

Módulo de Descoberta Automática de Monitores de Máquinas Virtuais Xen

Guilherme Piêgas Koslovski¹, Márcio Parise Bouffleur¹,
Andrea Schwertner Charão¹
{guilherm, bouffleur, andrea}@inf.ufsm.br

¹Laboratório de Sistemas de Computação - LSC
Universidade Federal de Santa Maria - UFSM
Santa Maria – RS – Brasil

Abstract. *Virtualization technologies have recently gained visibility as a solution to take advantage of modern architectures. Virtual machine monitors, like Xen, allow simultaneous and independent execution of several operating systems over the same physical computer, sharing underlying resources. The performance and functionality of Xen virtual machine monitor motivate its use in many distributed environments. This work describes the design and implementation of a software module which enables discovering hosts running Xen in a local network. Such module is useful to system managers that need to administrate multiple hosts in virtualized environments.*

Resumo. *Ambientes virtualizados têm sido foco de diversas pesquisas atuais e constituem uma solução vantajosa para o bom aproveitamento de arquiteturas modernas. Monitores de máquinas virtuais, tais como Xen, permitem a execução simultânea e independente de diversos sistemas operacionais sobre um mesmo computador real, compartilhando os recursos existentes. As vantagens obtidas com a virtualização baseada no monitor Xen impulsionam sua utilização em diversos ambientes distribuídos. Este trabalho descreve a implementação de um módulo que permite a descoberta de computadores com o monitor Xen ativo em redes locais. Tal módulo é útil para administradores de sistemas que precisam gerenciar múltiplos computadores em ambientes virtualizados.*

1. Introdução

A virtualização de recursos computacionais é uma tecnologia que possibilita abstrair um *hardware* real através de sua representação utilizando máquinas virtuais. Este trabalho explora ambientes em que a virtualização ocorre em nível de sistemas operacionais, isto é, em que diversos sistemas podem ser executados como máquinas virtuais sobre um equipamento real, compartilhando seus recursos de *hardware* e *software*. Esta abordagem foi inicialmente difundida na década de 60 para permitir o compartilhamento do poder computacional dos *mainframes* IBM [Goldberg 1974], e atualmente tornou-se foco de diversas pesquisas, buscando explorar de forma mais eficiente os recursos dos computadores modernos.

Para permitir o compartilhamento do *hardware* entre diferentes máquinas virtuais, existe uma camada de *software* responsável pelo gerenciamento e escalonamento

do acesso aos recursos, denominada Monitor de Máquinas Virtuais (MMV). Dentre os MMV's existentes, Xen [Barham et al. 2003] é um monitor de código aberto que destaca-se por oferecer suporte a diversas famílias de processadores e possibilitar a migração de máquinas virtuais entre computadores interconectados.

Atualmente, as vantagens obtidas pela utilização de ambientes virtualizados vêm sendo exploradas em diversos trabalhos nas áreas de redes e sistemas distribuídos, auxiliando em pontos como administração de *clusters* de computadores, consolidação de servidores e balanceamento de carga entre computadores interconectados [Huang et al. 2006, Figueiredo et al. 2003, Stahl 2005]. Neste tipo de ambiente, é importante conhecer-se a identificação dos múltiplos computadores executando máquinas virtuais, sendo que esta informação é trabalhosa de se obter manualmente. Com base nisso, este trabalho visa o desenvolvimento de um módulo para descoberta de computadores com o MMV Xen em execução¹, buscando auxiliar o administrador destes ambientes em suas tarefas de gerenciamento.

O restante deste trabalho está organizado da seguinte forma: na seção 2 descreve-se as principais características de monitores de máquinas virtuais, destacando-se o MMV Xen. Na seção 3 são apresentadas as tarefas básicas necessárias para o gerenciamento de um ambiente virtualizado baseado em Xen. Já a seção 4 apresenta a proposta para o descobridor de computadores hospedeiros, descrevendo seu projeto e implementação. Na seção 5 traçam-se as considerações finais sobre o presente trabalho.

2. Monitor de Máquinas Virtuais Xen

Monitores de Máquinas Virtuais caracterizam-se por ser uma camada de *software* que permite a execução simultânea de diversos sistemas operacionais sobre uma mesma arquitetura real. Este sistema é responsável pelo gerenciamento das máquinas virtuais que fornecem uma abstração do *hardware* existente. Através do MMV, os sistemas em execução podem ter acesso a um conjunto de recursos de forma eficiente e semelhante à arquitetura original do computador.

O MMV também garante que cada sistema operacional execute independentemente dos demais sobre cada máquina virtual. Para possibilitar este isolamento, monitores devem possuir o maior nível de privilégios de execução em um computador, de forma que seja possível interceptar e gerenciar o acesso aos recursos de *hardware*.

O MMV Xen é um monitor de código aberto, desenvolvido pela Universidade de Cambridge, que atualmente oferece suporte para diversas famílias de processadores como IA-32, AMD64 e EM64T. Xen permite a utilização dos sistemas operacionais Linux e NetBSD como sistemas hospedeiros, e Linux, NetBSD, FreeBSD e Windows XP como sistemas hospedados.

Na arquitetura de Xen, utiliza-se uma nomenclatura para distinguir as máquinas virtuais (*domains*) do monitor propriamente dito (*hypervisor*). Os domínios (*domains*) representam os sistemas operacionais hospedados, sendo que existe um domínio administrativo (*Domain0*) que fornece as ferramentas necessárias para a manipulação do monitor e dos demais sistemas operacionais.

¹Código fonte disponível em <http://www.inf.ufsm.br/~bouffleur/XenDiscover.tar.gz>

O isolamento e a independência de *hardware* alcançada com a execução sobre uma máquina virtual permite o processo de encapsulamento e migração de um sistema operacional entre computadores interconectados. Xen implementa um mecanismo denominado *live migration* [Clark et al. 2005], que permite a migração de uma máquina virtual sem interromper a execução dos serviços.

3. Gerenciamento de Ambientes Virtualizados

Ambientes virtualizados caracterizam-se basicamente pela existência de um conjunto de computadores interligados, utilizando uma solução de virtualização, como o MMV Xen, por exemplo. Normalmente, estes hospedeiros executam um ou mais sistemas operacionais distintos, que compartilham os recursos da arquitetura.

Nestes ambientes, é comum a execução, por parte do administrador, de tarefas de gerenciamento de máquinas virtuais como: criação, destruição, desativação e migração. Estas tarefas podem ser executadas local e remotamente nos *hosts* hospedeiros [Inc. 2006, RedHat 2006].

Nativamente, Xen oferece uma interface administrativa local para manipulação de máquinas virtuais, através do utilitário *xm*. Para manipulação remota, faz-se necessário a conexão individual em cada *host* utilizando um usuário com privilégios de super-usuário, através de SSH (*Secure SHell*) por exemplo. Este processo deve ser realizado para execução de qualquer procedimento em máquinas virtuais alocadas remotamente, exigindo do administrador um conhecimento antecipado dos *hostnames* ou endereços IP dos computadores hospedeiros.

Particularmente em ambientes virtualizados com um alto número de computadores hospedeiros, existe uma dificuldade de identificação dos computadores que estão ativos em um determinado momento. Tal fato serviu de motivação para o desenvolvimento de uma solução para descoberta automática de computadores virtualizados.

4. Módulo de Descoberta de Computadores Virtualizados

4.1. Visão Geral

Este módulo busca auxiliar administradores de ambientes virtualizados a obter uma visão dos computadores hospedeiros que estão ativos em um determinado momento, auxiliando na tarefa de conexão remota para realização de procedimentos administrativos. Na concepção deste módulo, buscou-se projetá-lo de forma que se possa utilizá-lo como uma ferramenta independente ou como um pacote auxiliar para o desenvolvimento de ferramentas administrativas.

A utilização básica do módulo desenvolvido se dá via linha de comando, informando-se a porta utilizada no ambiente para conexões com o MMV Xen. Esta porta é estabelecida no momento da instalação e configuração dos computadores virtualizados e deve ser conhecida para realizar-se outras tarefas administrativas como a migração de máquinas virtuais ou a utilização de uma API para gerenciamento remoto [Mellor et al. 2006]. A saída do comando de descoberta é uma lista de computadores que estão com o MMV Xen ativo no ambiente em questão.

4.2. Projeto e Implementação

A figura 1 apresenta um diagrama UML simplificado, onde identifica-se as quatro classes que compõem o módulo. A classe *XenDiscover* é instanciada no início do processo de descoberta, com o número da porta usada para conexão com Xen.

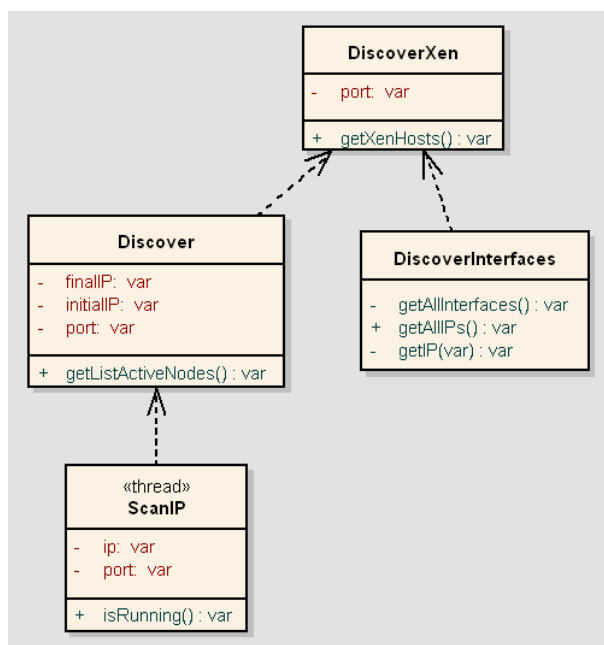


Figura 1. Diagrama UML simplificado do módulo de descoberta de *hosts*

Após a inicialização de *XenDiscover*, a seqüência de passos de execução para a descoberta é a seguinte:

1. Descobre-se quais interfaces de rede estão ativas no computador utilizado para gerenciamento
2. Identifica-se os endereços IP's atribuídos a cada interface
3. Inicializa-se a busca individual nas interfaces identificadas:
 - Percorre-se todos os endereços IP's possíveis para a classe
 - Cria-se uma *thread* independente para cada tentativa de conexão no endereço identificado

A invocação do método principal `getXenHosts()` retorna para o administrador uma lista contendo os *hostnames* dos hospedeiros identificados. Este recurso, além de ser utilizado individualmente, pode ser utilizado para o desenvolvimento de ferramentas administrativas que ofereçam uma interface para administração de ambientes virtualizados [RedHat 2006, Hergert 2005, Inc. 2006].

A figura 2 apresenta o diagrama de seqüência dos passos necessários para a descoberta dos *hosts*. Observa-se inicialmente a invocação do método público `getActiveNodes` em *DiscoverXen*. Este método aciona o mecanismo de procura, invocando os demais métodos existentes no módulo. Particularmente, observa-se a necessidade de utilização de *threads* na implementação da classe *ScanIP*, para permitir a execução simultânea de tentativas de conexão à hospedeiros.

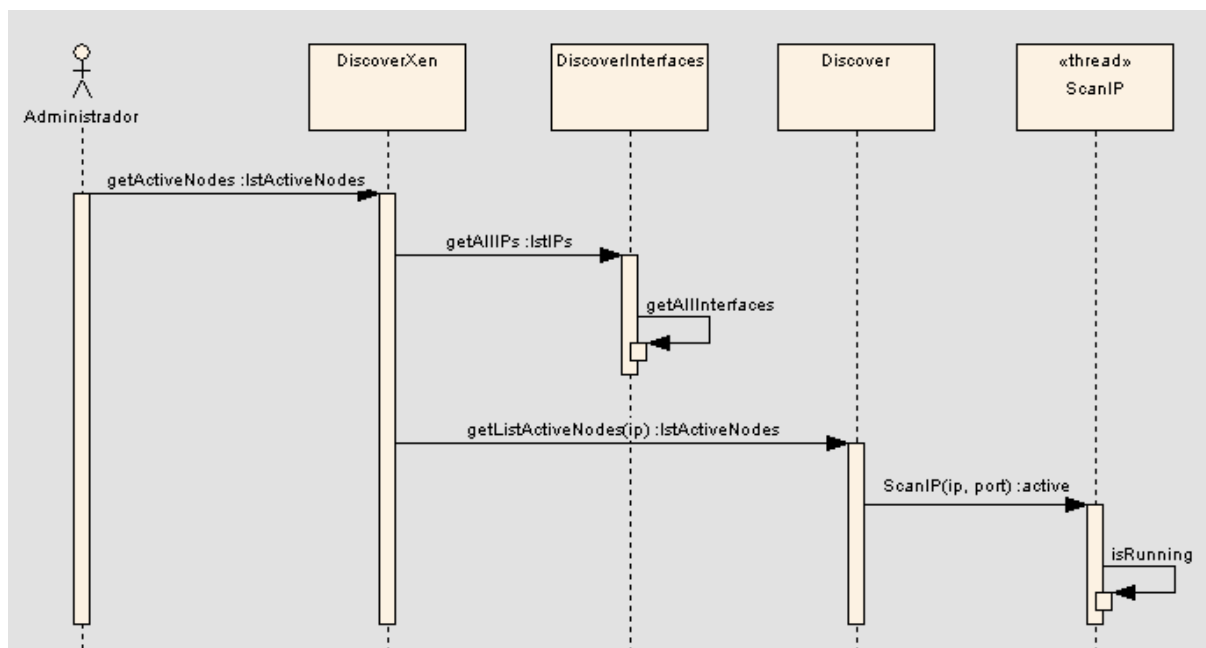


Figura 2. Diagrama de seqüência do módulo de descoberta de *hosts*

Para a implementação deste módulo utilizou-se a linguagem de programação Python, em sua versão 2.5, em conjunto com bibliotecas para manipulação de *sockets* e *threads*. A escolha da linguagem justifica-se por manter uma compatibilidade com a documentação do monitor Xen [Mellor et al. 2006], que atualmente oferece suporte para as linguagens C e Python.

4.3. Utilização

O módulo desenvolvido pode ser usado via linha de comando. Ao acionar o mecanismo, deve-se informar a porta utilizada pelo MMV Xen para comunicação externa. A figura 3 fornece um exemplo de uso do módulo, em que obtém-se como saída uma lista de 3 *hosts* com o MMV Xen ativo.

```

$ ./DiscoverXen.py 8002
['polar.inf.ufsm.br', 'carbono12.inf.ufsm.br', 'capivara.inf.ufsm.br']
  
```

Figura 3. Exemplo da utilização do módulo de descoberta

5. Conclusão

Este trabalho apresentou uma contribuição na área de gerenciamento de ambientes virtualizados baseados em Xen. Para manipular máquinas virtuais através da interface nativa de Xen, administradores de ambientes virtualizados precisam identificar os computadores que estão ativos em um determinado momento antes de realizar a conexão remota. Para facilitar esta tarefa, desenvolveu-se uma ferramenta para descoberta de computadores hospedeiros com o MMV Xen em execução. Apresentou-se o processo de implementação

desta ferramenta, descrevendo-se sua forma de utilização como módulo independente ou como pacote auxiliar para o desenvolvimento de ferramentas administrativas.

Como trabalhos futuros, pretende-se integrar o módulo desenvolvido a uma ferramenta de gerenciamento de ambientes virtualizados que está em atualmente em desenvolvimento, sob responsabilidade dos mesmos autores deste artigo. O módulo de descoberta também pode ser aprimorado, incluindo-se por exemplo a capacidade de descoberta em redes com múltiplas sub-redes. Para isso, o mecanismo de descoberta precisa ser modificado para atravessar *gateways* conhecidos.

Referências

- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003). Xen and the art of virtualization. In *Proc. 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, pages 164–177, Bolton Landing, USA. ACM.
- Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., and Warfield, A. (2005). Live migration of virtual machines. In *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI '05)*, Boston, USA. Usenix.
- Figueiredo, R., Dinda, P., and Fortes, J. (2003). A case for grid computing on virtual machines. In *Proc. International Conference on Distributed Computing Systems (ICDCS)*.
- Goldberg, R. (1974). Survey of virtual machine research. *IEEE Computer*, 7(6):34–45.
- Hergert, C. (2005). Xen Manager. Disponível em: <http://xenmanager.dronestudios.com/cgi-bin/trac.cgi>.
- Huang, W., Liu, J., Abali, B., and Panda, D. (2006). A case for high performance computing with virtual machines. *The 20th ACM International Conference on Supercomputing*.
- Inc., X. S. (2006). Xen enterprise data sheet v11022006, version 3.1. Technical Report 02-11-2006, Xen Source Inc.
- Mellor, E., Sharp, R., Scott, D., and Harrop, J. (2006). Xen management API draft - revision 0.4.3. Technical Report 25-08-06, XenSource Inc.
- RedHat (2006). Virt-manager virtual machine manager. Disponível em: <http://virt-manager.et.redhat.com/>.
- Stahl, E. (2005). Virtualization security and integrity in the IBM @server POWER5 environment. Technical Report 090105, IBM Systems and Technology Group.