

Apache2 e MySQL 5 sobre Máquinas Virtuais Xen e KVM

Lucas Mendes Ribeiro Arbiza¹, Diego Luís Kreutz¹

¹Grupo de Pesquisa em Sistemas de Informação — GPSI
Núcleo de Tecnologia da Informação e Comunicação — NTIC
Universidade Federal do Pampa — UNIPAMPA

{lucas,diego}@ntic.unipampa.edu.br

Resumo. *O objetivo deste trabalho é avaliar o desempenho de máquinas virtuais Xen e KVM para serviços comuns em redes, como HTTP (Apache) e banco de dados (MySQL). Para tanto, foram utilizadas máquinas virtuais dos dois virtualizadores e diferentes ferramentas de medição do Apache2 e do MySQL 5. Os experimentos foram realizados em cenários convencionais, com instalações típicas das máquinas virtuais Xen e KVM e dos dois serviços, cuja finalidade foi retratar os casos mais comuns de instalação e uso desses servidores virtualizados.*

1. Introdução

A virtualização de sistemas tem despontado como uma das novas grandes tendências da computação. Alguns dos principais fatores que tem levado a esse novo impulso na área de virtualização são as ferramentas e tecnologias existentes hoje no mercado, capazes de proporcionar otimização no uso de recursos computacionais, isolamento, segurança e novas possibilidades administrativas. Algumas pesquisas e aplicações no âmbito dessa nova tendência de virtualização de sistemas são apresentadas e comentadas em [Kreutz 2007].

No cenário de redes, duas soluções de virtualização de sistemas tem chamado cada vez mais a atenção dos administradores de sistemas, o Xen e o KVM. Eles representam duas soluções de monitores de máquinas virtuais (MMVs). O Xen é baseado em paravirtualização, enquanto que o KVM em virtualização nativa, dependendo de suporte a virtualização em nível de hardware. Ambos vem chamando a atenção de administradores de redes e estão disponíveis para uso na maioria das distribuições GNU/Linux.

O Xen, entre administradores de redes, é considerado um dos mais eficientes e práticos monitores de máquinas virtuais. Ele também é uma das soluções de virtualização mais difundidas e utilizadas nesse meio. O KVM, por outro lado, ainda é pouco difundido e utilizado na prática por esses usuários. Neste contexto, um dos objetivos deste trabalho é apresentar um pouco mais sobre esses dois monitores de máquinas virtuais. Além disso, outro objetivo é apresentar resultados práticos de avaliação de máquinas virtuais Xen e KVM com serviços bastante utilizados em praticamente todos os domínios de rede, HTTP e banco de dados. No caso, considerando ambientes GNU/Linux, as avaliações de desempenho das máquinas virtuais foram realizadas com o Apache2 e o MySQL 5. Os resultados apresentam um comparativo estatístico entre os dois serviços sobre os dois tipos de máquinas virtuais.

A seção 2 apresenta uma descrição das soluções de paravirtualização Xen e KVM. A seção 3 discrimina os serviços Apache2 e MySQL 5, avaliados sobre as máquinas virtuais. A seção 4 apresenta as ferramentas de benchmark utilizadas nos testes e os resultados obtidos. A seção 5 apresenta as principais inferências do trabalho.

2. Virtualização com Xen e KVM

2.1. Xen

O Xen [Barham et al. 2003a, Dong 2006] é um *hypervisor* que roda na camada de maior privilégio entre o hardware e uma ou mais máquinas virtuais por ele hospedadas e gerenciadas [Campbell and Jeronimo 2006]. Com esta técnica — paravirtualização — o desempenho do sistema virtualizado é alto e a execução é segura devido ao isolamento entre os sistemas hospedados (*guests*). Contudo, para contornar as adversidades impostas pela paravirtualização, especialmente em relação a sistemas operacionais privados, empresas como Intel e AMD tem investido no desenvolvimento de tecnologias de virtualização em hardware. Além dos investimentos em tecnologias de hardware, devido às funcionalidades e às atrativas características do Xen, inovações tem surgido visando facilitar a utilização e o gerenciamento de máquinas virtuais.

A paravirtualização cria uma abstração de hardware para cada máquina virtual muito similar ao hardware existente, diferente da virtualização total (*full-virtualization*), onde um hardware virtual completo é criado, emulado. As máquinas virtuais que executam sob o comando de um *hypervisor* paravirtualizado compartilham os recursos de hardware através de APIs, que implicam na modificação do kernel do sistema operacional, provendo níveis de desempenho similares aos que seriam obtidos rodando o sistema diretamente sobre o hardware. É importante notar que a paravirtualização não envolve custos computacionais tão honerosos quanto a virtualização baseada em emulação.

A figura 1 mostra o *hypervisor* do Xen como uma camada de abstração de hardware que possibilita o acesso de diversas máquinas virtuais aos recursos físicos do hardware. O gerenciamento das máquinas virtuais é realizado pelo domínio 0 (Domain 0), em verde escuro na figura. O domínio 0 também é uma máquina virtual, porém com privilégios de administração. Através do domínio 0 pode-se criar, destruir, gerenciar máquinas virtuais, programar parâmetros, alocar memória física e permitir acesso aos dispositivos de armazenamento e rede [Barham et al. 2003b]. Este domínio é sempre inicializado quando a máquina real (hardware) é inicializada.

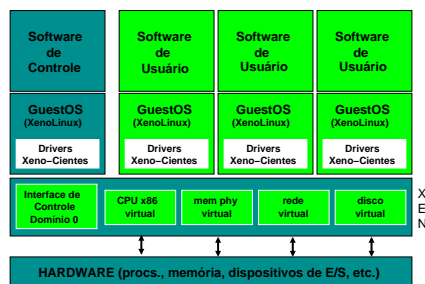


Figura 1. Arquitetura do Xen

2.2. KVM (Kernel-based Virtual Machine)

O KVM [Qumranet 2009, RedHat Emerging Technology Project 2009, Jones 2007] é uma infraestrutura de software no kernel do Linux para suporte à virtualização. Em linhas gerais, o KVM é uma combinação entre um driver de virtualização Linux em modo kernel e um programa em nível de usuário (Qemu) modificado, responsável pela interface com o driver.

Uma das peculiaridades do KVM sobre muitas das outras soluções de virtualização é o fato de ele ser capaz de tirar vantagem dos recursos de virtualização providos pelo hardware dos processadores mais modernos. Entre essas tecnologias podem ser citadas a Intel-VT [Neiger et al. 2006] e a ADM-V [Crosby and Brown 2006]. Por outro lado, um aspecto negativo é o fato de o KVM não ter suporte para execução de máquinas virtuais em sistemas desprovidos de suporte a virtualização em nível de hardware. Nessa categoria enquadra-se grande parte do hardware atualmente disponível e utilizado pelos usuários finais.

Um Linux convencional possui apenas dois modos: kernel e usuário. O KVM, como ilustrado na figura 2, adiciona um terceiro modo, denominado *guest*. O modo *guest* possui o seu próprio kernel com modo kernel e usuário.

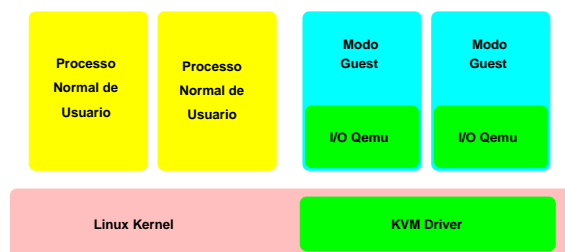


Figura 2. Arquitetura do KVM

A divisão de trabalho entre os diferentes modos pode ser caracterizada da seguinte maneira [Qumranet 2009]: 1) o **modo guest** executa códigos guest que não sejam operações de entrada e saída (I/O); 2) o **modo kernel** muda para o modo guest e gerencia quaisquer retornos do modo guest devido à operações de entrada e saída ou instruções especiais; e 3) **modo usuário** executa operações de entrada e saída para o guest.

Através da integração ao kernel do Linux, o hypervisor KVM permite explorar os mais recentes recursos de hardware e características de escalabilidade com muito pouco esforço adicional. Um dos fatores que levou a integração do KVM ao kernel do Linux, em detrimento ao Xen, é o fato dele ser bastante simples, possuindo apenas dois componentes em sua estrutura, o que diminui de forma significativa os potenciais impactos negativos na manutenção, segurança e atualização de um sistema operacional.

Os componentes do KVM são [Qumranet 2009]: 1) um driver de dispositivo para o gerenciamento do hardware virtualizado, onde o driver exporta as suas capacidades através de um dispositivo de caractere identificado como `/dev/kvm`; e 2) um componente em nível de usuário para emulação de hardware, através de um processo qemu levemente modificado. Este processo modificado mapeia a memória física do guest e chama o modo kernel para executar o modo guest. Já o modelo de entrada e saída é derivado diretamente do qemu [Bellard 2005].

O KVM suporta uma variedade grande de sistemas, incluindo Linux, Windows, BSD, Solaris, ReactOS, entre outros. Isso deve-se ao fato de o KVM fazer uso de recursos de virtualização em hardware, suportando teoricamente qualquer sistema operacional.

3. Ferramentas e Ambiente de Testes

3.1. Apache2 e MySQL 5

O Apache [The Apache Software Foundation 2009] e o MySQL [Sun Microsystems, Inc. 2009] são serviços tipicamente utilizados na grande maioria dos domínios de rede. Além disso, são serviços que, na média, demandam uma maior quantidade de recursos computacionais, diferentemente de outros serviços básicos, como DNS, DHCP e SMTP. É comum o tráfego e a intensidade das demandas serem maiores sobre serviços HTTP e banco de dados.

O Apache é derivado de um projeto que teve por objetivo criar e manter um servidor HTTP de código aberto. Hoje ele é um dos servidores Web mais robustos e utilizados no mundo inteiro, sendo suportado e mantido para plataformas Unix, Linux e Windows.

O MySQL é um dos mais populares e utilizados sistemas de gerenciamento de banco de dados. Assim como o Apache, é um projeto de código aberto e distribuição livre. A quantidade de ferramentas e linguagens com suporte para MySQL, disponíveis no mercado, é bastante grande. As primeiras experiências de novos programadores e administradores de sistemas geralmente acabam sendo com MySQL devido a sua popularidade, simplicidade, documentação e suporte.

Neste trabalho, foram escolhidos para avaliação das máquinas virtuais Xen e KVM os serviços Apache2 e MySQL 5. Ambos foram instalados com configurações padrão, iguais em todas as máquinas virtuais.

3.2. Ferramentas de benchmark

No caso do Apache2, foram utilizadas as ferramentas de benchmark `httpperf` e `ab` (Apache Benchmark). Adicionalmente, foi criado um script para forçar testes que envolvessem muitas operações de entrada e saída (I/O) nas máquinas virtuais.

O `httpperf` [Hewlett-Packard Development Company, L.P. 2009] é uma ferramenta de teste de desempenho em servidores Web cujo objetivo é prover uma ferramenta robusta, de alto desempenho, que facilite a construção de micro e macro benchmarks. O `ab` [die.net 2009] é um utilitário do próprio Apache2. Esta ferramenta tem por objetivo testar a instalação do Apache, principalmente no quesito de quantas requisições por segundo o sistema é capaz de atender.

A avaliação dos servidores MySQL em máquinas virtuais foi realizada com a suíte de benchmarks do MySQL-server [Sun Microsystems, Inc. 2009]. Foram considerados os resultados dos testes com 7 módulos da suíte, que compreendem operações de alteração de tabelas, inserção e remoção de tabelas e registros, consultas, atualização de informações e testes de conexão.

3.3. O ambiente de testes

O ambiente físico utilizado para os testes foi constituído de 16 PCs e 1 switch. **Características dos PCs:** Processador Intel(R) Core(TM) 2 Quad de 2.4 GHz, HD SATA de 360 GB 7200 RPM, 4 GB de memória RAM DDR2 800 MHz (2 x 2 GB), placa de vídeo nVidia Corporation GeForce 8400 GS 128 bits de 512 MB, memória cache de 4Mb. **Características do Switch Fast Ethernet:** D-link Xtark com 50 portas - 48 portas Fast

Ethernet 100 Mbps e 2 portas combo para fibra-ótica. **Sistema instalado nas máquinas:** Ubuntu Server 9.04 AMD 64 bits. **Versão do Xen:** xen-3.3 com kernel Linux 2.6.24-24-xen. **Versão do KVM:** kernel Linux 2.6.28-14-generic, libvirt 0.6.1, qemu 0.10.0.

No caso do Xen, foram utilizados discos virtuais em arquivos, com sistema de arquivos ext3. No caso do KVM, foram utilizados discos virtuais Qemu (qcow2), com sistema de arquivos ext3. Tanto os discos virtuais do Xen quanto do KVM são comuns e, em especial, fáceis de migrar de um servidor para outro, uma característica muita vezes desejável em sistemas virtualizados, dando maior liberdade e agilidade aos administradores de rede.

Como sistema foi utilizado o Ubuntu Server 9.04. As máquinas virtuais Xen e KVM foram configuradas com os discos virtuais e 128 MB de memória.

4. Estatísticas de desempenho

4.1. Apache2

Os números apresentados nesta seção representam a média de 20 execuções. Os testes foram repetidos para as máquinas virtuais Xen e KVM, nas mesmas condições de ambiente e configuração.

O script, utilizando o comando `wget`, realiza o download de 10.240 arquivos entre 1 a 256 KB, totalizando 533 MB de dados. No caso das máquinas virtuais Xen, o tempo médio de download de todos os arquivos ficou em 2 minutos e 3 segundos. Já no caso das máquinas virtuais KVM, o tempo médio ficou em 2 minutos e 6 segundos. Como a diferença ficou em apenas 3 segundos, concluí-se que o desempenho de ambos é bastante similar para servidores Apache2 com grande quantidade de arquivos pequenos disponíveis para download.

O comando `ab` utilizado nos testes foi `"ab -k -n 10000 -c 2"`. Isso significa que cada teste foi composto de 10000 requisições, com 2 requisições simultâneas. A tabela 1 apresenta os resultados médios dos testes realizados. Como pode ser observado, o Xen teve um desempenho ligeiramente superior ao KVM. Contudo, essa pequena diferença de desempenho dificilmente seria perceptível aos usuários finais do sistema Web.

MMV/métrica	taxa (Kbps)	requisições por segundo	tempo por requisição (ms)
Xen (ms)	11477,341	44,783	22,33
KVM (ms)	11445,385	44,655	22,39

Tabela 1. Medidas de desempenho do `ab`

No caso do `httperf`, foi executado o seguinte comando: `"httperf -timeout=2 -client=0/1 -server=IP -port=80 -uri=/ -rate=10 -send-buffer=1024 -recv-buffer=4096 -num-conns=100 -num-calls=2"`, com taxa de criação das seções igual a 10, buffer de envio de 1024 bytes, buffer de recepção de 4096 bytes, número de conexões igual a 100 e uma seção com sequência de *burst* igual a 2. A tabela 2 apresenta os resultados dos testes. Como pode ser observado, o Xen apresenta medidas de melhor valor, com excessão do desvio padrão, sendo mais eficiente que o KVM.

O tempo de teste do `httperf`, tanto para o Xen quanto para o KVM, ficou em 9,902 segundos. Ao total, foram 100 conexões, 200 requisições e 200 respostas.

MMV/métrica	mínimo	média	máximo	mediana	desvio padrão
Xen (ms)	1,6	1,9	14,5	1,5	1,3
KVM (ms)	2,25	2,55	13,5	2,5	1,1

Tabela 2. Medidas de desempenho do httpperf

Os dados apresentados demonstram que na maioria dos casos, de máquinas virtuais com Apache2, o Xen terá um desempenho ligeiramente superior ao KVM. Por outro lado, o Xen implica em um kernel modificado, enquanto que o KVM suporta qualquer kernel, tanto de sistemas livres quanto comerciais.

4.2. MySQL 5.0

O benchmark foi executado 8 vezes para cada máquina virtual, Xen e KVM. Em ambas as máquinas virtuais estava instalado o `mysql-server-5.0`.

Para a realização dos testes foram utilizados 8 computadores, sendo 4 hospedeiros de máquinas virtuais com MySQL (2 Xen e 2 KVM) e outros 4 computadores utilizados para executar o benchmark contra as máquinas virtuais.

Cada execução do benchmark em servidores Xen foi realizada, na média, em 695,25 segundos, enquanto que a média de cada execução em servidores KVM foi de 936,19 segundos. Isso representa uma diferença de tempo de 25,6% na execução de serviços MySQL que geram grandes cargas de dados e ou operações.

Os dados dos gráficos das figuras 3, 4 e 5, representando o uso da CPU em relação ao tipo de operação, também demonstram a melhor eficiência do Xen. Com isso, conclui-se que, em termos de eficiência para servidores virtuais Apache2 e MySQL, o Xen ainda é a melhor opção. Mas, a diferença, em alguns casos, é bastante pequena, podendo ser mais interessante o uso do KVM, pois este não implica em modificações no kernel dos sistemas operacionais. Sendo assim, o KVM permitirá também a execução simultânea de outros sistemas operacionais, sejam eles livres ou comerciais, sobre um mesmo hardware.

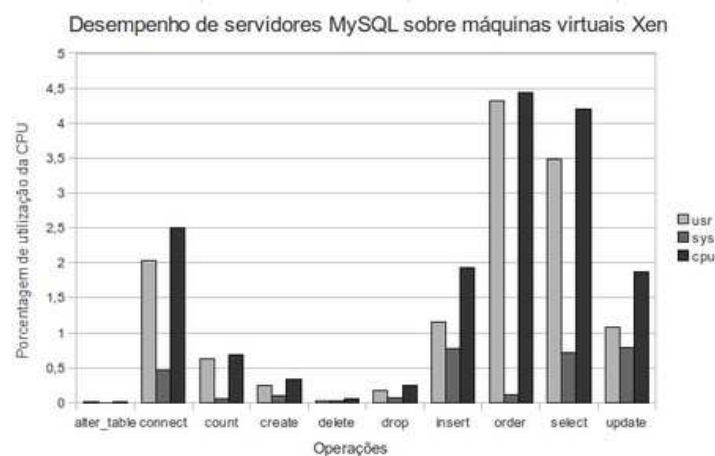


Figura 3. Utilização de CPU do Xen

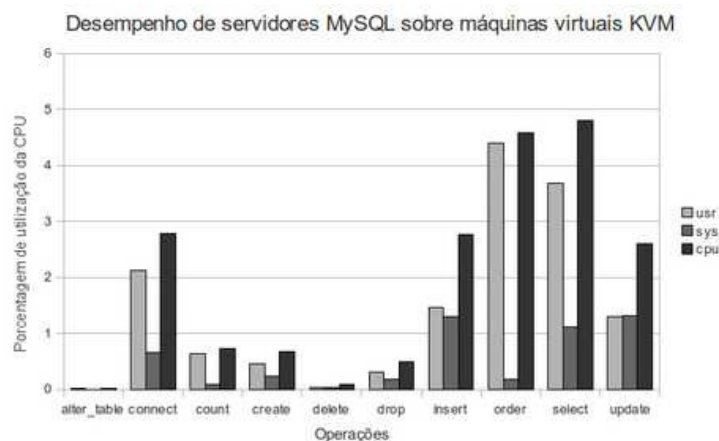


Figura 4. Utilização de CPU do KVM

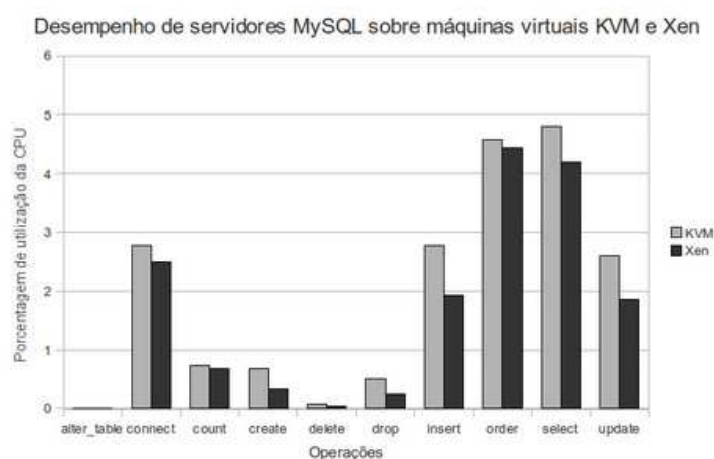


Figura 5. Comparativo entre o Xen e o KVM

5. Considerações Finais

A virtualização tem despontado como uma das grandes novas tendências na área da computação. Uma das formas de constatar-se essa realidade é através da observação do crescente uso de monitores de máquinas virtuais, como o Xen e o KVM, objetos de avaliação deste trabalho.

Os resultados dos benchmarks executados sobre as máquinas virtuais Apache2 e MySQL demonstraram que, na média, o desempenho do Xen ainda é superior ao do KVM. Isso deve-se principalmente ao fato de o KVM depender da arquitetura de emulação do Qemu, enquanto que o Xen utiliza paravirtualização, tendo um overhead médio menor no acesso e gerenciamento dos recursos de hardware.

Por outro lado, o KVM pode ser a melhor opção para ambientes mistos, onde diferentes sistemas operacionais e ou kernels serão utilizados, incluindo Linux, Windows, BSD, Solaris, ReactOS, entre outros. Como o Xen depende de kernels específicos, modificados, ele seria inviável em contextos desse gênero. Além disso, o KVM é mais simples, logo, menos propenso a problemas de segurança, mais fácil de ser mantido e menos intrusivo ao núcleo dos sistemas operacionais.

Referências

- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003a). Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA. ACM Press.
- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003b). Xen and the art of virtualization. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, volume 37, 5 of *Operating Systems Review*, pages 164–177, New York. ACM Press.
- Bellard, F. (2005). QEMU, a fast and portable dynamic translator. In *USENIX Annual Technical Conference, FREENIX Track*, pages 41–46. USENIX.
- Campbell, S. and Jeronimo, M. (2006). *An Introduction to Virtualization*. Intel Press. http://www.intel.com/intelpress/sum_vp10.htm.
- Crosby, S. and Brown, D. (2006). The virtualization reality. *ACM Queue*, 4(10):34–41.
- die.net (2009). ab - apache http server benchmarking tool. <http://linux.die.net/man/1/ab>.
- Dong, Y. (2006). Xen and Intel virtualization technology for IA-64. In Anonymous, editor, *Proceedings of Gelato ICE: Itanium Conference and Expo: Spotlighting Linux on Itanium-based Platforms, October 1-4, 2006, Biopolis, Singapore*, pages ??–??.
- Hewlett-Packard Development Company, L.P. (2009). httpperf. <http://www.hp1.hp.com/research/linux/httpperf/>.
- Jones, M. T. (2007). Discover the linux kernel virtual machine. <http://www.ibm.com/developerworks/linux/library/l-linux-kvm/>.
- Kreutz, D. (2007). Conceitos e aplicações dos monitores de máquinas virtuais. Technical report, Programa de Pós Graduação em Informática (PPGI), Universidade Federal de Santa Maria (UFSM), Santa Maria, RS.
- Neiger, G., Santoni, A., Leung, F., Rodgers, D., and Uhlig, R. (2006). Intel Virtualization Technology: Hardware support for efficient processor virtualization. *Intel Technology Journal*, 10(3):167–177.
- Qumranet (2009). Kvm: Kernel-based virtualization machine. www.qumranet.com/files/Qumranet_WhitePaper_03.pdf.
- RedHat Emerging Technology Project (2009). Kernel based virtual machine. http://www.linux-kvm.org/page/Main_Page.
- Sun Microsystems, Inc. (2009). Mysql. <http://www.mysql.com/>.
- The Apache Software Foundation (2009). Apache http server project. <http://httpd.apache.org/>.