

Virtualização de Imagens de Sistemas Operacionais em Laboratórios de Informática

Operating Systems Images Virtualization on Computer Laboratories

Lucas Ferreira da Silva¹, lferreira@inf.ufsm.br
Rafael Gauna Trindade²
Rafael Boufleuer³
João Vicente Ferreira Lima⁴

Universidade Federal de Santa Maria, Santa Maria, Rio Grande do Sul

Submetido em 28/08/2017

Revisado em 04/09/2017

Aprovado em 30/10/2017

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 deseja instanciar.

Palavras chave: Virtualização. Sistemas Operacionais. libvirt.

Abstract: This work presents a solution based on virtual machines for the computing laboratories managed by the NCC of UFSM. Each physical computer offers a set of virtual systems to the user; thus, each course has a safe and personalized virtual system. The NCC developed an interface that allows users to choose a virtual system to run.

Keywords: Virtualization. Operating Systems. libvirt.

¹ lattes.cnpq.br/6181181274261109

² lattes.cnpq.br/5002147023862451

³ lattes.cnpq.br/6323270665322650

⁴ lattes.cnpq.br/6266546896929217

Introdução

O Núcleo de Ciência da Computação (NCC) da Universidade Federal de Santa Maria (UFSM) é responsável pelos laboratórios de ensino que atendem aos cursos que realizam aulas no Centro de Tecnologia (CT) da UFSM e necessitam de um ambiente computadorizado. O papel desempenhado pelo NCC consiste em manter serviços básicos de rede, zelar pelo bom funcionamento dos laboratórios de ensino e atender as demandas de instalação de ferramentas de *software* para a realização de disciplinas em laboratório.

Diversos níveis de tarefas aplicáveis necessitam de um laboratório bem equipado, não apenas fisicamente mas também digitalmente, no âmbito de sistemas e aplicações para computadores. 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. Cada disciplina ou minicurso demanda requisitos específicos de *software* e sistema operacional, muitas vezes incompatíveis mutuamente. É importante salientar 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 programas, sistemas operacionais e especificações de hardware (memória, espaço mínimo em disco, poder de processamento, entre outros) diferentes, aumentando a complexidade de gerir um sistema único que atenda as necessidades para todas essas especificações.

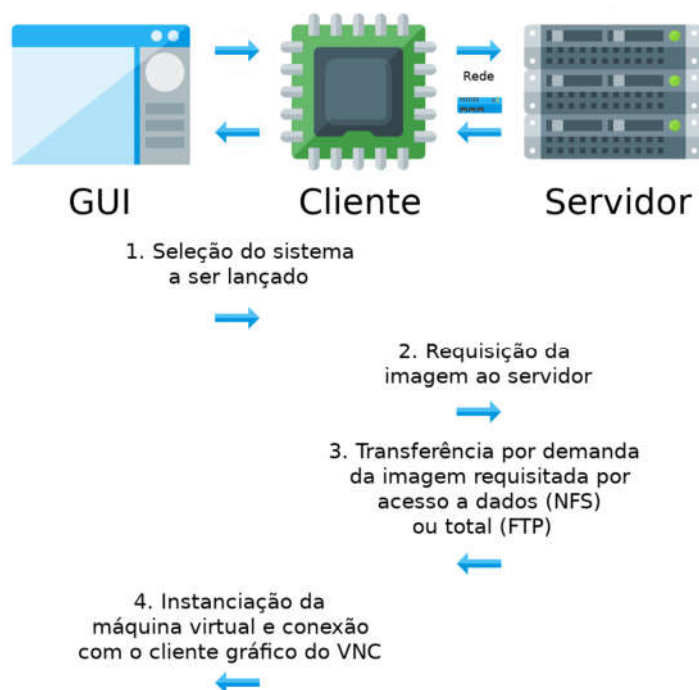
Nesse contexto, a manutenção dos laboratórios de ensino do NCC torna-se impraticável visto que cada alteração no sistema dos laboratórios deve ser propagada a dezenas de computadores, consumindo tempo e recursos. Historicamente, o NCC utilizou por muitos anos o sistema de boot remoto BpBatch⁵ para replicar imagens de disco (KREUTZ, 2003), isto é, cópias digitais de uma ou mais partições de uma unidade física de armazenamento de dados, no formato de um único arquivo que por vezes pode ser comprimido sem perdas, dependendo da ferramenta empregada para a criação da réplica, também chamada de clone. Devido à descontinuidade do BpBatch, passou-se a utilizar o

⁵ www.bpbatch.org

sistema FOG para replicar imagens de sistemas GNU/Linux e Windows. Porém, a utilização única e exclusiva do FOG (TRINDADE, 2015) apresentara diversas desvantagens como dificuldade de manter constantes alterações do sistema por demanda dos cursos, tamanho do arquivo clone final do sistema e problemas no momento de replicar esses arquivos em determinados computadores com defeitos específicos de *hardware* que impossibilitavam o processo.

Este trabalho apresenta uma solução baseada na virtualização de ambientes de trabalho (máquinas virtuais) para laboratórios de ensino dos cursos de Informática atendidos pelo 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 gráfica desenvolvida pelo NCC permite aos usuários a escolha de qual sistema deseja utilizar. Uma versão resumida de todo o processo de seleção e transferência pode ser visualizada no infográfico apresentado na Figura 1. O restante do artigo explica por partes os detalhes do desenvolvimento da ferramenta, começando por uma revisão bibliográfica dos trabalhos relacionados, logo após explicando alguns conceitos por trás de virtualização de sistemas, definição de imagem de disco de máquina física e como a ferramenta FOG foi usada para solucionar o problema de clonagem de discos, um detalhamento do compartilhamento e sistema de dependência das imagens utilizadas pela ferramenta desenvolvida, especificação da interface de usuário criada e da versão que faz uso do protocolo FTP ao invés do sistema NFS. Por fim alguns resultados decorrentes do período de uso da solução são apresentados, e uma conclusão é formulada sobre sua eficiência em relação ao problema proposto.

Figura 1 — Infográfico com uma versão abstraída dos passos de seleção, requisição, descarga e instanciação de uma máquina virtual utilizando-se o sistema.



Trabalhos Relacionados

Dentre outras soluções de gerenciamento de imagens de máquinas virtuais, o Oracle VM Virtualbox é uma programa multiplataforma de virtualização que permite a criação, configuração e instanciação de imagens de sistemas operacionais virtualizados (Oracle Corporation, 2004). Apesar de propor uma aplicação bem similar à busca pelo NCC, seu uso fora descartado devido ao excesso de recursos que sua interface gráfica dispõe, e que preferencialmente devem ser abstraídos dos usuários finais a fim de manter a consistência das imagens originais. Além disso, para cada imagem nova adicionada, uma clonagem completa com uma nova configuração deveria ser realizada em todos os computadores do laboratório para que o Virtualbox, hospedado nas máquinas clientes, reconhecesse a nova opção.

Alguns trabalhos que abordam o uso de um sistema de gerenciamento de múltiplas instâncias de máquinas virtuais podem ser classificados como correlatos. Rossi (2011) apresenta uma experiência acadêmica do uso de uma infraestrutura virtualizada em laboratórios de computação, visando o ensino

prático sobre a interconexão de dispositivos e sistemas computacionais. Dispondo de um laboratório com 20 computadores, fez uso do monitor de máquinas virtuais Xen e o software para roteamento Zebra. Com a criação de quatro máquinas virtuais por computador, possibilitou aos seus usuários a formação de uma rede virtualizada, a qual funciona de forma isolada da rede física, possibilitando a criação de configurações sem riscos de prejudicar a estrutura física e as configurações de rede de sua instituição.

Ao explorar a ferramenta XenDesktop, Aguiar (2013) propõe a adoção da virtualização de *desktops* e da computação em nuvem dentro de instituições de ensino, com objetivo de reduzir custos e esforços com equipes de manutenção de laboratórios. Em seu estudo, ele exhibe os resultados de um questionário contendo cinco questões direcionadas para levantar informações acerca da realidade encontrada pelas universidades, submetido a gestores de TI de diferentes universidades. As informações levantadas através do questionário revelam um panorama favorável a utilização de tecnologias de virtualização. A ferramenta XenDesktop permite virtualizar estações de trabalho, passando a oferecer seus aplicativos como um serviço móvel, acessível através de diversos dispositivos, incluindo clientes magros (*Thin Clients*), disponibilizando seu acesso tanto através de uma rede de computadores local, como pela internet, usando redes de dados móveis. Sua eficiência, no entanto, baseia-se na existência de uma ou mais máquinas servidoras capazes de lidar com a carga de trabalho de inúmeras instâncias concorrentes de máquinas virtuais que executam aplicações pesadas com uso de, na maioria dos casos, processamento gráfico, ignorando dessa forma boa parte do poder de processamento disponível nas máquinas clientes. O autor sugere o uso compartilhado de uma mesma instância por diversos usuários, entretanto a solução ainda tende a apresentar problemas de escalabilidade se o número de clientes for grande, como é o caso típico de laboratórios de computação, onde, tomando o NCC como exemplo, é comum encontrar em torno de 30 máquinas em uso simultaneamente por laboratório.

Buscando um panorama similar ao dos clientes magros que fizesse proveito das capacidades de *hardware* dos computadores clientes e diminuísse a carga com processamento nos servidores, Moreira (2004) descreve a

abordagem de Terminais Inteligentes (*smart clients*) como uma solução estratégica visando a otimização de recursos computacionais. Terminais Inteligentes possuem razoável capacidade de processamento e de memória RAM locais. Moreira classifica como desperdício de recursos utilizar máquinas servidoras para realizar tarefas que podem rodar em estações clientes. Um *script* de instalação denominado RBL fora desenvolvido para realizar o uso sistemático de aplicações locais permitindo o pleno controle de balanceamento de carga na rede local. A estrutura cliente/servidor foi a base para a implementação empreendida, tendo os servidores provendo serviços como distribuição de IP, transferência de arquivo de um *kernel* para inicialização do sistema, exportação do sistema de arquivos pela rede e serviços de autenticação. Com a disponibilização desses serviços criou-se um servidor que atendesse um sistema computacional relativamente robusto e eficiente para a época. Através dos serviços disponibilizados pelo servidor, cada estação cliente adquire sua própria configuração mantida no servidor para leitura e escrita. O servidor disponibiliza um sistema GNU/Linux básico, usando apenas alguns arquivos compartilhados. Foi criada a possibilidade de balanceamento de carga em ambientes com algumas aplicações rodando no servidor e outras nos clientes, configurados pela instalação de aplicativos em diretórios compartilhados ou não, respectivamente. O sistema, entretanto, assim como um cliente magro, apresenta-se como totalmente dependente da conexão constante com o servidor. Problemas na rede podem interromper o desenvolvimento de uma atividade, podendo inclusive levar a sua perda completa caso uma cópia de segurança não tenha sido feita.

Sistema de Virtualização

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 ou GNU/Linux) e listar os *softwares* que precisam ser instalados. A virtualização é completa por meio do QEMU (*Quick Emulator*) (BELLARD, 2005) e KVM (*Kernel-based Virtual Machine*) (KIVITY, 2007) com extensões de hardware habilitadas.

KVM é um conjunto de módulos do kernel Linux que permitem a um processo em espaço de usuário executar código em um modo de processo

especial. Esse modo permite que um programa em espaço de usuário capture eventos interessantes como operações de entrada e saída. Já o QEMU usa as interfaces providas pelo KVM para implementar a virtualização completa do sistema, emulando *hardware* comum como um disco de armazenamento, placa gráfica, *slots* de um barramento PCI (*Peripheral Component Interconnect*), etc. No caso do KVM, qualquer solicitação de entrada e saída que um sistema operacional convidado faz é interceptada e encaminhada para o modo de usuário a ser emulado pelo processo QEMU (JONES, 2007).

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 do sistema físico qualquer ação por parte dos usuários, preservando sua integridade para usos futuros sem uma recorrente carência de manutenção.

Imagem da Máquina Física

A imagem da máquina física consiste em um sistema Debian GNU/Linux de configuração básica a fim de suportar máquinas virtuais e prover um ambiente de trabalho que permita a execução de um sistema operacional virtualizado com servidor gráfico. O computador possui um usuário único chamado *ncc* sem quaisquer permissões administrativas, exceto a possibilidade de instanciar máquinas virtuais. Dessa forma, usuários dos laboratórios não podem alterar as configurações do sistema operacional físico de cada computador. A imagem desse sistema instalado não ultrapassa 1 GB.

O espelhamento da imagem física é realizado por meio da ferramenta FOG. 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 armazenamento em um servidor na rede gerenciado pelo FOG. Dessa forma, essa imagem pode ser espelhada em diversos computadores pela rede e cadastrados no FOG, criando máquinas “clones” da máquina inicial (TRINDADE, 2015). A Figura 2 apresenta uma fotografia da etapa de espelhamento da imagem clonada em computadores de um dos laboratórios do NCC. O processo é demorado, chegando a levar mais de 2 horas para um

laboratório inteiro, e depende exclusivamente da velocidade e disponibilidade da rede.

Figura 2 — Processo de *download* de uma imagem de disco à computadores de um laboratório do NCC através da ferramenta FOG.



Compartilhamento de Imagens

O sistema implementado possui uma arquitetura cliente-servidor, onde o servidor disponibiliza imagens de máquinas virtuais aos computadores clientes através do protocolo NFS (*Network File System*) de compartilhamento de arquivos. Optou-se por essa estratégia devido aos seguintes fatores:

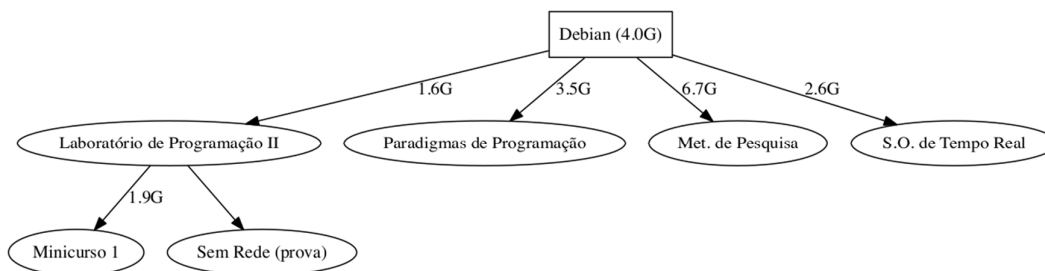
- Atualizações em imagens virtuais não demandam alterações nas máquinas físicas dos laboratórios, pois estão residentes no servidor;
- Os computadores dos laboratórios possuem discos rígidos de tamanhos distintos e, portanto, nem todos poderiam armazenar todas as imagens virtuais necessárias. Além disso, o crescimento na quantidade de imagens disponíveis afeta conseqüentemente o tempo total de clonagem com a ferramenta FOG devido ao crescimento de dados em disco necessários para serem copiados e replicados.

A fim de economizar o espaço de armazenamento no servidor NFS, utilizou-se o formato de disco QCOW2 (MCLOUGHLIN, 2008). As principais características que justificaram sua escolha foram:

- O arquivo de imagem cresce a medida que os dados são armazenados. Dessa forma, é possível criar discos virtuais que consomem somente o necessário em espaço físico;
- Suporte a cópia em escrita (*copy-on-write*), o que permite criar uma imagem base e utilizá-la sem alterações. Dessa forma, pode-se criar diferentes configurações de um mesmo sistema sem alterar o original e descartar alterações quando necessário.

A Figura 3 ilustra por meio de uma árvore a relação entre uma imagem virtual de um sistema operacional base Debian e suas imagens derivadas. A diferença de tamanho em disco entre a imagem base e as derivadas é indicada pelas setas contínuas e o espaço adicional gerado pelas alterações. Setas sem valor (imagem virtual “Sem rede” da Figura 2) indicam alterações de configuração que não modificam o sistema tais como remoção de interfaces de rede, aumento de memória, etc. Exceto a imagem base, as outras imagens virtuais correspondem a uma disciplina ou demanda de uso dos laboratórios de ensino. A principal desvantagem do formato QCOW2 é falta de suporte a montagem direta, ou seja, é necessário um utilitário do QEMU para acessar os arquivos da imagem.

Figura 3 — Imagem raiz (retângulo) e derivadas para cada disciplina (elipse).



O desempenho no acesso das imagens virtuais é garantido graças à um sistema de cache do NFS a fim de evitar gargalos de rede. O FS-Cache (HOWELLS, 2006) reduz os acessos à rede ao armazenar localmente os arquivos remotos em uso. Cada computador físico dedica uma partição física para cache, formatada preferencialmente para Ext4, um sistema de arquivos com uso de *journaling*.

Não obstante, as máquinas virtuais descartam modificações de seus usuários através do armazenamento *copy-on-write* do QCOW2, ou seja, alterações no sistema são escritas em uma imagem local temporária. Essa imagem local com alterações dos usuários é descartada sempre que a máquina física é reiniciada ou desligada. Dessa forma, os usuários podem instalar novos programas ou alterar o sistema conforme suas necessidades em uma atividade didática de forma isolada e sem prejudicar à máquina física. Em casos em que se é necessário persistir de alguma maneira os documentos criados durante o período de uso da imagem pelo aluno, sugere-se que os arquivos sejam armazenados em algum serviço hospedado na internet, ou, se especificado pelo professor que realizou a demanda da imagem, é possível utilizar uma imagem criada que possui suporte a acesso de arquivos pessoais armazenados em um servidor NFS do NCC, além de possuir um sistema de autenticação para o usuário usando o protocolo LDAP (*Lightweight Directory Access Protocol*), protocolo que também controla o sistema de autenticação do serviço de acesso remoto por meio de um servidor SSH (*Secure Shell*), serviço ofertado pelo NCC aos alunos dos cursos de Sistemas de Informação e Ciência da Computação e do curso de pós-graduação em Informática da UFSM. Ainda não há suporte a dispositivos conectáveis por porta USB (*Universal Serial Bus*), devido a alguns empecilhos com os módulos KVM.

Interface com o Usuário

O computador cliente, por sua vez, executa uma aplicação responsável por oferecer ao usuário uma interface gráfica para seleção e execução das máquinas virtuais disponíveis, implementada com a linguagem de programação Python e a biblioteca de interface gráfica GTK+. A Figura 4 ilustra a interface do programa cliente implementado para uso nos laboratórios. Ele 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 para gerenciar máquinas virtuais.

O design da interface é fortemente influenciado pela interface gráfica do programa Virtualbox, que dispõe uma lista de sistemas ao usuário juntamente com um ícone referente a distribuição instalada na imagem selecionada e uma captura de tela da mesma. Essa lista de sistemas é gerada a partir da interpretação de um arquivo no formato JSON, disponibilizado no servidor juntamente com as imagens, que contém metadados das imagens disponíveis. Tal formato permite uma fácil atualização da lista por parte da equipe de manutenção devido à sua simplicidade, além de permitir uma fácil integração com outros sistemas para sua edição, como, por exemplo, uma interface web. Um exemplo de configuração no formato proposto para exibição de uma máquina virtual na lista pode ser conferido na Figura 5, que apresenta a imagem criada para uma disciplina do curso de Ciência da Computação.

Figura 4 — Interface gráfica do seletor de máquinas virtuais.

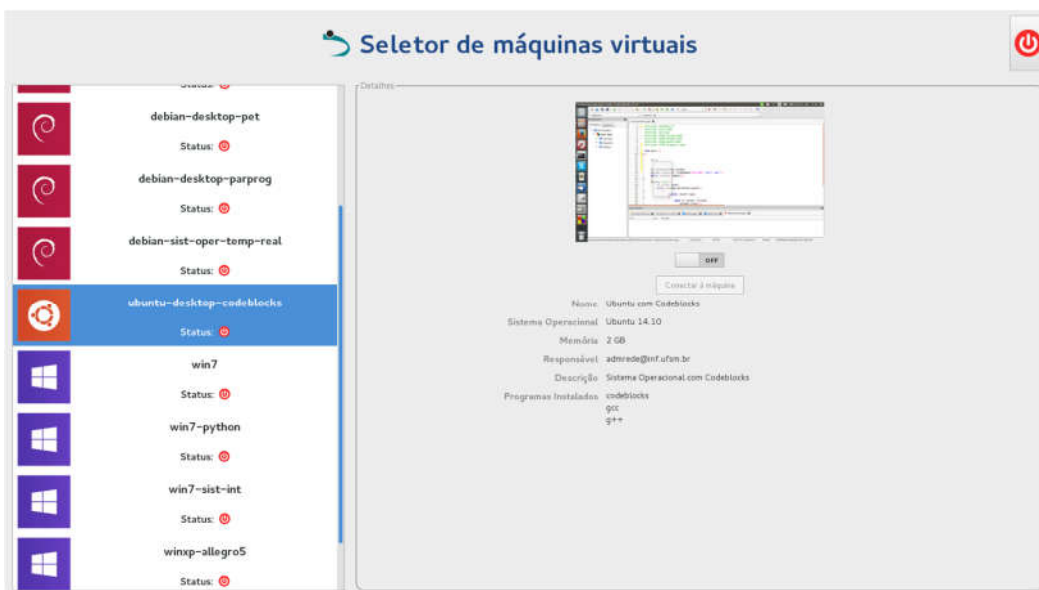


Figura 5 — Entrada JSON para uma máquina virtual utilizada nos laboratórios

```

"progpar":{
  "nome":"Debian para Paradigmas da Programação",
  "id":"debian-desktop-parprog",
  "id_pai":"debian-desktop",
  "distro":"debian",
  "so":"Debian 8 Jessie Stable",
  "memoria":"2 GB",
  "responsavel":"admrede@inf.ufsm.br",
  "descricao":"Sistema Operacional com ferramentas\nespecificas para a disciplina\nde Paradigmas da Programação",
  "ss":"screenshots/parprog.png",
  "ocultar":"false",
  "programas":["gcc", "g++", "ghc", "swi-prolog", "jdk 8", "netbeans", "git"]
},

```

A interpretação deste arquivo JSON também permite aos mantenedores do sistema filtrarem quais imagens dentre as existentes serão disponibilizadas para *download* aos alunos, possibilitando, por exemplo, não permitir que determinadas imagens base sejam exibidas, pois geralmente não possuem nenhuma ferramenta instalada além de pacotes de *softwares* mínimos para o funcionamento de um sistema operacional com ambiente gráfico.

Implementação utilizando o protocolo FTP

Além da implementação utilizando o protocolo NFS, posteriormente também fora desenvolvida uma versão do sistema no qual o *download* é realizado utilizando o protocolo FTP (*File Transfer Protocol*) e as imagens são transferidas sob demanda. Essa implementação alternativa foi motivada pelos fatores velocidade da rede e presença de conexão entre os computadores cliente e servidor, onde as imagens estão armazenadas, pois a mesma não pode ser garantida, podendo acarretar no mau funcionamento durante o uso do sistema ou até mesmo na impossibilidade de lançamento do mesmo, caso não exista nenhuma conexão com o servidor NFS que o disponibiliza.

Dados esses possíveis problemas relacionados à rede, a versão com suporte a FTP utiliza-se do conceito de uma *cache* local para o armazenamento das imagens conforme as mesmas são utilizadas, garantindo assim que o sistema funcione sem o acesso a rede, pois a imagem que se deseja executar não precisa ser novamente baixada do servidor caso a mesma já tenha sido utilizada anteriormente, isso porque ela já estará armazenada na *cache* local em decorrência de uma transferência prévia.

Para otimizar o *download* das imagens, tirou-se proveito de seu sistema de hierarquia provido pelo formato QCOW2, sendo que quando se deseja utilizar uma imagem que possua dependência em outras, ou seja, derivada de outras

imagens, somente serão baixadas as dependências da escolhida, além dela própria. Se as dependências também forem bases de outras imagens que foram previamente transferidas para o computador cliente em questão, não haverá a necessidade de novos *downloads* redundantes, podendo conseqüentemente serem reaproveitadas para demais imagens derivadas que eventualmente venham a ser requisitadas para o servidor posteriormente.

Dadas as mudanças da forma de como é feito a descarga das imagens por meio do protocolo FTP, foi adicionado um novo atributo nos objetos do arquivo JSON, responsável por indicar o arquivo de imagem considerado como nó pai na árvore de dependências, que pode ser vista na Figura 3. Esse novo campo passou a ser necessário porque, diferente da implementação com NFS que transfere conforme o uso todas as imagens necessárias ao mesmo tempo, a implementação com FTP realiza o *download* de cada arquivo necessário para que a demanda seja atendida, buscando no arquivo JSON o identificador da imagem e sua dependência, sanando as dependências por meio do *download* recursivo de cada imagem até que a raiz da árvore de dependências (que não é dependente de nenhuma outra) seja baixada.

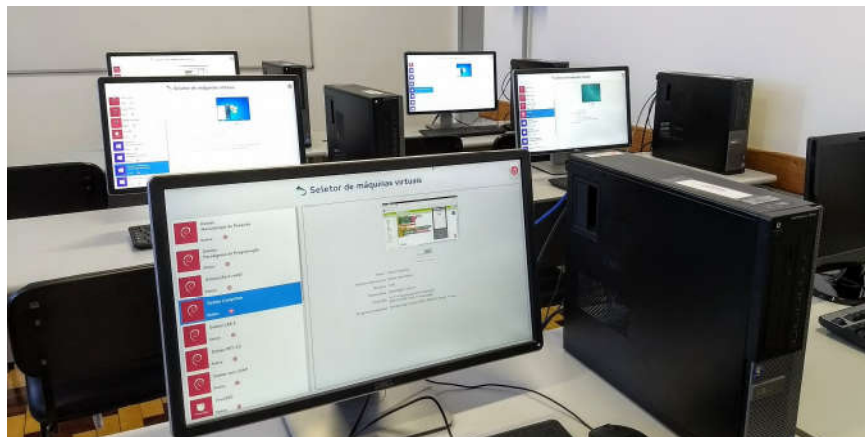
Como essa implementação alternativa apresenta-se como uma opção para laboratórios com problemas de velocidade de conexão, o *download* das imagens pode apresentar lentidão dependendo da rede, tangendo tempos na escala de alguns minutos conforme o laboratório. Desta forma, torna-se necessário um *feedback* do tempo a ser esperado pelo usuário, o que é feito através da exibição de uma barra de progresso na interface do sistema, informando o andamento do *download* de todos os arquivos necessários para o lançamento do sistema operacional escolhido.

Resultados

O sistema fora implantado nos laboratórios de computação (Figura 6) pertencentes ao NCC no início do primeiro semestre letivo de 2016, e desde então sua aplicabilidade pode ser comprovada na prática. Tarefas paulatinas como a replicação de imagens, comuns em inícios de semestres devido a demanda relativamente constante por instalações de software foram praticamente extinguidas, sendo substituídas pelo processo de carregar a

imagem virtual original do sistema desejado contida no servidor e modificá-la ou criar uma extensão QCOW2 dela com a modificação, de forma a não alterar a original.

Figura 6 — Aplicativo seletor de máquinas virtuais disponíveis aos usuários.



Em questão de tempo de transferência, nota-se que a múltipla carga simultânea das imagens sobre a rede pode trazer algum atraso em termos de alguns segundos. Porém, essa carga é reduzida uma vez que não é necessário o *download* de toda a imagem, e sim somente dos setores necessários para o funcionamento da mesma. O sistema de *cache* local permite a reutilização de dados já transferidos, eliminando comunicações repetitivas.

Apresentando-se também como um fator considerável, os computadores existentes nos laboratórios do NCC estão ligeiramente obsoletos para alguns sistemas, como as versões mais recentes do Ubuntu GNU/Linux, por exemplo, o qual possui um ambiente de área de trabalho que consome muitos recursos e processamento, apesar de mais recentemente ter sido descontinuado oficialmente⁶. Esse fator conta para um desempenho mais modesto do ambiente virtualizado como um todo.

⁶ arstechnica.com/information-technology/2017/04/ubuntu-unity-is-dead-desktop-will-switch-back-to-gnome-next-year/

Conclusão

Este trabalho apresentou uma solução baseada na virtualização de ambientes de trabalho (máquinas virtuais) para laboratórios de ensino dos disciplinas de computação 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. O código fonte da interface descrita encontra-se em um repositório online⁷ com acesso aberto.

Referências

- BELLARD, F. **QEMU, a Fast and Portable Dynamic Translator**. in: Proceedings of the Annual Conference on USENIX Annual Technical Conference, ATEC '05, Berkeley, CA, USA. USENIX Association, p. 41, 2005.
- HOWELLS, D. **FS-Cache: A Network Filesystem Caching Facility**. in: Proceedings of the Linux Symposium, v. 1, p. 427-440, 2006.
- KIVITY, A., KAMAY, Y., LAOR, D., LUBLIN, U., LIGUORI, A. **kvm: the linux virtual machine monitor**. In Proceedings of the Linux symposium, v. 1, p. 225-230, 2007.
- KREUTZ, D. L. **Sistema de boot remoto: Instalação, configuração e geração de imagens de sistemas operacionais**. Relatório Técnico, UFSM. Santa Maria: 2003.
- ROSSI, F. D., BRANDOLT, J. F., DA ROZA, M. P., DA ROZA, J. C., WAGNER, R. **Infraestrutura virtualizada em laboratórios de computação: novas perspectivas de expansão para o ensino**. in: XVII Congreso Argentino de Ciencias de la Computación. La Plata, Argentina, p. 548-556, out. 2011.
- AGUIAR, L. G. F., FRUFREK, G. L. **Aplicação de Virtualização de Desktops e Cloud Computing no contexto acadêmico minimizando custos e esforços com equipes de manutenção**. Online, 2013. Disponível em <<http://www.academia.edu/4819068>>.
- MOREIRA, R. J., COSTA, H. L. A., JUNIOR, J. C. DE SOUZA, G. M. T., LEITÃO, U. A. **Terminais Inteligentes: alternativa estratégica para otimização de recursos computacionais**. in: 5º Workshop sobre Software Livre - WSL 2004, p. 81-84. Porto Alegre: jun. 2004.
- TRINDADE, R. G., CHARÃO, A. S., LIMA, J. V. F. **Automatização de Instalações de Software e Sistemas Operacionais em Laboratórios de Ensino Usando a**

⁷ bitbucket.org/nccadm/gui-client

Ferramenta FOG. in: XXX Jornada Acadêmica Integrada da UFSM. Santa Maria: out. 2015.

MCCLOUGHLIN, M. **The QCOW2 Image Format.** Online, set. 2008. Disponível em <<https://people.gnome.org/~markmc/qcow-image-format.html>>.

JONES, M. **Descubra a Máquina Virtual do Kernel Linux: Conheça a Arquitetura e Vantagens da KVM.** Online, abr. 2007. Disponível em <<https://www.ibm.com/developerworks/br/library/l-linux-kvm/>>.

Fog Team. **The FOG Project.** Online, 2016. Disponível em <<https://fogproject.org>>.

Oracle Corporation. **Oracle VM Virtualbox: User Manual.** Online, 2004. Disponível em <<https://www.virtualbox.org/manual/>>.