

# Avaliação do Uso de Xen em Ambientes de Computação de Alto Desempenho

Márcio Parise Bouffleur, Guilherme Piegas Koslovski, Andrea Schwertner Charão  
Laboratório de Sistemas de Computação (LSC)  
Curso de Ciência da Computação – Universidade Federal de Santa Maria (UFSM)  
Campus UFSM – 97105-900 – Santa Maria – RS – Brazil  
{bouffleur, guilherm, andrea}@inf.ufsm.br

## Resumo

*A virtualização de recursos computacionais tem sido alvo de pesquisas e desenvolvimentos recentes. Máquinas virtuais provêem uma abstração do hardware e arquitetura subjacentes, permitindo que múltiplos sistemas e aplicações executem concorrentemente sobre os recursos virtualizados. Em ambientes de computação de alto desempenho, a virtualização pode auxiliar no gerenciamento de recursos em clusters e grades computacionais, mas pode comprometer o desempenho das aplicações que executam nestes ambientes. Neste artigo, apresenta-se uma avaliação experimental do impacto da utilização do monitor de máquinas virtuais Xen em um ambiente de execução de aplicações paralelas e distribuídas. Em particular, analisa-se a sobrecarga imposta pela virtualização e pelo mecanismo de migração de máquinas virtuais oferecido por Xen.*

## 1. Introdução

A virtualização de recursos computacionais é um tema recorrente em trabalhos de pesquisa e desenvolvimento em diversas áreas da ciência da computação. A abstração oferecida pelas máquinas virtuais permite mapear recursos reais de uma mesma arquitetura às necessidades de diferentes sistemas e aplicações, contribuindo para a flexibilidade, a portabilidade, a interoperabilidade e a gerenciabilidade de sistemas de *software*.

Na área de computação de alto desempenho, técnicas de virtualização podem ser utilizadas para facilitar o gerenciamento e o aproveitamento de recursos em *clusters* e grades computacionais. Dentre os benefícios da virtualização, destacam-se a segurança e a confiabilidade obtidas através do isolamento oferecido pelas máquinas virtuais, a facilidade de configuração do *software* básico (sistema operacional, *middleware*, etc.) conforme a necessidade das

aplicações e a possibilidade de controle dinâmico de recursos através da migração de máquinas virtuais.

Em um ambiente virtualizado, as tarefas de gerenciamento, escalonamento e alocação dos recursos disponíveis são executadas por um Monitor de Máquinas Virtuais (MMV). Este monitor virtualiza o *hardware* existente, oferecendo uma interface individual para cada máquina virtual, permitindo desta forma uma execução transparente e independente de uma arquitetura específica. Dentre os monitores de máquinas virtuais existentes, o sistema Xen [3] destaca-se por ser um MMV de código aberto e por permitir a migração de máquinas virtuais entre computadores sem a necessidade de interromper os processos que se executam sobre estas máquinas.

Alguns trabalhos recentes avaliam o desempenho de Xen em diferentes plataformas de *hardware* e *software*, e indicam que sistemas operacionais executando sobre este MMV têm desempenho próximo ao de sistemas que executam em ambiente não-virtualizado [14, 4]. No entanto, existem poucas avaliações do impacto do uso de Xen na execução de aplicações típicas de computação de alto desempenho. Desta forma, o objetivo deste trabalho é ampliar os resultados experimentais sobre técnicas de virtualização aplicadas a sistemas computacionais de alto desempenho, através de uma avaliação de Xen e de seu mecanismo de migração em diferentes casos neste domínio de aplicações.

Este artigo está organizado na seguinte forma: na seção 2 descreve-se o funcionamento e as principais características de monitores de máquinas virtuais, enquanto na seção 3 apresenta-se Xen e seu mecanismo para migração de máquinas virtuais. As principais contribuições deste trabalho encontram-se nas seções 4 e 5, onde discute-se alguns trabalhos que utilizam Xen em ambientes para computação de alto desempenho e, em seguida, apresenta-se a avaliação experimental de desempenho de Xen. Na seção 6, faz-se algumas considerações finais sobre este trabalho e sobre os resultados obtidos.

## 2. Monitores de máquinas virtuais

A virtualização de recursos computacionais é um conceito que foi bastante difundido no início da década de 70, permitindo o compartilhamento e melhor aproveitamento do poder computacional dos *mainframes* existentes [8]. Atualmente, fatores como aumento de desempenho dos computadores atuais e utilização de ambientes interligados impulsionaram novamente a utilização de ambientes virtualizados.

Soluções de virtualização envolvem basicamente o uso de uma camada de *software* que permite que múltiplas máquinas virtuais (possivelmente executando diferentes sistemas operacionais) utilizem os recursos de uma dada máquina real. O gerenciamento destas máquinas virtuais é de responsabilidade de um Monitor de Máquinas Virtuais. Um MMV oferece, aos diferentes sistemas executando sobre suas máquinas virtuais, a ilusão um ambiente real, semelhante à arquitetura original do computador hospedeiro. A eficiência de um monitor é medida de acordo com sua intrusividade na interceptação e manipulação das instruções virtualizadas.

Dentre as abordagens de implementação de monitores de máquinas virtuais, distingue-se os sistemas de virtualização hospedada dos sistemas de virtualização clássica [15]. Em sistemas de virtualização hospedada, o monitor é instalado sobre um sistema operacional hospedeiro, e suas máquinas virtuais são executadas sobre essa camada de *software* oferecida. Um exemplo de MMV que utiliza esse modelo de implementação é o sistema *VMware* [16].

Em sistemas de virtualização clássica, o monitor de máquinas virtuais é uma camada instalada diretamente sobre o *hardware*, permitindo que a execução do monitor ocorra com o mais alto nível de privilégio. Desta forma, as interrupções e chamadas de sistema solicitadas pelos sistemas operacionais virtualizados podem ser interceptadas e manipuladas pelo monitor. Este modelo de implementação, tipicamente utilizado em *mainframes*, apresenta uma limitação perante aos computadores atuais que utilizam arquiteturas semelhantes à IA-32, onde um sistema operacional, embora virtualizado, ainda consegue executar instruções em um nível de maior privilégio.

Para contornar esta limitação na abordagem de virtualização clássica, sistemas como Denali [17] e Xen [3] utilizam uma técnica conhecida como *paravirtualização*, onde a interface apresentada às máquinas virtuais difere ligeiramente da interface real da arquitetura subjacente. Esta abordagem requer uma adaptação do sistema operacional virtualizado, particularmente para lidar com instruções que executariam com maior privilégio.

## 3. Visão geral de Xen

Xen [3] é um MMV de código aberto, com suporte a arquiteturas das famílias IA-32, AMD64 e EM64T, e que permite usar Linux ou NetBSD como sistemas operacionais hospedeiros. Nestas arquiteturas, Xen utiliza a técnica de paravirtualização mencionada na seção anterior, permitindo a execução de múltiplos sistemas operacionais em um mesmo computador hospedeiro. Embora os sistemas operacionais virtualizados precisem ser modificados para funcionar sobre Xen, esta abordagem é eficiente e não requer modificações em aplicações que executam sobre as máquinas virtuais. A virtualização oferecida através de Xen resulta em uma máquina virtual independente do *hardware* existente, que pode ser encapsulada e migrada entre computadores interligados em rede. O restante desta seção apresenta a arquitetura de Xen e suas ferramentas auxiliares, além de descrever o mecanismo de migração suportado por este MMV.

### 3.1. Arquitetura e ferramentas

A arquitetura de Xen distingue dois elementos principais: o monitor de máquinas virtuais propriamente dito (também chamado *hypervisor*) e as máquinas virtuais que executam sobre o monitor (chamadas de domínios Xen). O monitor tem acesso direto ao *hardware* e executa no modo de maior privilégio do processador, enquanto as máquinas virtuais são impedidas de executar diretamente as instruções em modo privilegiado. Nesta arquitetura, um domínio especial (chamado *Domain0*) é responsável pela criação, terminação e migração dos demais domínios hospedados. Para isso, este domínio tem acesso a uma interface de controle oferecida pelo monitor.

A interface oferecida pelo MMV inclui chamadas que os domínios hospedados devem utilizar para execução de instruções e operações privilegiadas. Além disso, tem-se um mecanismo para notificação de eventos às máquinas virtuais e um mecanismo para transferência de mensagens de entrada e saída entre máquinas virtuais.

Para virtualizar a memória física, o MMV particiona (reserva) estaticamente a memória entre os domínios, provendo assim um forte isolamento entre eles. Um domínio também pode convenientemente liberar memória para economizar recursos, reduzindo assim sua reserva de memória e devolvendo-a para Xen.

O acesso à memória por parte dos domínios não sofre uma penalidade tão elevada como em outras técnicas de virtualização, como por exemplo a utilizada em *VMware* [16], que se utiliza de tabelas de páginas virtuais, invisíveis ao gerenciador de memória. Ao invés de virtualizar as tabelas de páginas, o MMV permite a todos os domínios o acesso somente de leitura ao gerenciador

de memória, enquanto o acesso para atualizações é feito através de uma chamada ao *hypervisor*. Este, por sua vez, as valida antes de serem aplicadas, de modo a garantir o isolamento das máquinas virtuais. Essa técnica aumenta consideravelmente o desempenho, pois diminui a complexidade no acesso à memória.

O conjunto de ferramentas de administração oferecidas por Xen permitem facilmente gerenciar todos os aspectos do sistema. Através da ferramenta `xm` tem-se acesso a informações sobre o estado das máquinas virtuais, bem como à submissão de comandos, como *checkpoint* e migração para outro computador.

A instalação do sistema como um todo é relativamente simples. Até o presente momento porém, as principais distribuições do sistema operacional Linux ainda não chegaram a uma forma de instalação pré-compilada que possa ser utilizada para a distribuição automática em diversos computadores simultaneamente, o que pode ser um empecilho para implantação em *clusters* de computadores. Contudo, esta situação tende a se resolver em um futuro próximo, devido à rápida popularização de Xen.

### 3.2. Migração de máquinas virtuais

Xen implementa um mecanismo chamado de *live migration* [5], onde uma máquina virtual pode migrar de um computador hospedeiro para outro sem que sua execução precise ser suspensa e reiniciada mais tarde. Para isso, a memória da máquina virtual é copiada iterativamente para o computador de destino, de modo a permitir que os serviços oferecidos pelo sistema virtualizado continuem em execução. O mecanismo de *live migration* inicia com uma reserva de recursos (pré-cópia) no computador de destino, seguida de uma transferência de dados iterativa e por demanda, onde somente são transferidos os dados alterados durante a última iteração realizada.

Durante as iterações, o mecanismo de *live migration* busca identificar uma condição onde a suspensão da máquina virtual e a ativação no computador de destino resulte em um tempo de indisponibilidade mínimo. Caso não seja possível identificar uma condição satisfatória para a migração e o número de iterações realizadas seja alto, a execução da máquina virtual é suspensa e a migração é concluída, evitando que o tempo total de migração seja elevado.

É importante mencionar que Xen não utiliza nenhum método de migração de sistemas de arquivos, supondo que os sistemas operacionais virtualizados utilizem um sistema de compartilhamento de arquivos tal como NFS. Além disso, para implementar a migração das conexões de rede, Xen optou pela difusão de um *ARP reply* [13] informando que o endereço IP da máquina migrada foi movido para uma nova localização. Neste modelo de implementação não é necessário o redirecionamento de pacotes, diminuindo a de-

pendência residual entre a máquina virtual migrada o computador de destino.

## 4. Xen e a virtualização em plataformas de computação de alto desempenho

A utilização de soluções de virtualização em ambientes de computação de alto desempenho tem sido explorada em diversos trabalhos recentes. Huang et al. [9], por exemplo, apresentam uma solução de virtualização baseada em Xen, capaz de explorar eficientemente uma rede de interconexão de alto desempenho tal como InfiniBand, além de prover um mecanismo escalável de gerenciamento de imagens de máquinas virtuais para utilização em aglomerados de computadores.

Quetier et al. [14] apresentam uma análise sobre a escalabilidade de arquiteturas virtualizadas em grades computacionais, onde diversos *microbenchmarks* foram executados a fim de comparar quatro implementações de tecnologias que oferecem virtualização de recursos. Nesta avaliação, o monitor de máquinas virtuais Xen apresentou um bom desempenho em diversos quesitos analisados, embora algumas deficiências tenham sido identificadas (por exemplo na comunicação entre máquinas virtuais através da interface virtual de rede, podendo ter impacto sobre as aplicações em execução).

No trabalho desenvolvido por Figueiredo et al. [7] discute-se a aplicabilidade de soluções de virtualização em grades computacionais. Questões relativas às vantagens e formas de utilização de um ambiente virtualizado em grades computacionais foram discutidas, analisando-se também a sobrecarga da virtualização sobre as aplicações (neste caso utilizando-se VMware como solução de virtualização).

Childs et al. [4] fazem um comparativo entre Xen e UML (*User Mode Linux*) [6] para a hospedagem do serviço de conexão ao Grid-Ireland – que é composto de quatro servidores – em uma única máquina física. Neste trabalho concluiu-se que a sobrecarga imposta por Xen é suficientemente tolerável face às vantagens da utilização de um ambiente virtualizado para as unidades de acesso à grade.

É importante mencionar que, dentre os trabalhos analisados, não encontrou-se uma análise do impacto da migração sobre aplicações de alto desempenho.

## 5. Avaliação de desempenho de Xen

Os testes apresentados ao longo desta seção tiveram como objetivo avaliar a utilização de Xen em cenários característicos de ambientes de computação de alto desempenho. Inicialmente, buscou-se quantificar a sobrecarga causada pelo monitor de máquinas virtuais sobre o tempo total de execução de uma aplicação de computação distribuída,

típica de grades computacionais (seção 5.1). Em um segundo momento, avaliou-se o impacto da virtualização sobre um *benchmark* paralelo tipicamente utilizado em *clusters* (seção 5.2). Por fim, avaliou-se o impacto do mecanismo de migração de máquinas virtuais sobre a execução do *benchmark* paralelo e da aplicação de computação distribuída (seção 5.3).

Para realização destes testes, configurou-se um ambiente de experimentação composto por cinco computadores Compaq Pentium 4, com memória de 512MB e cache de 512KB, utilizando o sistema operacional Ubuntu Linux versão 5.10. Para interligar estes computadores, utilizou-se uma rede FastEthernet a 100Mbps. Dentre estes computadores, quatro foram usados para hospedar as máquinas virtuais e um deles foi configurado como servidor de arquivos NFS. Em todos os experimentos com virtualização, utilizou-se a versão 3.0.2-2 de Xen.

### 5.1. Impacto sobre uma aplicação de computação distribuída

Neste experimento utilizou-se uma aplicação que realiza a análise de grandes conjuntos de dados coletados por sensores meteorológicos [11]. Esta aplicação utiliza uma abordagem *peer-to-peer* (P2P) [10] para computação paralela e distribuída, e foi implementada sobre a infra-estrutura P2P oferecida pela plataforma JXTA [1]. Nesta aplicação, cada *peer* é capaz de processar tarefas enviadas por processos clientes. Uma tarefa consiste basicamente na análise de um conjunto de dados representando um dia de coleta.

Para realizar esta avaliação foram utilizados três computadores do ambiente de experimentação: um responsável pelo processamento (*peer*), outro para envio e solicitação de tarefas (cliente) e outro atuando como servidor NFS para as máquinas virtuais. Somente uma tarefa foi lançada para execução, o que justifica o uso de apenas um *peer*. Esta tarefa não exige comunicação durante seu processamento, mas requer muitos acessos a dados armazenados em disco.

O gráfico 1 apresenta os tempos de execução da aplicação com e sem o uso de virtualização, respectivamente identificados como *MeteoP2P-Xen* e *MeteoP2P*. Analisando-se os resultados obtidos, observa-se um aumento de aproximadamente 16% no tempo total de execução da aplicação quando esta é executada sobre o monitor Xen.

O aumento no tempo de execução quantifica a sobrecarga imposta pela virtualização, neste caso devida principalmente ao alto número de operações que exigem acesso a dados armazenados em disco.

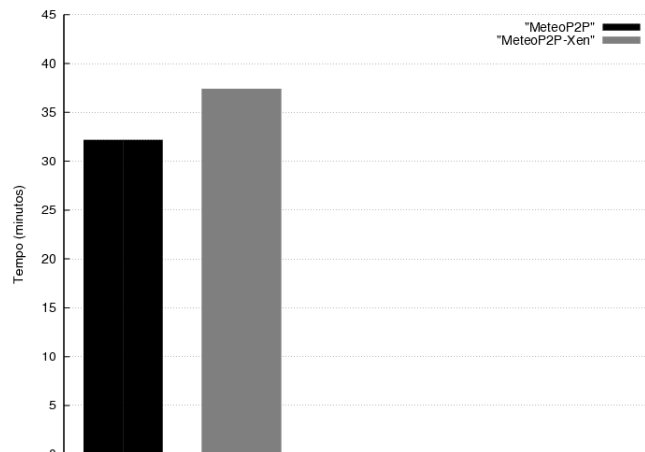


Figura 1. Sobrecarga na aplicação de *MeteoP2P*

### 5.2. Impacto sobre um *benchmark* paralelo

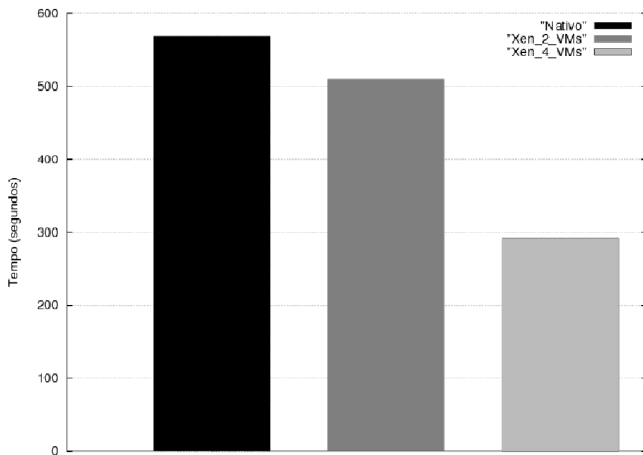
Neste experimento utilizou-se o *benchmark* HPL (*High Performance Linpack*) [12], que calcula a solução de um sistema denso de equações lineares em arquiteturas com memória distribuída, utilizando MPI (*Message Passing Interface*) para troca de mensagens. Este *benchmark* é empregado como critério para a classificação dos supercomputadores de maior desempenho na atualidade (TOP500) [2].

Para execução de HPL, utilizou-se a configuração padrão contida no pacote de instalação do *benchmark*, onde o processamento é distribuído entre quatro processos MPI paralelos.

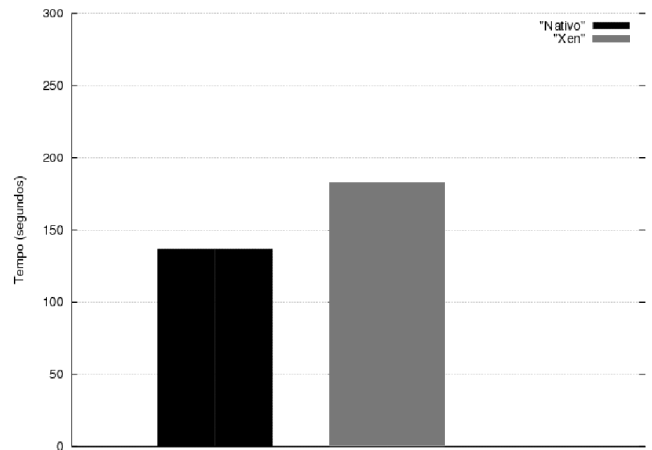
Em uma primeira avaliação, o *benchmark* HPL foi executado utilizando-se três computadores do ambiente de teste: dois computadores para execução paralela e um computador atuando como servidor NFS. A arquitetura nativa foi comparada com a arquitetura virtualizada em duas situações: na primeira, utilizou-se duas máquinas virtuais alocadas a computadores distintos; na segunda, utilizou-se quatro máquinas virtuais, alocando-se duas máquinas virtuais a cada computador.

O gráfico na figura 2 apresenta o resultado da execução do HPL sobre os dois computadores. Neste gráfico, observa-se uma redução do tempo total de execução obtido com a arquitetura virtualizada. Esta redução é de aproximadamente 10% utilizando-se duas máquinas virtuais, e de aproximadamente 49% utilizando-se quatro máquinas virtuais.

Esta redução do tempo de execução ocorre devido à comunicação através do compartilhamento de páginas de memória implementada pelo monitor Xen, que resulta em uma comunicação eficiente, seja entre máquinas virtuais ou



**Figura 2. Tempos de execução utilizando 2 computadores (2 processos MPI em cada máquina)**



**Figura 3. Tempos de execução utilizando 4 computadores (1 processo MPI em cada máquina)**

entre uma máquina virtual e o monitor. Esta vantagem foi explorada durante a execução de mais de um processo MPI sobre o mesmo computador.

Em uma segunda avaliação, utilizou-se o ambiente completo de experimentação, com cada um dos quatro computadores executando uma máquina virtual. Desta forma, tanto na execução nativa quanto na execução virtualizada, cada computador recebeu uma tarefa resultante do particionamento efetuado pelo *benchmark* HPL.

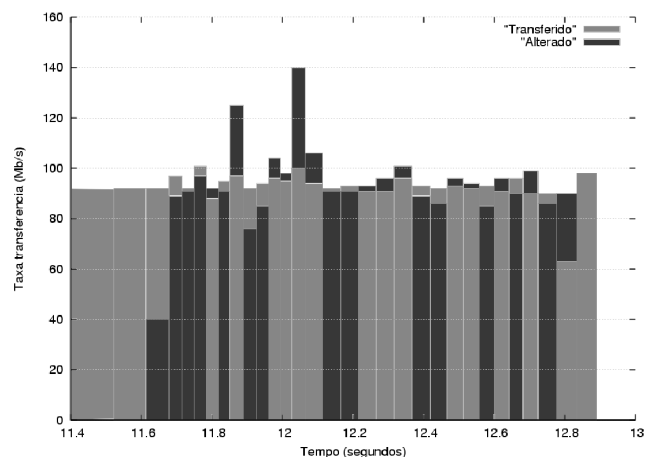
Na figura 3 observa-se a sobrecarga imposta pelo monitor Xen, resultando em tempo de execução aproximadamente 46 segundos maior. Neste ambiente de teste, a vantagem oferecida pelo compartilhamento de páginas implementado pelo monitor não foi explorada, já que cada computador executou apenas um processo MPI.

### 5.3. Impacto da migração de máquinas virtuais

Nesta etapa da avaliação, investigou-se o impacto do mecanismo de *live migration* do monitor Xen na execução das duas aplicações apresentadas nas seções anteriores.

Inicialmente avaliou-se a migração de máquinas virtuais no caso do *benchmark* HPL com 4 processos MPI, sendo executados sobre dois computadores do ambiente de experimentação. Cada computador ficou responsável pela execução de uma máquina virtual, sendo que forçou-se a migração de uma destas máquinas para um terceiro computador durante a execução do *benchmark*. É importante mencionar que o *benchmark* HPL exige um grande número de acessos à memória em um reduzido intervalo de tempo, além de uma quantidade considerável de comunicação entre

processos. Estes fatores influenciam diretamente no tempo de migração de uma máquina virtual.



**Figura 4. Migração de uma máquina virtual executando HPL**

Conforme mencionado na seção 3.2, o mecanismo de *live migration* implementado pelo monitor Xen transfere os dados armazenados na memória principal através de iterações. A figura 4 ilustra as iterações ocorridas durante o processo de migração da máquina virtual executando o *benchmark* HPL. Para cada iteração, apresenta-se o volume de dados transferidos (em cinza) e alterados (em preto) por unidade de tempo.

Na iteração inicial é transferido um volume grande de

dados, representando a fase de pré-cópia utilizada pelo mecanismo de *live migration*. Nesta fase ocorre uma transferência inicial de todos os dados armazenados na memória principal, para permitir que posteriormente somente os dados alterados sejam transferidos. O tempo total de execução da fase de pré-cópia foi de aproximadamente 11,7 segundos (para facilitar a visualização das outras iterações, o gráfico na figura 4 teve o eixo das abscissas deslocado, iniciando em 11,4 segundos ao invés de iniciar no instante zero). Durante este processo é possível observar a interferência da reserva de recursos sobre os demais processos em execução, demonstrada pela inexistência de dados alterados (representados em preto) no intervalo de tempo relativo a essa iteração.

As iterações executadas buscam atingir uma condição onde a migração ocorra de forma rápida, transferindo-se o menor número possível de dados de modo a provocar um tempo mínimo de indisponibilidade do sistema. Na migração representada pela figura 4 a condição mínima não foi encontrada, ou seja, a migração ocorreu após a execução do número limite de iterações.

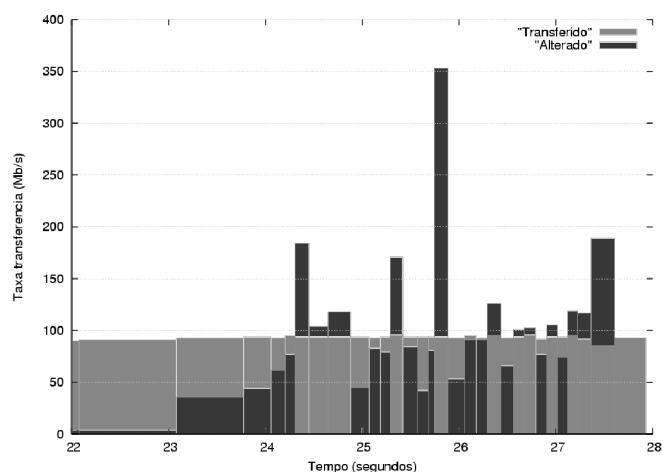
Embora a identificação da melhor condição de migração não tenha ocorrido neste caso, o processo de migração da máquina virtual durou aproximadamente 12,9 segundos, em um tempo total de execução de 4 minutos e 39 segundos, representando aproximadamente 4,6% do tempo total de execução.

No segundo caso considerado, partiu-se do mesmo ambiente de experimentação descrito na seção 5.1 para investigar-se a migração de uma máquina virtual executando a aplicação de computação distribuída (MeteoP2P). Neste caso, migrou-se a máquina virtual que processava os dados para um terceiro computador. A figura 5 apresenta a evolução do processo de migração ocorrido neste caso.

Na figura 5 observa-se as trinta iterações ocorridas para efetuar a migração da máquina virtual. Nesta migração o tempo de execução da iteração inicial correspondente à fase de pré-cópia foi de aproximadamente 22,57 segundos, representando 81% em um tempo total de migração de 27,77 segundos. Este tempo inicial de transferência é proporcional ao grande volume de dados manipulados pela aplicação MeteoP2P (o gráfico representado pela figura 5 também teve o eixo das abscissas deslocado para facilitar a visualização).

Assim como no primeiro caso avaliado, a identificação da melhor condição para migração não ocorreu, ocasionando a suspensão da máquina virtual e a sua ativação no computador de destino após atingir-se o número máximo de iterações. Nesta análise, o tempo total de execução obtido foi de aproximadamente 38 minutos e 40 segundos e o tempo total de migração foi de 27,77 segundos, representando aproximadamente 1,2% do tempo total de execução.

As avaliações efetuadas utilizando-se o mecanismo de



**Figura 5. Migração de uma máquina virtual executando MeteoP2P**

*live migration* em aplicações características de ambientes de computação de alto desempenho indicam que a sobrecarga imposta pelo mecanismo de migração não compromete significativamente o desempenho final da aplicação, viabilizando a utilização deste mecanismo para controle dinâmico de recursos.

## 6. Conclusão

Neste trabalho investigou-se o impacto da utilização de ambientes virtualizados em plataformas de alto desempenho. O principal critério considerado nesta análise foi a intrusividade do monitor de máquinas virtuais Xen sobre aplicações de computação paralela e distribuída, inclusive em casos de migração de uma máquina virtual para outro hospedeiro físico.

Os resultados obtidos mostraram que a sobrecarga envolvida no uso de Xen foi relativamente baixa nos diferentes casos considerados. Tais resultados indicam que Xen pode constituir uma alternativa viável de virtualização em plataformas de processamento de alto desempenho.

## Referências

- [1] Project JXTA. Disponível em: <http://www.jxta.org/>. Acesso em: julho 2006.
- [2] TOP500 supercomputer sites. Disponível em: <http://www.top500.org/>. Acesso em: julho 2006.
- [3] 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.

- [4] S. Childs, B. Coghlan, D. O'Callaghan, G. Quigley, and J. Walsh. A single-computer grid gateway using virtual machines. In *Proc. 19th International Conference on Advanced Information Networking and Applications (AINA'05)*. IEEE Computer Society, 2005.
- [5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI '05)*, Boston, USA, May 2005. Usenix.
- [6] J. Dike. *User Mode Linux*. Prentice Hall PTR, 2006.
- [7] R. Figueiredo, P. Dinda, and J. Fortes. A case for grid computing on virtual machines. In *Proc. International Conference on Distributed Computing Systems (ICDCS '03)*, 2003.
- [8] R. Goldberg. Survey of virtual machine research. *IEEE Computer*, 7(6):34–45, 1974.
- [9] W. Huang, J. Liu, B. Abali, and D. Panda. A case for high performance computing with virtual machines. *The 20th ACM International Conference on Supercomputing*, 2006.
- [10] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-Peer Computing. Technical Report HPL-2002-57, HP Labs, Palo Alto, USA, 2002.
- [11] M. V. Neves, T. Scheid, A. S. Charão, G. S. Welter, and O. L. L. de Moraes. Análise paralela e distribuída de dados micrometeorológicos utilizando a plataforma JXTA. In *Proc. Workshop of Computational Grids and Applications (WCGA '06)*, 2006.
- [12] A. Petitet, R. C. Whaley, J. Dongarra, and A. Cleary. HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. Disponível em: <http://www.netlib.org/benchmark/hpl/>. Acesso em: julho 2006.
- [13] D. C. Plummer. Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48 bit Ethernet address for transmission on Ethernet hardware. RFC 826 (Standard), Nov. 1982.
- [14] B. Quetier, V. Neri, and F. Cappello. Scalability comparison of 4 host virtualization tools. Technical Report 1433, INRIA/LRI, Université Paris-Sud, 2006.
- [15] J. E. Smith and R. Nair. The architecture of virtual machines. *IEEE Computer*, 38(5):32–38, 2005.
- [16] J. Sugerman, G. Venkitachalam, and B.-H. Lim. Virtualizing I/O devices on VMware workstation's hosted virtual machine monitor. In *Proc. 2001 Usenix Annual Technical Conference*, pages 1–14. Usenix Assoc., 2001.
- [17] A. Whitaker, M. Shaw, and S. D. Gribble. Denali: Lightweight virtual machines for distributed and networked applications. Technical Report 02-02-01, University of Washington, 2002.