) para laboratórios de ensino dos cursos de Informática atendidos

pelo NCC da UFSM. Com a adoção do sistema de máquinas virtuais, os principais resultados obtidos foram a redução na manutenção de software dos computadores de laboratório e flexibilização na implantação de software para disciplinas ou minicursos.

Como trabalhos futuros, pretende-se melhorar a interface com o usuário, assim como armazenar imagens de máquinas virtuais localmente sob demanda. A aplicação para os computadores clientes desenvolvida em Python é *Open Source* e encontra-se em <https://bitbucket.org/nccadm/gui-client>.

Referências bibliográficas

- [1] BELLARD, F. QEMU, a **Fast and Portable Dynamic Translator**. In: Proceedings of the Annual Conference on USENIX Annual Technical Conference. Berkeley, CA, USA: USENIX Association, 2005. (ATEC '05), p. 41–41. Disponível em: <<http://dl.acm.org/citation.cfm?id=1247360.1247401>>.
- [2] KIVITY, A. et al. **kvm: the linux virtual machine monitor**. In: Proceedings of the Linux symposium. [S.l.: s.n.], 2007. v. 1, p. 225–230.
- [3] TEAM, F. **The FOG Project**. 2016. Disponível em: <<https://fogproject.org/>>.
- [4] MCLOUGHLIN, M. **The QCOW2 Image Format**. 2008. Online.
- [5] HOWELLS, D. FS-Cache: **A Network Filesystem Caching Facility**. In: Proceedings of the Linux Symposium. [S.l.: s.n.], 2006. v. 1, p. 427–440.