

Análise Comparativa de Desempenho do Hipervisor Xen: Paravirtualização versus Virtualização Total

Ronaldo Canofre Mariano dos Santos, Andrea Schwertner Charão

Laboratório de Sistemas de Computação (LSC)

Mestrado em Ciência da Computação – Universidade Federal de Santa Maria (UFSM)

Campus UFSM - 971405-900 – Santa Maria – RS – Brasil

{canofre, andrea}@inf.ufsm.br

Resumo

Tecnologias de virtualização têm sido amplamente utilizadas em sistemas computacionais, tanto em nível acadêmico quanto comercial. O desenvolvimento de novos processadores que implementam o suporte à virtualização ampliou ainda mais este cenário. Dentre as tecnologias de virtualização atuais, o hipervisor Xen está entre os mais populares e destaca-se por oferecer duas abordagens distintas: a virtualização total assistida por hardware e a paravirtualização. Este artigo apresenta uma comparação de desempenho entre essas abordagens, utilizando Xen e analisando características como acesso a disco, memória, comunicação de rede e utilização do processador. O resultado deste trabalho mostra que a utilização do Xen paravirtualizado apresenta um melhor desempenho face ao seu suporte à virtualização total.

1 Introdução

Nos últimos anos a virtualização tem sido um tema presente nas diversas áreas da computação. Sua capacidade de executar vários sistemas operacionais de forma simultânea sobre um mesmo *hardware* possibilita um uso mais eficiente dos recursos disponíveis, uma melhor gerenciabilidade do ambiente, compatibilidade de *software*, flexibilidade e mais segurança devido ao isolamento de falhas, dentre outras vantagens.

Para o uso da virtualização, diversos métodos e ferramentas têm sido propostas, analisadas e utilizadas, fazendo com que a escolha de uma destas soluções não seja trivial. Dentre as ferramentas existentes, soluções proprietárias como VMWare [19], ou *open source* como Xen [22], OpenVZ [17] e KVM [12], vêm ganhando uma atenção considerável e têm sido amplamente utilizadas e comparadas atualmente [11, 13], sendo também alvos de

análises de desempenho [18].

A ferramenta Xen é uma das soluções mais populares e oferece duas abordagens distintas de virtualização, que exigem modificações no núcleo do sistema hospedado (paravirtualização) ou recriam em *software* toda a estrutura de *hardware* (virtualização total, do inglês *full virtualization*). Ambas as formas de implementação inserem impactos distintos no desempenho de operações básicas do sistema, como acesso a disco, memória, comunicação de rede e utilização do processador (CPU).

Com o desenvolvimento de extensões para arquitetura x86 pela AMD e Intel, culminando com a inclusão de novas tecnologias (Pacifica e Vanderpoll respectivamente) em seus processadores, tornou-se possível realizar a virtualização total sem a necessidade de replicar todos os componentes físicos da máquina real. Isso ocorre porque estes processadores oferecem suporte nativo a virtualização, permitindo que sistemas hospedados sejam executados como processos, diretamente sobre o *hardware*.

Neste trabalho, tem-se como objetivo quantificar o impacto da paravirtualização e da virtualização total, com suporte assistido por *hardware*, sobre o desempenho de um sistema utilizando Xen. Para isso, utilizam-se *benchmarks* para avaliar o desempenho de rede, processador, disco e memória em uma arquitetura com tecnologia Intel VT.

O restante deste artigo está organizado como segue: na Seção 2, expõe-se os conceitos gerais sobre virtualização e as tecnologias existentes. Na Seção 3, abordam-se as principais características do hipervisor Xen e as formas como ele implementa o acesso aos dispositivos físicos. Na Seção 4, apresenta-se a metodologia e as ferramentas utilizadas. Na Seção 5 são apresentados e discutidos os resultados obtidos, constituindo a principal contribuição deste trabalho. Por fim, na Seção 6, são apresentadas as considerações finais sobre o trabalho.

2 Tecnologias de Virtualização

No final dos anos 50 e início dos 60 se estabelecia um novo conceito de utilização de máquinas de grande porte, com o emprego de uma técnica que possibilitava o compartilhamento de um mesmo *hardware* por diversas Máquinas Virtuais (MV).

De modo geral, a técnica de virtualização consiste na inserção de uma camada extra de *software* denominada Monitor de Máquinas Virtuais (MMV) ou hipervisor, entre o sistema físico e o sistema operacional, podendo ou não contar com um suporte a virtualização em *hardware*.

Assim, torna-se possível a execução de diversos sistemas operacionais de forma independente, cujos acessos aos dispositivos físicos são controlados por esta camada extra de *software*, proporcionando características como confiabilidade, flexibilidade, gerenciabilidade e um bom isolamento de falhas [5].

Para realizar a execução de um ou vários SO's sobre um mesmo *hardware*, o MMV necessita utilizar-se de alguma técnica que permita a ele fornecer ao SO convidado as características e funcionalidades necessárias para que o mesmo execute [8]. Nas seções a seguir são abordadas duas implementações diferentes de monitores e o suporte oferecido pelos processadores Intel e AMD.

2.1 Paravirtualização

Em arquiteturas semelhantes à IA-32, o MMV não pode interceptar algumas instruções geradas pela MV, permitindo assim que esta execute com um maior nível de privilégio. Dessa forma, visando possibilitar a virtualização nessas arquiteturas e buscando um melhor desempenho na execução dos sistemas convidados [4], propôs-se o uso da paravirtualização.

Nesta técnica são realizadas algumas modificações no núcleo dos SO's convidados e hospedeiro, para que possam executar em paralelo contornando estas limitações [23]. As modificações realizadas no hipervisor resultam em uma interface de *hardware* virtual similar ao *hardware* real, possibilitando uma melhor interação entre MV e MMV, resultando em um menor impacto no desempenho. O principal exemplo de utilização é a ferramenta Xen.

2.2 Virtualização Total

Alguns sistemas operacionais não permitem as alterações necessárias para rodar em um ambiente paravirtualizado, sendo preciso executarem sobre uma réplica fiel da arquitetura para qual foram projetados.

Para tais sistemas, pode-se optar pela técnica de virtualização total, onde o hipervisor replica virtualmente toda a arquitetura necessária. No entanto a recriação de

uma cópia idêntica do sistema físico acaba inserindo uma maior sobrecarga, conforme já verificado em diversos trabalhos [1].

As ferramentas VMWare Server e Virtual Box [16] são os exemplos mais comuns que implementam essa abordagem, executada sobre um sistema hospedeiro. Os monitores Xen e VMWare ESX, se apoiados por um suporte à virtualização fornecido pelo processador, podem executar diretamente sobre o *hardware* oferecendo virtualização total.

2.3 HVM - Hardware Virtual Machine

O suporte a virtualização nativo no *hardware*, necessário para construção de monitores eficientes, encontrava-se presente somente em computadores de grande porte, como os *mainframes*. Atualmente, esse suporte foi disponibilizado nos processadores Intel e AMD, através de tecnologias denominadas respectivamente de IVT *Intel Virtualization Technology* ou AMD-V *AMD Virtualization*.

O suporte oferecido por estas tecnologias executa cada máquina virtual como um processo independente, diminuindo a necessidade de interrupções constantes do MMV. Ambas as tecnologias definem dois modos de execução, o modo *root* utilizado pelo monitor e que possui todos os privilégios de operação e o modo *non-root*, com os privilégios necessários para que a MV execute sem saber que está sendo virtualizada ao mesmo tempo que possa ser controlada pelo hipervisor [6].

Essa tecnologia oferece vantagens como a execução de um MMV mais simples e com menos responsabilidades, não sendo necessário o compartilhamento do espaço de endereçamento de memória com as MV's, resultando em uma utilização mais eficaz da mesma. Ainda possibilita a virtualização das interrupções, uma transição mais eficiente entre os modos de operação e acesso completo do monitor ao estado da execução [3].

3 Monitor de Máquinas Virtuais Xen

O Xen é um MMV de código aberto que suporta arquiteturas como IA-32 e AMD64. Implementando originalmente a técnica de paravirtualização, a partir da versão 3.0 passou a oferecer também a virtualização total, desde que executado sobre um *hardware* HVM. A arquitetura Xen segue uma estrutura de 4 anéis, sendo o anel 0 o nível mais baixo, e o anel 3 o mais alto e menos privilegiado.

Quando executa de forma paravirtualizada, o hipervisor Xen, o *kernel* e as aplicações, executam respectivamente nos anéis 0, 1 e 3, como pode ser visto na Figura 1 (a). Desta forma o monitor tem maior privilégio de execução e controla todas as chamadas de sistema e interrupções da MV. Embora sejam necessárias algumas modificações no

núcleo do sistema, essa abordagem demonstra um desempenho superior quando comparada à virtualização total.

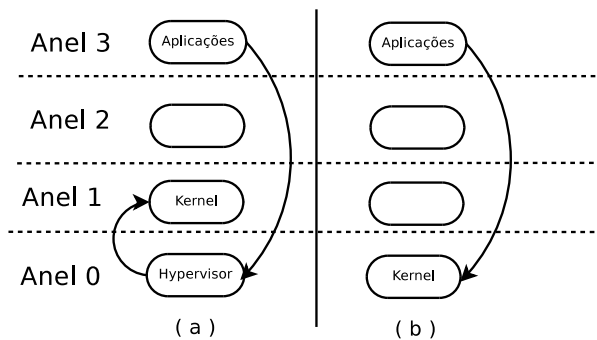


Figura 1. Organização da arquitetura utilizada pelo MMV Xen

No entanto, levando em consideração o suporte atual fornecido pelas arquiteturas mencionadas anteriormente, o *kernel* e as aplicações passam a executar nos anéis 0 e 3 respectivamente, conforme a Figura 1 (b), fazendo com que as aplicações passem a ter acesso direto ao núcleo, eliminando a interferência constante necessária do monitor para traduzir chamadas de sistemas e interrupções [2].

Dessa forma torna-se possível através do MMV Xen, virtualizar aplicações que não permitem que seu código seja modificado. A utilização desta técnica já foi alvo de análises e comparações, mostrando que WMWare ESX 3.0.1 e Xen 3.2, ambos oferecendo virtualização total, apresentaram desempenhos semelhantes [15, 20].

Embora este suporte em *hardware* simplifique o MMV, ele não o elimina, o que faz com que o Xen gerencie de formas distintas o acesso a dispositivos e controle das máquinas virtuais, de acordo com a tecnologia adotada. A seguir será abordada a forma de implementação dos dispositivos foco deste trabalho, utilizada pelo Xen.

Em máquinas paravirtualizadas (PVM), a implementação de acesso a disco e dispositivos de rede é realizada com a utilização de *device drivers* especiais que exportam e compartilham os recursos com os domínios. Cada domínio (DomU) utiliza-se de um *frontend driver* para se comunicar com o *backend* do domínio zero (Dom0), que acessa diretamente os dispositivos físicos. Nos domínios totalmente virtualizados (HVM), estes dispositivos são emulados e o acesso é realizado com o auxílio de um *daemon* Qemu no Dom0, e um *firmware* virtual do Xen que simulam o acesso a estes dispositivos [21], o que tende a inserir uma maior sobrecarga no desempenho destes dispositivos.

A virtualização de memória em domínios paravirtualizados é obtida através de um mapeamento estático da mesma realizado pelo MMV, reservando uma porção de memória

para cada MV, fazendo com que cada domínio acesse somente a sua região mapeada. A diferença nos domínios HVM consiste em funções adicionais no processador e controlador de memória que permitem um acesso mais direto a mesma.

Por fim, a utilização dos recursos de CPU em domínios PVM é feita através da interceptação do conjunto de instruções pelo monitor, que executa em um nível mais privilegiado e devolve para o domínio, ao passo que em máquinas HVM, o acesso se dá na maioria das vezes diretamente ao processador, sendo interferido pelo MMV somente nos casos em que as instruções possam prejudicar ou danificar o sistema.

4 Análise Comparativa

Os testes realizados têm como principal objetivo analisar o desempenho do acesso a disco, memória, utilização de CPU e comunicação de rede face às técnicas de virtualização implementadas pelo MMV Xen. O ambiente de testes foi configurado de três formas: sem virtualização, com virtualização total e com paravirtualização, para qual foi desabilitado o suporte provido pelo processador. Em todos os testes foi criada apenas uma MV em cada máquina e para a análise de rede, os testes foram realizados sobre duas máquinas idênticas interconectadas.

As máquinas utilizadas foram dois servidores Intel x86_64 SGI Altix XE 210, com CPU Intel Xeon E5335 2.0 GHz e memória de 8GB. O *kernel* utilizado para os testes foi o 2.6.20, com Xen versão 3.2.0. A conexão entre as máquinas foi realizada através de um *switch* Ethernet de 100 Mbps e os testes foram realizados em uma rede isolada, proporcionando um ambiente controlado, evitando influências de tráfego adicional.

Os *benchmarks* utilizados para cada análise foram escolhidos devido a sua ampla utilização em avaliações de desempenho, sendo os mesmos abordados nas seções a seguir. Buscando um resultado mais confiável, realizaram-se 10 execuções para todos os valores obtidos, calculando-se a média aritmética, o desvio padrão e o coeficiente de variação.

Para a análise de desempenho da comunicação de rede, escolheu-se o *benchmark* Netperf [7], que permite realizar diversos testes através das modificações de seus parâmetros e métricas. Nos testes realizados foram alterados os tamanhos das mensagens utilizando como métrica a taxa de transferência, através de comunicações Request-Response com os protocolos TCP e UDP. Os grupos de tamanhos de mensagens utilizados foram classificados como pequenas (até 512 bytes), médias (até 512 Kbytes) e grandes (até 45 Mbytes), tendo sido estes tamanhos escolhidos por representarem desde serviços básicos de rede até transferências de arquivos.

O *benchmark* STREAM [10] é utilizado na análise de largura de banda de memória. Este programa mede o desempenho através de quatro testes de processamento vetorial, onde os vetores são aumentados para eliminar o reuso de cache e descrever os resultados em termos de largura de banda contínua. Os resultados obtidos com este teste são:

- *Copy*: analisa a taxa de transferência através da operação $a(i) = b(i)$;
- *Scale*: realiza operações aritméticas simples através da operação $a(i) = q * b(i)$;
- *Add*: acrescenta um terceiro operando para permitir que múltiplas operações de load/store sejam testadas em máquinas vetoriais, através da operação $a(i) = b(i) + c(i)$;
- *Triad*: múltiplas operações de soma e multiplicação através da operação $a(i) = b(i) + q * c(i)$;

Para realizar as medições de acesso ao disco, optou-se por uma ferramenta nativa do SO Linux, o *dd*, que realiza cópias de arquivos, de entrada padrão para uma saída padrão, podendo utilizar diferentes tamanhos de blocos de entrada e saída. A execução do *dd* retorna dois valores, a taxa de transferência de dados entre os arquivos em Mbytes/s e o tempo dessa operação em segundos.

A análise de utilização de CPU foi realizada com a utilização do Linpac-pc, que contém dois conjuntos de rotinas: um para decomposição de matrizes e outro para resolver o sistema de equações lineares resultantes da decomposição. O mesmo foi escolhido devido à métrica utilizada e a sua ampla utilização em testes de desempenho. Os testes foram realizados com matrizes de tamanho 100 x 100, utilizando precisão simples, sendo a saída deste *benchmark* o resultado das operações de ponto flutuante sobre as matrizes, em Mflops.

5 Resultados Obtidos

Ao longo desta Seção apresenta-se os resultados obtidos durante a execução dos *benchmarks* escolhidos, realizando uma análise comparativa entre as abordagens utilizadas. Na Tabela 1, são apresentados todos os coeficientes de variação (CV) máximos obtidos em cada grupo de testes, onde as siglas P, M e G representam os tamanhos dos grupos de mensagens (pequenas, médias e grandes respectivamente), e as letras C, S, A e T, as iniciais de cada teste de memória (*Copy*, *Scale*, *Add* e *Triad* respectivamente).

5.1 Desempenho de Rede

Nos gráficos das Figuras 2 e 3 tem-se os resultados das transferências de mensagens consideradas de tamanho pe-

Tabela 1. Coeficientes de Variação

Teste	PVM	HVM	SV
TCP P	$8,2 \times 10^{-2}$	$3,73 \times 10^{-2}$	$13,76 \times 10^{-2}$
TCP M	$1,06 \times 10^{-2}$	$17,38 \times 10^{-2}$	$1,62 \times 10^{-2}$
TCP G	$1,49 \times 10^{-2}$	$7,38 \times 10^{-2}$	$1,53 \times 10^{-2}$
UDP P	0,11	0,01	0,12
UDP M	0,02	0,6	0,17
Disco	0,37	0,06	0,26
Mem C	$3,26 \times 10^{-2}$	$1,27 \times 10^{-2}$	$1,71 \times 10^{-2}$
Mem S	$3,31 \times 10^{-2}$	$1,38 \times 10^{-2}$	$1,37 \times 10^{-2}$
Mem A	$2,54 \times 10^{-2}$	$0,83 \times 10^{-2}$	$0,94 \times 10^{-2}$
Mem T	$2,48 \times 10^{-2}$	$0,75 \times 10^{-2}$	$0,42 \times 10^{-2}$
CPU	$4,42 \times 10^{-4}$	$8,92 \times 10^{-2}$	$4,52 \times 10^{-2}$

queno e médio, obtidas com o protocolo TCP. Estes tamanhos são considerados comuns em serviços básicos de rede, o que justifica a sua escolha.

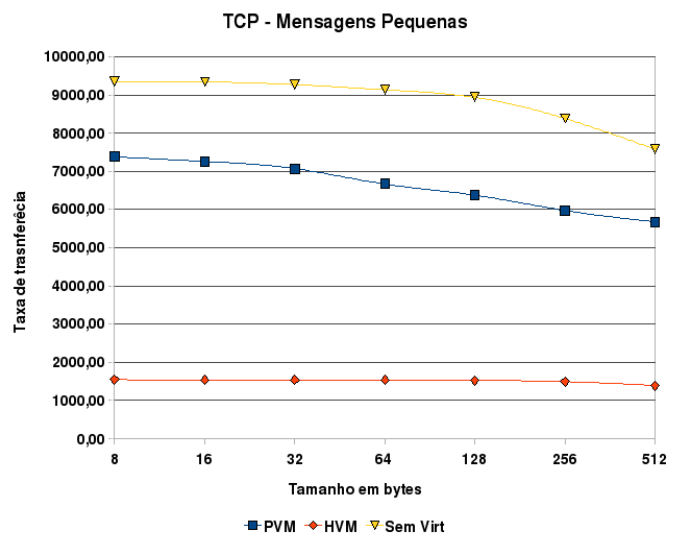


Figura 2. Taxas de transferência TCP, com mensagens pequenas, até 512 bytes

No grupo de mensagens com tamanho até 512 bytes, nota-se o visível impacto da virtualização sobre o desempenho da rede, independente da técnica utilizada. No grupo de mensagens 512 Mbytes, o impacto apresentado foi mais acentuado na técnica de virtualização total, tendo a paravirtualização um desempenho relativamente superior, aproximando-se do desempenho da máquina real a medida que o tamanho das mensagens aumenta.

Os resultados das transferências com mensagens de tamanho grande com a utilização do protocolo TCP são mostrados na Figura 4, onde é possível notar que as mensagens atingem no máximo 45 Mbytes. Esta limitação foi imposta

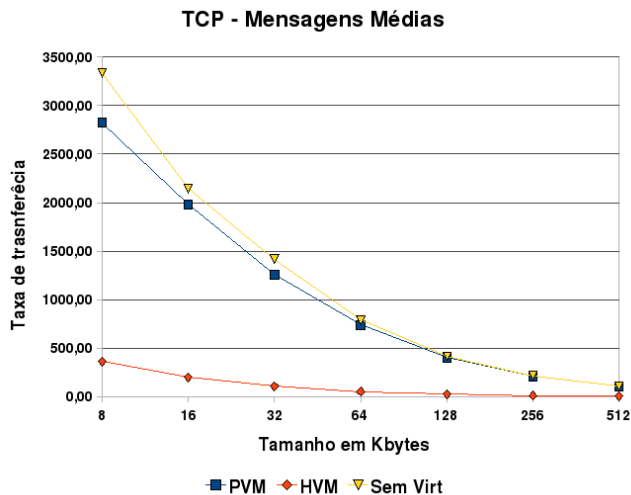


Figura 3. Taxas de transferência TCP, com mensagens médias, até 512 Kbytes

pelo *benchmark* Netperf, o qual não permite o envio de mensagens maiores que 50 Mbytes em média.

A escolha destes tamanhos de mensagens se justifica pelo seu uso em aplicações que realizam transferências de arquivos. Neste grupo de mensagens, as máquinas HVM mantêm seu desempenho inferior ao passo que as máquinas PVM apresentam um desempenho eficiente, praticamente igualando-se ao da máquina real.

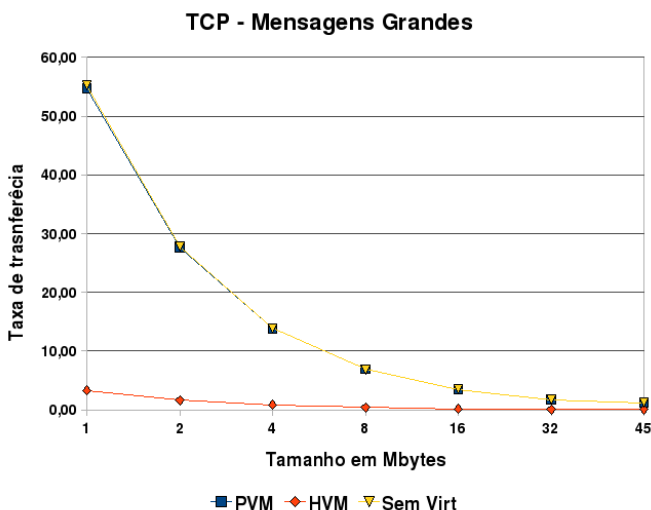


Figura 4. Taxas de transferência TCP, com mensagens grandes, até 45 Mbytes

A análise realizada com o protocolo UDP é apresentada

nas Figuras 5 e 6 a seguir, onde são analisadas mensagens de dois grupos de tamanho, pequeno e médio. A realização de testes com um grupo de mensagens maiores que 62 Kbytes, não foi possível devido as limitações impostas pelo *benchmark* utilizado.

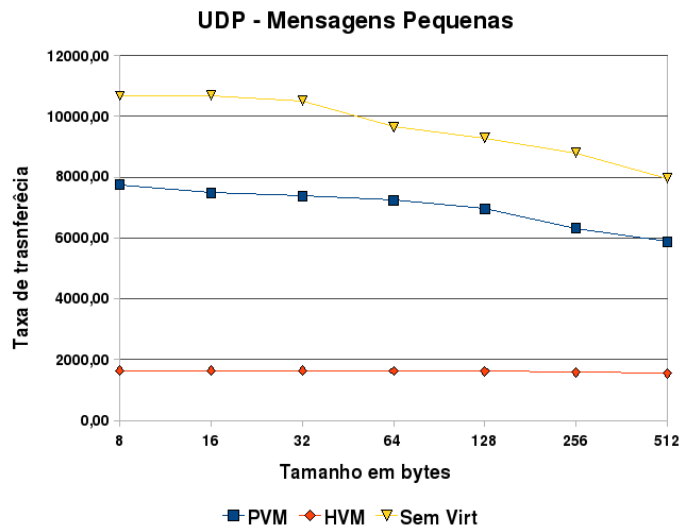


Figura 5. Taxa de transferência UDP, com mensagens pequenas, até 512 bytes

Na transferência de mensagens UDP até 512 bytes é possível observar o impacto de ambas as tecnologias utilizadas, sendo mais acentuada nas máquinas HVM. Com o aumento do tamanho das mensagens até 62 Mbytes, ocorre uma diminuição da sobrecarga imposta pelas máquinas PVM, aproximando-se do desempenho do sistema hospedeiro, conforme já observado em outras análises [18, 14]. É possível observar que da mesma forma que com o protocolo TCP, à medida que aumentam os tamanhos das mensagens do protocolo UDP, a diferença de desempenho diminui.

5.2 Desempenho de memória

O gráfico mostrado na Figura 7 expõe as taxas obtidas nos testes de memória. Através deste pode-se observar uma superioridade no desempenho das máquinas HVM em relação as PVM, mas de maneira geral, o desempenho da memória não sofre muito impacto independentemente da técnica de virtualização empregada.

5.3 Desempenho de Disco

A seguir, são analisados os resultados obtidos com o programa dd, mantendo seu tamanho de blocos padrão (512 bytes), modificando apenas os tamanhos dos arquivos a serem

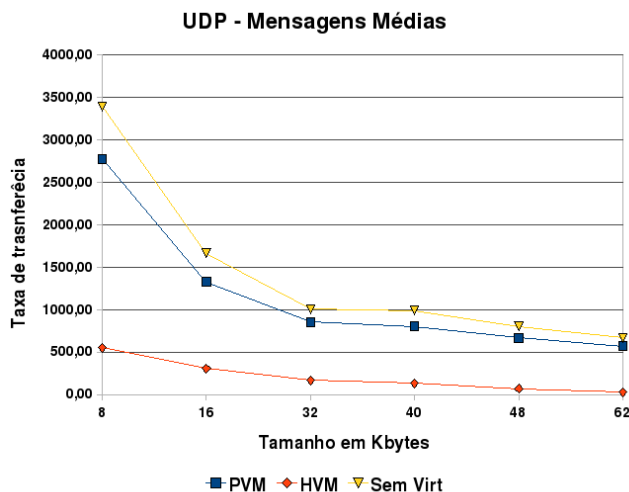


Figura 6. Taxa de transferência UDP, com mensagens médias, até 62 Mbytes

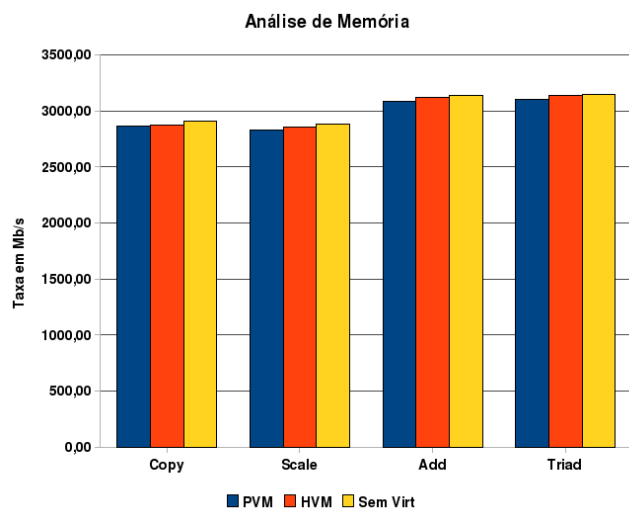


Figura 7. Taxa de análise de memória em Mbytes/s

criados. Os testes realizados envolveram arquivos de 6 tamanhos distintos, variando entre 128 Mbytes e 4 Gbytes, os quais foram escolhidos devido a englobarem tamanhos de arquivos normalmente utilizados em diversas aplicações.

A taxa de transferência para cada tamanho de arquivo obtida é mostrada na Figura 8, onde observa-se de forma geral que o desempenho da máquina HVM é inferior ao da máquina PVM. Mais detalhadamente, para arquivos menores, a sobrecarga inserida pela virtualização total é bem elevado ao passo que o da paravirtualização é relativamente

baixo. À medida que o tamanho dos arquivos vai aumentando, a máquina PVM aproxima-se das taxas da máquina HVM, apresentando um relativo distanciamento do desempenho sem virtualização.

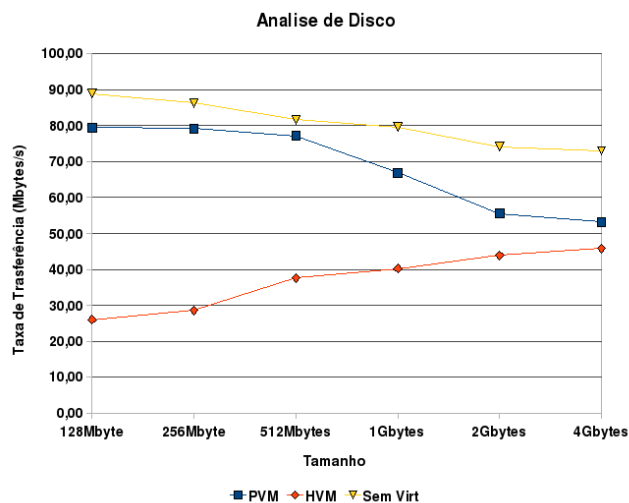


Figura 8. Taxa de transferência de arquivos no disco em Mbytes/s

O segundo dado obtido pelo aplicativo é o tempo de transferência entre os arquivos, mostrado na Figura 9, a qual confirma a maior sobrecarga da máquina HVM e reforça o baixo impacto causado pela técnica de paravirtualização, pois os tempos de transferência são similares aos da máquina real.

5.4 Utilização de CPU

As análises de CPU realizadas com o *benchmark* Linpac mostraram resultados praticamente idênticos entre as tecnologias empregadas e a máquina real, permanecendo em torno de 218,30 Mflops.

5.5 Análise dos Resultados

Conforme já mencionado anteriormente, nestas análises foi possível verificar que embora uma MV HVM tenha acesso direto ao *kernel*, sem interferência direta do MMV, a sobrecarga causada pelo *firmware* na MV e o *daemon* no hipervisor é consideravelmente grande face a máquinas PVM. A emulação dos dispositivos de I/O prejudica consideravelmente o desempenho de rede e acesso a disco em máquinas virtuais HVM, as quais apresentam um desempenho inferior tanto para máquina real como para PVM, principalmente para mensagens e arquivos menores, nos quais o desempenho sem virtualização é significativamente maior.

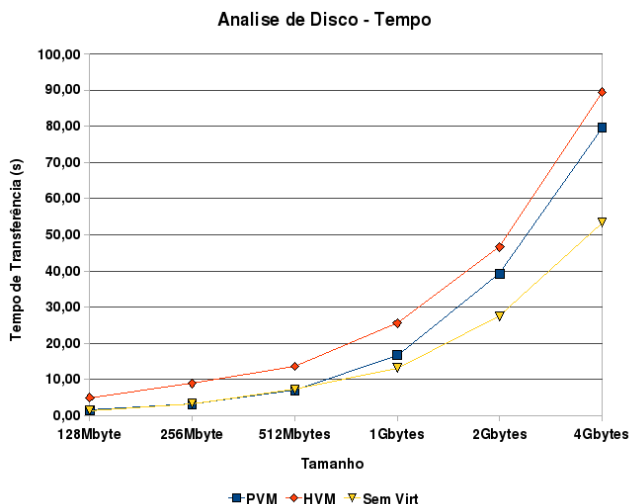


Figura 9. Tempo de transferência de arquivos no disco em segundos

Embora ambas as técnicas utilizadas mapeiem e reservem uma porção da memória para a MV, na virtualização de acesso a memória, o melhor desempenho da técnica de virtualização total é justificado pela diferença na forma como o Monitor de Máquinas Virtuais implementa este acesso em relação as máquinas paravirtualizadas. As características que proporcionam um melhor desempenho aos domínios HVM são o não compartilhamento de memória com as máquinas virtuais e a utilização de instruções que permitem um acesso direto a mesma, diminuindo assim a sobrecarga sobre a virtualização de componente.

Com relação a utilização da CPU, o desempenho obtido foi praticamente o mesmo, apresentando mínimas variações nas medições, conforme a Tabela 1, e uma diferença de 0,5 Mflops entre a MV HVM e as outras máquinas.

6 Considerações Finais

Neste trabalho analisou-se o desempenho das comunicações de rede entre máquinas não virtuais e máquinas virtualizadas com as tecnologias de paravirtualização e virtualização total implementadas pelo hipervisor Xen.

De modo geral, a virtualização assistida por *hardware* oferecida pelo Xen deixa muito a desejar nos quesitos de comunicação de rede e acesso a disco, se comparada com a sua abordagem de paravirtualização, a qual permanece como uma solução mais eficaz e com um menor impacto sobre o desempenho destes dispositivos.

No entanto, se a necessidade de utilização de sistemas cujo *kernel* não permite modificações sobrepõem as neces-

sidades de desempenho, é imperativo uso da técnica de virtualização total, visto que a paravirtualização não permite a execução destes sistemas.

Assim, dependendo da abordagem utilizada, podem-se obter resultados de desempenho significativamente diferentes, que atrelados ao tipo de uso das MV e aos sistemas a serem utilizados, podem prover um embasamento para que administradores de sistemas e até mesmo usuários convencionais realizem uma opção entre utilizar ou não a virtualização ou no caso de uso da mesma, qual tecnologia melhor atende ao seus requisitos.

Como trabalhos futuros, tem-se a intenção de comparar estas abordagens com a virtualização suportada pelo KVM – *Kernel-based Virtual Machine*, disponibilizado no *kernel* Linux a partir da versão 2.6.20 [9].

Referências

- [1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proc. 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, pages 164–177, Bolton Landing, USA, Oct. 2003. ACM.
- [2] D. Chisnall. *The Definitive Guide to the Xen Hypervisor*. Prentice Hall, Boston, 2007.
- [3] G. A. da Silva Junior. Criação de sistemas em tempo real em ambientes virtualizados. Technical report, Monografia (Trabalho de Graduação) — Curso de Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2005.
- [4] A. B. de Castro. Máquinas virtuais em ambiente seguros. Technical report, Monografia (Dissertação de Mestrado) — Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2006.
- [5] R. Goldberg. Survey of virtual machine research. *IEEE Computer*, 7(6):34–45, 1974.
- [6] Intel Corporation. Intel, 2008. Disponível em: <http://www.intel.com/>. Acesso em: junho de 2008.
- [7] R. Jones. *Netperf Manual 2.4.3*. Hewlett-Packard, 2007. Disponível em: <http://www.netperf.org/netperf/>. Acesso em: agosto de 2007.
- [8] M. Laureano. *Máquinas Virtuais e Emuladores*. Novatec, São Paulo, 2006.
- [9] LinuxInsight. Finally user-friendly virtualization for linux, 2008. Disponível em: <http://www.linuxinsight.com/finally-user-friendly-virtualization-for-linux.html>. Acesso em: junho de 2008.
- [10] J. D. McCalpin. Stream, 1996. Disponível em: <http://www.cs.virginia.edu/stream/>. Acesso em: junho de 2008.
- [11] P. Padala, X. Zhu, Z. Wang, S. Singhal, and K. Shin. Performance Evaluation of Virtualization Technologies for Server Consolidation. Technical report, Hewlett-Packard Development Company, L.P., 2007.
- [12] Qumranet. Kernel based virtual machine, 2008. Disponível em: <http://kvm.qumranet.com/kvmwiki>. Acesso em: junho de 2008.

- [13] B. Quétiér, V. Neri, and F. Cappello. Scalability Comparison of Four Host Virtualization Tools.
- [14] A. H. Schmidt, M. P. Bouffleur, R. C. M. d. Santos, and A. S. Charão. Análise de Desempenho da Virtualização de Rede nos Sistemas Xen e OpenVZ. *V Escola Regional de Redes de Comutadores, Santa Maria, RS*, 2007.
- [15] X. Source. A performance comparasion of commercial hipervisors, 2007. Disponível em: http://www.xensource.com/Document/hypevisor_performance_comparison.1.0.5_with_esx-data.pdf. Acesso em: junho de 2008.
- [16] Sun Microsystems, Inc. Virtual Box, 2008. Disponível em: <http://www.virtualbox.org/>. Acesso em: junho de 2008.
- [17] SWsoft. Server Virtualization Open Source Project, 2007. Disponível em: <http://openvz.org/>. Acesso em: agosto de 2007.
- [18] F. Urschei, J. P. Pelegriani, M. A. L. S. Lima e Silva, E. M. Midorikawa, and T. C. C. Carvalho. Análise Multiparamétrica do Overhead de Rede em Máquinas Virtuais. *IV Workshop de Sistemas Operacionais (WSO'2007)*, 2007.
- [19] VMware Inc. Página da empresa VMware Inc., 2007. Disponível em: <http://www.vmware.com>. Acesso em: agosto de 2007.
- [20] Xen-BR. Esclarecimentos dos benchmarks da VMware e da Xensource, 2007. Disponível em: <http://wiki.xen-br.org/index.php?title=Vmware-xen>. Acesso em: agosto de 2007.
- [21] Xen.org. Xen architecture overview, 2008. Disponível em: <http://xen.org/xen/documentation.html>. Acesso em: junho de 2008.
- [22] Xensource. Xen Hypervisor, 2008. Disponível em: <http://www.xen.org/>. Acesso em: junho de 2008.
- [23] L. Youseff, R. Wolski, B. Gorda, and C. Krintz. Paravirtualization for HPC Systems. Technical report, UCRL-TR-225347, Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2006.