Uma análise de recursos virtualizados em ambiente de HPC

Thais C. de Mello[2], Bruno Schulze[1], Raquel C. G. Pinto[2], Antonio R. Mury[1]

[1]Coordenação de Ciência da Computação - Laboratório Nacional de Computação Científica
Av. Getúlio Vargas 333, Quitandinha, Petrópolis, RJ.

[2]Sistemas e Computação - Instituto Militar de Engenharia
Praça General Tiburcio 80, Urca, Rio de Janeiro, RJ

{thaiscm, schulze, aroberto}@lncc.br, raquel@ime.eb.br

Abstract. *How the cloud paradigm can support the solution of High Performance Computing (HPC) problems has not been fully answered yet. This raises a number of questions and, among them, the effect of resources virtualization in terms of performance. We present the analysis of the use of virtual clusters and the factors considered crucial to the adoption of such solution, mainly on infrastructure development to facilitate their creation and use of a custom environment for education, training, testing or development.*

Resumo. *A questão de quanto o uso do paradigma de nuvem pode servir de apoio a solução de problemas de Computação de Alto Desempenho (HPC) ainda não foi totalmente respondida. Suscita uma série de questionamentos e, dentre eles, o efeito da virtualização de recursos em termos de desempenho. Neste artigo é feita uma análise do uso de clusters virtuais e os fatores considerados determinantes para adoção desse tipo de solução, principalmente no desenvolvimento da infraestrutura que facilite sua criação e o uso de um ambiente personalizado para fins educativos, de treinamento, de teste ou de desenvolvimento.*

1. Introdução

O desenvolvimento científico e tecnológico observa uma crescente necessidade e dependência de recursos computacionais de alto desempenho e capacidade de armazenamento. Isso se deve ao fato de que o desenvolvimento das atividades de pesquisas nas diversas áreas da ciência vem gerando um conjunto de dados cada vez maior e mais complexo, tornando o uso da computação de alto desempenho imprescindível para que a análise dos dados, seu armazenamento e compartilhamento sejam feitos de forma rápida e eficiente.

A utilização da computação intensiva, como acima descrito, para obtenção de resultados científicos, caracteriza o termo e-Science. Tal conceito deve permitir o acesso aos recursos computacionais distribuídos, ao armazenamento, ao compartilhamento de dados e resultados, viabilizando o desenvolvimento de pesquisas colaborativas de forma global e a grandes distâncias.

Entretanto, torna-se um grande desafio prover um ambiente de *hardware* e *software* que ofereça a seus usuários uma abstração sobre os recursos computacionais necessários para execução de simulações. Similarmente, prover uma infraestrutura de

gerência dos elementos computacionais utilizados e produzidos durante essas simulações também se mostra um processo complexo. Além disso, o avanço científico exige um investimento crescente em equipamentos e infraestrutura, de forma a estar em sintonia com as necessidades de tal avanço.

Uma alternativa tecnológica adotada pelos centros de e-Science foi o uso de Computação em *Grid*, que permite a utilização de diversos recursos computacionais heterogêneos e dispersos por várias instituições. Essa característica possibilita que seus usuários compartilhem recursos de forma confiável, consistente e transparente. Contudo, a experiência obtida em diversos projetos concebidos a partir da infraestrutura de *Grid* mostrou alguns problemas, tais como instalação, configuração e usabilidade complexas, além da necessidade de controles administrativos intensivos, o que dificultou sobremaneira o aumento dessa infraestrutura em termos de uso e espaço. Ainda assim, o *Grid* permitiu um avanço em termos de computação utilitária no qual a estratégia é baseada no encapsulamento da infraestrutura computacional na forma de serviços.

Nesse ínterim houve o desenvolvimento de novas tecnologias tanto em termos de capacidade de processamento, quanto em capacidade de redes de comunicação, o que possibilitou um avanço no uso mais efetivo e compartilhado dos recursos computacionais, principalmente pelo desenvolvimento de processadores dedicados à virtualização.

Por conseguinte, esse avanço da tecnologia de virtualização possibilitou que, a partir de uma única plataforma física, pudessem coexistir várias plataformas sendo executadas ao mesmo tempo e de forma isolada. Com isso, é possível ter, em um mesmo equipamento físico, vários ambientes virtuais diferentes, cada um com seus respectivos sistemas e aplicações.

A virtualização também torna possível a criação de um ambiente dinâmico e flexível, que aproveita o máximo dos recursos computacionais ociosos por meio de seu compartilhamento, diminuindo a necessidade de adquirir novos equipamentos. Com o reaproveitamento dos recursos já existentes, facilita-se o suporte e a manutenção, permitindo a existência de um número maior de plataformas virtuais sem ter o respectivo aumento no número de plataformas reais.

Esses avanços propiciaram o surgimento de um novo paradigma da computação, chamado de Computação em Nuvem. Tal tecnologia se propõe a ser um sistema de computação distribuída em larga escala, oferecendo um conjunto de recursos virtualizados, dinamicamente escaláveis, gerenciáveis em termos de capacidade de processamento, armazenamento, número de plataformas e serviços. Tem como princípio a oferta de serviços adequados às necessidades do usuário, de maneira a reforçar o conceito de elasticidade sob o ponto de vista da demanda de recursos, cuja alocação e desalocação ocorre em função da necessidade, tornando desnecessária a imobilização de um grande capital para sua aquisição.

A estrutura de computação em nuvem, que passaremos a tratar por *Cloud*, tem como características flexibilidade, escalabilidade e portabilidade. A flexibilidade se dá pelo fato da infraestrutura de TI poder ser adaptada em função do modelo de negócios vigente, sem que estes estejam necessariamente atrelados à infraestrutura passada. A escalabilidade se caracteriza pelo aumento e pela redução de recursos em função das necessidades, possibilitando uma resposta rápida à demanda. A portabilidade, por sua

vez, se dá pela dissociação entre o espaço geográfico e os recursos e as pessoas que os utilizam, permitindo que tais recursos estejam espalhados ou agregados em sistemas especializados, com a consequente redução de custo em função da economia de escala.

O conceito acima descrito tem eco no termo *Virtual Workspace*, que pode ser compreendido como um ambiente de execução configurável e gerenciável de forma a atender aos requisitos dos seus usuários, definindo, para tanto, quais os requisitos de *hardware*, configuração de *software*, isolamento de propriedades e outras características. Com base nesse conceito e no de *Cloud*, o presente artigo se propõe a apresentar a análise de um conjunto de testes que foram realizados para verificar a viabilidade do desenvolvimento de uma infraestrutura cuja essência é facilitar a criação de ambientes de *clusters* virtuais.

O objetivo é possibilitar que seus usuários possam criar e utilizar um ambiente personalizado para fins educativos, treinamento, teste ou desenvolvimento. Para tanto, eles apenas configurarão características básicas, sem que precisem deter conhecimentos técnicos sobre a estrutura, manutenção ou configuração do ambiente. Esse ambiente tem como principais características: a flexibilidade, ao possibilitar que o usuário configure e seja capaz de utilizar várias estruturas de *clusters* em um mesmo conjunto de recursos homogêneos ou não; a escalabilidade, na medida que é possível aumentar ou reduzir o número de nós em função de suas necessidades ou restrições do ambiente; e a portabilidade, ao permitir que o ambiente criado seja armazenado tanto para uso futuro quanto para ser utilizado em outra estrutura disponível, dissociado do espaço físico em que se encontra.

Este trabalho está estruturado como apresentado a seguir: Na seção 2 são apresentados os trabalhos relacionados ao uso de *clusters* virtuais, virtualização de recursos, *Grid* e *Cloud*. Na seção 3 são apresentados os objetivos dos testes realizados, equipamentos, aplicações e a metodologia de análise. Na seção 4 é feita a análise dos resultados e na seção 5 são apresentadas as conclusões.

2. Trabalhos Relacionados

O uso de computação intensiva em apoio à pesquisa e ao processamento de grande quantidade de dados científicos exige utilização colaborativa dos recursos. A infraestrutura para o tratamento desse tipo de informação e as ferramentas para auxiliarem nessa tarefa devem possibilitar que seus usuários (e-scientists) as utilizem de forma transparente. A otimização no uso dos recursos em grandes centros de HPC e sua integração com a infraestrutura de e-Science apresentam desafios, destacando-se a necessidade de algoritmos mais efetivos para o processamento da informação e o uso racional dos recursos, caracterizados pela necessidade de ferramentas e códigos mais eficientes e escaláveis, capazes de melhor aproveitar o crescente aumento do número de núcleos em um único processador [Riedel et al. 2007].

A experiência obtida com o uso do *Grid* serviu de base para um modelo de computação distribuída mais flexível e adequado às necessidades específicas dos usuários. Além disso, permitiu um avanço em termos de computação utilitária e, ao incorporar a arquitetura aberta de serviços de *Grid* (OGSA), motivou o aparecimento de uma nova estratégia baseada no encapsulamento e na oferta de serviços. Essa abordagem reforça o conceito de elasticidade na demanda de recursos computacionais, em que a variação dessa necessidade é acompanhada pela correspondente variação na alocação. Apesar da

tecnologia de Grid poder ser utilizada em uma ampla variedade de aplicações comerciais e de seu sucesso na área acadêmica, sua evolução não alcançou plenamente os benefícios inicialmente pretendidos - a escalabilidade, o fato de ser centrada nos usuários, a facilidade de uso, configuração e gerenciamento [Stanoevska-Slabeva et al. 2009].

Surge assim, como resposta a essas necessidades, o modelo de *Cloud*. Este se caracteriza por ser um modelo econômico e de negócios que apresenta grande flexibilidade de recursos, permitindo que até mesmo as pequenas corporações possam ser capazes de se manterem atualizadas e competir em condição de igualdade com corporações maiores [Stanoevska-Slabeva et al. 2009].

Segundo [Foster et al. 2008] tanto *Grid* quanto *Cloud* procuram reduzir o custo da computação, aumentando a confiabilidade, a flexibilidade e transformando o conceito de computação de "algo" que se compra e que nós operamos em "algo" operado por terceiros. Ambas as infraestruturas têm as mesmas necessidades, tanto em termos de gerenciamento de recursos, quanto pelo fato de permitirem que seus usuários possam utilizá-los em toda a sua capacidade. Contudo, a evolução que o *Cloud* representa está na mudança do foco de uma infraestrutura que fornece recursos computacionais para uma baseada em economia de escala, fornecendo recursos/serviços de forma mais abstrata a seus usuários.

Com base no exposto anteriormente, é de grande importância atentar para o conceito de *Virtual Workspace* apresentado por [Keahey et al. 2005], que consiste em disponibilizar de forma automática recursos e prover o ambiente de execução que atenda às necessidades do usuário. Percebe-se que uma das principais ferramentas dos cenários acima descritos é a virtualização de recursos, que apesar de ser um conceito utilizado há bastante tempo, teve suas funcionalidades ampliadas e seu uso difundido nos últimos anos. Tal fato se deve ao desenvolvimento tecnológico alcançado sobretudo pelos avanços na arquitetura de processadores e pelo aumento da capacidade de rede. No caso dos processadores, houve não só o aumento da capacidade de processamento, mas também a implementação de instruções dedicadas à virtualização de recursos.

Em sintonia com a evolução tecnológica, vários aspectos motivaram e aceleraram o uso da virtualização nos últimos anos, tais como: a capacidade de consolidação de servidores, inclusive auxiliando a continuidade na utilização de sistemas legados; a confiabilidade, padronização e a velocidade na disponibilização de novos recursos/plataformas; a atualização, tanto de *hardware*, quanto de aplicações sem a necessidade de interrupções em serviços e servidores; e a segurança, por permitir a configuração de ambientes de testes, desenvolvimento e treinamento sem o comprometimento direto dos recursos em produção [Dittner and Rule 2007]. Há, no entanto, algumas limitações que ainda persistem, podendo ser citadas: a necessidade de um gerenciamento cuidadoso do nível de comprometimento/carga das máquinas hospedeiras; a correta avaliação e controle do que pode ser executado/transposto para uma máquina virtual; e a mudança de paradigma na administração e organização dos recursos virtualizados.

Em [Mergen et al. 2006], argumenta-se que o uso da virtualização em HPC possibilita tanto a configuração dedicada dos sistemas quanto a manutenção de compatibilidade com sistemas legados, ou seja, a coexistência nos mesmos recursos. A extensão do número de máquinas virtuais permite, por exemplo, que o Monitor de Máquinas Virtuais (VMM) apresente, para um *cluster* virtual, a ideia da existência de inúmeros nós virtuais

em poucos nós reais, permitindo que um teste em escala real de aplicações tipo MPI possa ser feito mesmo na presença de poucos recursos, ainda que estes estejam sendo utilizados por outras aplicações.

Os autores defendem, ainda, que o custo da virtualização em si e o benefício sob o ponto de vista do desempenho que ela realmente oferece são fatores que devem ser levados em consideração. Levanta o questionamento quanto aos requisitos necessários ao desenvolvimento de um VMM específico para o uso em HPC (tamanho das páginas de memória, escalonamento dos recursos virtualizados, mecanismos de DMA e alto desempenho em termos de I/O, balanceamento de carga, preempção e migração).

Em [Evangelinos and Hill 2008] temos a análise de um Provedor/Sistema de *Cloud* (Amazon EC2), utilizado para a configuração de um *cluster* virtual. Os autores avaliaram que é viável esse tipo de solução, principalmente quanto à facilidade de uso e à disponibilização de sistemas que meçam seu impacto nos paradigmas de HPC. Assim sendo, constataram que a solução de recursos "sob-demanda", associados a customização com foco na solução de um problema/cenário específico e gerenciados no momento desejado, é vantajosa mesmo que com perda de desempenho.

[Ranadive et al. 2008] argumentam que a virtualização tem se mostrado uma ferramenta eficaz para o tratamento de falhas em equipamento, aumentando a portabilidade das aplicações e auxiliando a descoberta e correção de falhas em algoritmos complexos. Todavia, pesquisadores mostram-se relutantes em utilizá-la em aplicações de HPC. Apontam, nesse sentido, dois fatores: o primeiro se deve em parte ao desejo de quererem explorar todas as capacidades dos recursos e plataformas; o segundo, considerado o mais importante, é a resistência quanto ao compartilhamento de recursos e seu efeito na degradação do nível de desempenho.

Fatores também apontados por [Gavrilovska et al. 2007] incluem: a determinação efetiva do nível de perda aceitável introduzida pela virtualização; a solução de problemas de inconsistência em relação ao *hardware* ou aplicações e o impacto do uso das plataformas com multiplos núcleos. [Ranadive et al. 2008] conclui que, dependendo do tipo de interação entre as aplicações e o nível de comunicação (I/O) efetuado na solução de aplicações utilizando MPI, mostra-se mais vantajoso associar cada máquina virtual a um núcleo, como componentes separados dentro de uma mesma plataforma.

[Brandt et al. 2009] alerta que, apesar das promessas propagadas pelo uso de *Cloud*, para um ambiente de HPC é necessária a existência de uma gerência que auxilie o usuário tanto na escolha dos recursos a serem alocados quanto na análise a posteriori. Essa análise tem por fim otimizar este ambiente virtual em utilizações futuras, devendo ser capaz de responder a parâmetros adicionais e ajustá-los dinamicamente mesmo em tempo de execução. De forma análoga, ela deve expor os parâmetros de avaliação e essas devem ser realizadas nos recursos virtuais e reais, possibilitando aos provedores de recursos/serviços oferecer um melhor nível de qualidade de serviços a seus usuários.

Nos trabalhos supra-relacionados, em relação à possibilidade do uso de recursos virtualizados para o oferecimento de serviços de alto desempenho aos usuários, destacamos os seguintes itens importantes: a necessidade de que esse ambiente seja flexível, portável e elástico; o desenvolvimento de mecanismo de avaliação dos recursos reais e virtuais para a otimização do seu uso; o impacto do compartilhamento de recursos; o de-

senvolvimento de mecanismos de balanceamento de carga de forma a minimizar a perda de desempenho e o desenvolvimento de ferramentas próprias para auxiliar os usuários na configuração e gerenciamento dos recursos desse ambiente.

Este trabalho insere-se em um projeto em andamento que objetiva verificar a viabilidade do desenvolvimento de uma infraestrutura que facilite a criação de *clusters* virtuais. Pretende-se, assim, possibilitar que os usuários sejam capazes de criar e utilizar um ambiente personalizado, recuperável, escalável e de fácil uso e configuração, seja para fins educativos, de treinamento, seja para teste ou desenvolvimento, de forma a otimizar o uso de equipamentos existentes em um ambiente de Grid, podendo, porém, tal infraestrutura ser aplicada em outros ambientes distribuídos. Na seção a seguir será apresentada a análise de um conjunto de testes que buscam avaliar o impacto do uso da virtualização em relação ao desempenho (carga de processamento e de comunicação) em um ambiente de HPC com um estudo de caso na utilização de *clusters* virtuais.

3. Objetivos e Métodologia de Análise

Nesta seção será apresentada a metodologia de análise utilizada para verificar não só o impacto do uso de *clusters* virtuais em um ambiente distribuído, mas também o efeito da concorrência no uso destes recursos. Os testes a seguir buscam determinar os valores de desempenho em termos de capacidade de processamento (Gflops) e capacidade de comunicação (MBs/s), utilizando dois pacotes de testes de *benchmarks* para sistemas paralelos ou distribuídos.

3.1. Objetivos

Os seguintes objetivos abaixo foram utilizados na estruturação dos testes realizados:

1. Determinar a perda de desempenho de um *cluster* virtual em comparação a um *cluster* real, com o uso do VMM escolhido para o trabalho;
2. Determinar o efeito do tipo de distribuição *Linux* utilizada em conjunto com o VMM; e
3. Determinar o efeito do uso concorrente dos recursos por diversos *clusters* virtuais.

3.2. Infraestrutura de equipamentos

Os experimentos foram realizados em 6 servidores, sendo que cada um contava com as seguintes características: dois processadores Xeon 5520 Quad Core (2.26GHz), com 12 GB de memória RAM DDR3 (1333Mhz), com tecnologia de *Hyperthread* e Instruções de Virtualização habilitadas e Rede Gigabyte.

3.3. Ferramenta de Cluster

O Pelican HPC [Creel 2008] foi utilizado na criação e no gerenciamento tanto do *cluster* real quanto do virtual. Ele tem como principal característica o fato de poder ser executado totalmente em RAMDISK (o SO trata a memória alocada como se fosse um disco rígido, dispensando o uso do disco real), tornando, assim, a montagem do *cluster* mais rápida. Sua instalação ocorre a partir de um *Live-CD* que possui as bibliotecas LAM-MPI e OpenMPI (32 e 64 bits). Os nós de trabalho são carregados via PXE (padrão de *boot* remoto Intel, que utiliza a rede para o carregamento das aplicações necessárias, permitindo, assim, estações sem disco) e usam o nó frontal como mestre.

3.4. Virtualizador

Utilizou-se um sistema de Virtualização Total [SUNMicrosytems 2010] que permite que um SO convidado possa ser executado, sem ser modificado, sobre o SO hospedeiro. Esse sistema de virtualização tem suporte a instruções de virtualização *Hardware-Assisted Intel Virtualization Technology* e suporta, no caso de Plataformas Hospedeiras (32/64 bits), Solaris, Linux, Mac OS X, Windows. Já no caso de Plataformas Convidadas (32/64 bits), *Solaris*, *Linux*, BSD, IBM-OS2 e *Windows*. Além disso, tem suporte a *Open Virtualization Format* (OVF), permitindo a importação e exportação de *aplliances* com outros tipos de virtualizadores e suporte a *Symmetric Multiprocessing* (32 núcleos).

3.5. Pacotes de Teste de Desempenhos

Para medir o desempenho dos *clusters* foram utilizados duas aplicações de *benchmarks*: a primeira com o objetivo de avaliar a capacidade de transmissão de dados (Beff - MB/s) e a segunda avalia a capacidade de processamento (PARPAC - Gflops)[Kredel and Merz 2004].

1. Beff - mede a carga acumulada de comunicação de sistemas paralelos ou distribuídos. Para os testes são utilizados diversos tamanhos de mensagens, padrões de comunicação e métodos. O algoritmo usa uma média que leva em consideração mensagens longas e curtas que são transferidas com diferentes valores de largura de banda em aplicações reais. O cálculo é feito levando-se em conta o número de processos MPI e o tamanho das mensagens. Os padrões são baseados em anéis com distribuições aleatórias. A comunicação é implementada de três diferentes maneiras com o MPI e para cada medida é usada a banda máxima.
2. PARPAC - simula a dinâmica de um fluido (lattice Boltzmann), calculando a permeabilidade da estrutura e tendo como retorno o cálculo de desempenho em Gflops. O código é escrito totalmente em C++ e de forma paralelizada, capaz de fazer a decomposição automática do domínio e o balanceamento de carga, com otimização da comunicação entre os nós calculados.

3.6. Tipos de Testes e Medidas realizadas

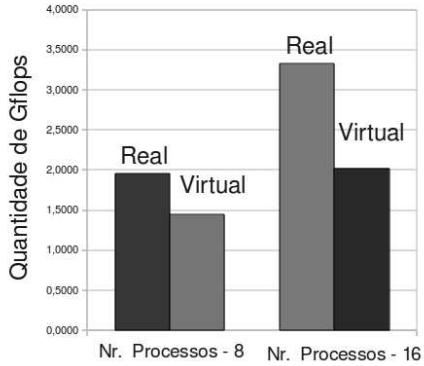
Para o cálculo dos resultados foram tomadas 30 medidas para cada teste, apresentando-se o cálculo de sua média e o intervalo de confiança.

1. Teste 1 - fez-se a medida de desempenho do *cluster* real, sendo este instalado e configurado com um servidor como *frontend* e cinco servidores como nós de trabalho. Esse teste teve como objetivo verificar o desempenho do *cluster* real instalado para servir de comparação com o Teste 2.
2. Teste 2 - inicializou-se uma máquina virtual em cada servidor e foi instalado o *cluster*. Uma máquina virtual foi utilizada como *frontend* e as outras cinco máquinas virtuais como nós de trabalho (cada máquina virtual foi configurada como possuindo 8 núcleos e 3546MB de memória). O teste 2 teve como objetivo medir o desempenho do *cluster* virtual. A opção de se utilizar um *frontend* com a mesma capacidade dos nós de trabalho se deve ao fato que ambos os pacotes de teste podem utilizar o *frontend* para o balanceamento de carga (inclusive esse uso foi notado durante os testes).

3. Teste 3 - 5 máquinas virtuais foram inicializadas em um único servidor (cada máquina virtual com 2 núcleos e 2048 MB de memória), uma máquina como *frontend* e quatro máquinas como nós de trabalho. O objetivo desse teste foi medir o efeito do tipo de distribuição *Linux* no ambiente de *cluster* virtual, ou seja, o impacto que o conjunto de bibliotecas que caracterizam uma distribuição tem em relação ao virtualizador, assim como a comparação entre o efeito da arquitetura de 32 Bits e de 64 Bits.
4. Teste 4 - criaram-se 3 *clusters* virtuais em um único servidor, cada um com 3 máquinas virtuais (2 núcleos e 1024 MB de memória), sendo uma como *frontend* e as outras duas como nós de trabalho. Cada *cluster* foi inicializado separadamente para verificar o impacto no desempenho pela concorrência nos recursos. O teste 4 teve como objetivo ver o efeito da concorrência no caso de existirem mais de um *cluster* sendo executado em um mesmo servidor.

4. Análise dos Resultados

O testes 1 e 2 foram realizados com a finalidade de comparar o desempenho do *cluster* real com o *cluster* virtual. Para a realização dos testes foi utilizado a distribuição *Linux* 64 Bits padrão do Laboratório de Computação Científica Distribuída - ComCiDis - LNCC. As medidas obtidas encontram-se apresentadas nas figuras 1 e 2. A Figura 1 apresenta o resultado em relação à carga de processamento e podemos notar que, no caso de 8 processos, o *cluster* virtual teve um desempenho de 73,7% em comparação com o *cluster* real. Para o caso de 16 processos, o desempenho caiu para 60,8%. A tabela da figura apresenta os valores obtidos tanto para o *cluster* real quanto para o virtual em termos de Gflops, com um intervalo de confiança de 95%.



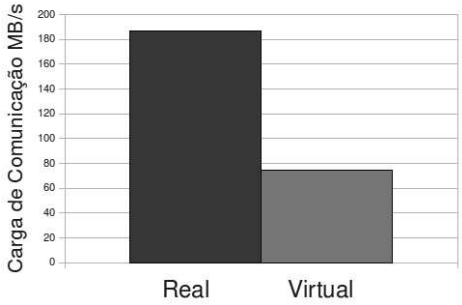
	Real 8 Processos	Virtual 8 Processos	Real 16 Processos	Virtual 16 Processos
Média	1,956	1,442	3,323	2,023
Intervalo Inferior	1,954	1,441	3,321	2,022
Intervalo Superior	1,959	1,443	3,326	2,024

Figura 1. Desempenho da capacidade de processamento entre o cluster real e o cluster virtual

Para o teste de carga de comunicação apresentado na (Figura 2) o desempenho alcançado pelo *cluster* virtual em relação ao *cluster* real foi de 39,8%. Este baixo desempenho tem como consequência o uso da virtualização total. A fim de permitir uma menor dependência da máquina hospedeira, aumentando a portabilidade, é criada uma camada de dispositivos I/O virtuais. No caso específico desse teste, é necessário que o sistema convidado (máquina virtual) escreva os dados na parte da memória virtual alocada para tanto. Porém, ao tentar fazê-lo, o VMM intercepta a tentativa do sistema convidado e

mapeia para o local de memória real, onde está mapeado o dispositivo de I/O real. O VMM tem que manter um controle e mapear todos os endereços de memória dos dispositivos virtuais e encaminhá-los para os dispositivos reais correspondentes. Tal fato causa uma sobrecarga que diminui significativamente o desempenho.

A solução atualmente encontrada para diminuir esta perda é o uso de bibliotecas "Virtio". Essas surgem como alternativas à proposta adotada pelo XEN na tentativa de fornecer uma série de *drivers* padrões para distribuições *Linux*, sendo que tais drivers são adaptáveis a vários tipos de VMM, não adicionando sobrecarga. O primeiro a usar esse tipo de solução foi o KVM [Russell 2008], [Nussbaum et al. 2009] e, apesar de o virtualizador utilizado neste trabalho já possuir suporte a estes tipos de biblioteca na sua última versão, não foi objeto deste trabalho, sendo programado para trabalhos futuros.



Carga de Comunicação MB/s

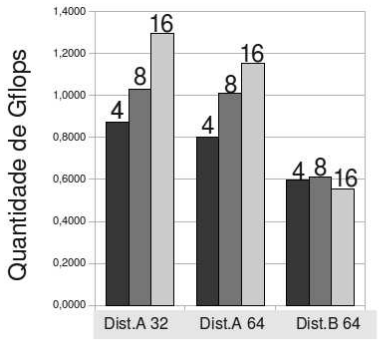
Real	Virtual
187,19	74,64
186,47	72,64
188,11	76,65

Figura 2. Desempenho carga de comunicação do cluster real com o cluster virtual

O teste 3 teve como objetivo ver o impacto do tipo de distribuição *Linux* no desempenho dos *clusters* virtuais. A Figura 3 apresenta a comparação do desempenho entre *clusters* virtuais utilizando como SO básico duas distribuições *Linux* diferentes, porém, com o mesmo pacote básico (chamaremos de distribuição A e B). Fez-se ainda uma comparação dentro da mesma distribuição utilizando SO de 32 bits e SO de 64 bits (Distribuição A32 e A64). A Figura 3 nos mostra que, à medida que a carga de processos foi aumentando na distribuição A32, sua capacidade de processamento também aumentou (de 0,87 Gflops com 4 processos a 1,3 Gflops com 16 processos).

A mesma distribuição no caso de 64 bits (A64) apresentou similarmente uma resposta positiva (de 0,80 Gflops com 4 processos para 1,2 Gflops com 16 processos). Percebe-se que a diferença entre ambas é praticamente nula, sendo que o motivo da diferença em favor da distribuição A32 se deve ao fato da distribuição A64 conter bibliotecas de 32 Bits, o que diminui ligeiramente o seu desempenho. Porém, a distribuição B64 apresentou um desempenho de 74,5% em relação à distribuição A64 com 4 processos, caindo para 48% com 16 processos. É importante notar que sua capacidade de processamento inicialmente subiu na mudança de 4 para 8 processos (0,60 Gflops para 0,61 Gflops - Figura 3 Distribuição B 64), porém caiu quando submetida a 16 processos (0,55 Gflops).

O motivo para o baixo desempenho em relação a outra distribuição e a queda, no caso de 16 processos, está relacionada à incapacidade que se notou em se distribuir igualmente o processamento das máquinas virtuais entre todos os núcleos da máquina real. A Figura 4 apresenta uma amostra da carga entre os núcleos da Distribuição



		Distribuição A32	Distribuição A64	Distribuição B64
Média	4 processos	0,8712	0,8032	0,5980
	8 Processos	1,0287	1,0113	0,6121
	16 Processos	1,2977	1,1550	0,5548
Limite Inferior	4 processos	0,8711	0,8031	0,5975
	8 Processos	1,0281	1,0113	0,6121
	16 Processos	1,2971	1,1550	0,5544
Limite Superior	4 processos	0,8713	0,8033	0,5984
	8 Processos	1,0292	1,0113	0,6122
	16 Processos	1,2982	1,1550	0,5555

Figura 3. Desempenho de distribuições Linux - capacidade de processamento

A64 e da Distribuição B64. No caso da capacidade de processamento esta diferença se deve à política de distribuição de recursos de cada uma delas, que em função do uso poderá se mostrar vantajosa ou não, neste caso a liberdade na distribuição de carga pelos núcleos/threads, beneficiou a distribuição A64.

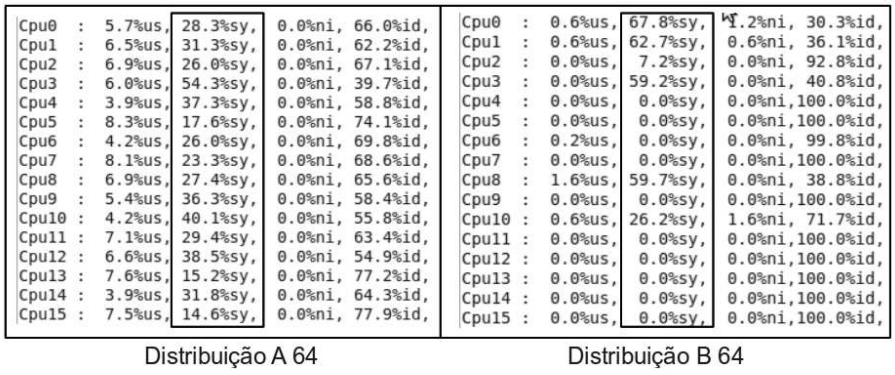
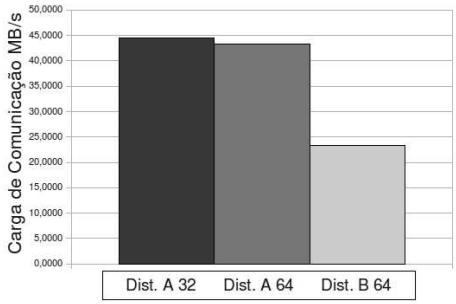


Figura 4. Desempenho de diferentes distribuições Linux - processamento

O mesmo pode ser observado no caso do teste de carga de comunicação apresentado na Figura 5 em que a distribuição A32 ficou cerca de 3% acima da distribuição A64 e a distribuição B64 ficou cerca de 47% abaixo da distribuição A32.

A Figura 6 apresenta os testes realizados para a medida de desempenho com a concorrência de *clusters* virtuais em um mesmo servidor. Foram realizados testes para 1 (utilizado como referência), 2 e 3 *clusters* concorrentes. No caso de 4 processos houve uma queda de cerca de 16% ao ser adicionado um segundo *cluster*, sendo que a perda se acentuou para 24% ao ser adicionado um terceiro *cluster*.

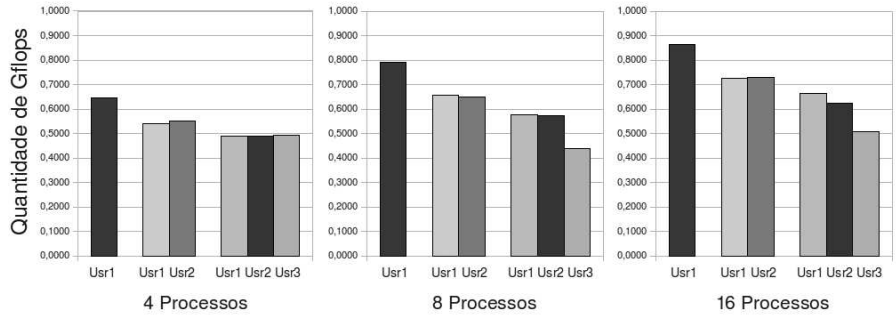
No caso de 8 processos, a perda para um *cluster* concorrente aumentou de apenas 1% em relação ao caso anterior de 4 processos (17%). Para 2 *clusters* concorrentes a perda foi de 44,5%, ou seja, quase o dobro em relação ao caso anterior. É possível notar que o desempenho do *cluster* do usuário 3 foi cerca de 24% menor em relação aos outros dois usuários. Isso se deve ao fato de o terceiro usuário ter sido inicializado por último encontrando, assim, o processamento do servidor já em uma situação de carga próxima de seu limite. Nessa situação, tanto o SO quanto o VMM não foram capazes de efetuar o balanceamento correto da carga.



	Distribuição A 32	Distribuição A 64	Distribuição B64
Média	44,4813	43,2703	23,3580
Limite Inferior	44,2852	43,0881	23,3102
Limite Superior	44,6775	43,4526	23,4058

Figura 5. Desempenho de diferentes distribuições Linux - carga de comunicação

No caso de 16 processos fica claro o efeito da concorrência sobre o desempenho dos *clusters*. Na adição de um segundo *cluster* o resultado se igualou aos anteriores, devido ao fato de haver ainda recursos no servidor capazes de absorver a carga. Porém, a adição do terceiro *cluster* acarretou na perda de cerca de 4,5% do usuário 2 em relação ao usuário 1, de 17,8% do usuário 3 em relação ao usuário 1 e de 13,3% do usuário 3 em relação ao usuário 2.



	Usuário 1		Usuário 1	Usuário 2		Usuário 1	Usuário 2	Usuário 3
4 processos	0,6449		0,5412	0,5510		0,4882	0,4908	0,4929
8 Processos	0,7920		0,6550	0,6511		0,5766	0,5750	0,4393
16 Processos	0,8652		0,7268	0,7311		0,6628	0,6238	0,5090

Figura 6. Desempenho com concorrência de usuários

Constata-se, pelos resultados obtidos, que há uma perda de desempenho, como já se esperava, do *cluster* virtual em relação ao *cluster* real, tornando-se acentuada no caso da medida de carga de comunicação, pelos motivos já citados. Essa perda pode ser reduzida pelo uso de paravirtualizadores, sob pena de redução na portabilidade. Pode ser mitigada, também, por meio do desenvolvimento e uso de bibliotecas que minimizem o efeito da perda com dispositivos de I/O, mas que ainda garantam a portabilidade em um determinado nível. No caso aqui apresentado o objetivo do projeto no qual este trabalho se insere é priorizar a portabilidade e a possibilidade de uso de um número maior de SO hospedeiros, mesmo que com um nível de desempenho menor.

Os testes tiveram como mérito a determinação do valor da perda que se observa no caso dessa escolha. A possibilidade do uso de um número maior de plataformas reais como hospedeiras, associada à coexistência de ambientes virtuais com o ambiente real,

torna-se atraente, ainda que haja perda de desempenho. Há de se enfatizar o fato de tal possibilidade ser utilizada em testes, ensino ou desenvolvimento e seus desdobramentos, sem a necessidade de dedicação exclusiva ou mudança da plataforma hospedeira.

Tais testes também mostraram a importância e o cuidado que se deve ter quando da escolha do tipo de ambiente de virtualização. Apesar de as diferenças entre paravirtualização, virtualização total e virtualização assistida por *hardware* já serem conhecidas, em função dos resultados apresentados é importante observar que outros fatores devem ser levados em consideração. A exemplo da escolha do *Kernel* (no caso das distribuições *Linux*, que são de código aberto e auditáveis), que se torna mais crítica nos casos de ambientes de virtualização proprietários. Além disso, há, no caso do *Linux*, o cuidado na escolha da distribuição pelo fato de serem utilizadas bibliotecas de 32 bits em ambientes de 64 Bits, o que pode contribuir negativamente para o desempenho [Bookman 2002].

Vale salientar que no caso das distribuições *Linux*, o *Kernel* vem sendo aprimorado de forma a melhor administrar a comunicação entre os dispositivos de entrada e saída e as máquinas virtuais. Além de um melhor gerenciamento de memória, destaca-se que, a partir da versão 2.6.32 do *Kernel* do *Linux*, foi introduzido o KSM (Kernel Samepage Merging)[REDHAT 2010]. Tal avanço consiste na varredura da memória de cada máquina virtual e, onde houver páginas idênticas de memória, estas são consolidadas em uma única página, que é então compartilhada entre todas as máquinas virtuais, o que aumenta tanto o desempenho quanto o número de máquinas virtuais que poderão ser hospedadas em uma única máquina real.

Em relação ao efeito da concorrência, foi possível notar que, enquanto havia recursos, houve balanceamento no uso dos núcleos do servidor, ainda que com certo nível de perda de desempenho. Com o aumento do número de processos - com 2 usuários, no caso de 4 processos, obteve-se 0,5412 Gflops, e, com 16 processos, 0,7268 Gflops - houve, inclusive, o aumento da capacidade de processamento. Todavia, quando a capacidade de recursos da servidora chegou ao seu limite e o balanceamento não foi possível, observou-se uma diferenciação da capacidade de processamento entre os usuários (para 3 usuários 4 processos praticamente não houve diferença, para 8 processos o usuário 3 se diferenciou e para 16 processos todos foram diferentes - Figura 6). Percebeu-se, ainda, durante os testes, o papel desempenhado pelo SO/VMM da máquina hospedeira na distribuição da carga das diversas máquinas virtuais e como isso contribuiu para o aumento do desempenho. Outro aspecto a ser considerado é a criação de um processo de escalonamento que monitorasse os recursos e, em caso de um aumento de carga de trabalho, e desde que haja recursos ociosos, rescalonasse a MV do *cluster* para tais recursos utilizando o que se conhece como migração em tempo real.

5. Conclusões

O estudo acima levantou um conjunto de questionamentos relacionados ao uso do ambiente de *Cloud* e mais especificamente sobre o custo da virtualização em HPC. Utilizamos como exemplo uma série de testes em que foram comparadas configurações e desempenho de *clusters* virtuais em relação a um *cluster* real, além de levantar alguns dos fatores que podem influenciar neste desempenho e que, por conseguinte, devem ser observados.

A escolha tanto do virtualizador como do tipo da aplicação que será executada

neste ambiente devem ser submetidos a uma análise criteriosa: itens como a infraestrutura de *hardware*, passando pela escolha do SO/VMM, inclusive com detalhamento no nível do *Kernel* e módulos, até a aplicação virtualizada propriamente dita. Tal cuidado não eliminará o custo da virtualização, mas este será minimizado.

O uso de *clusters* na forma de *appliances* apresenta grandes possibilidades, principalmente se for analisado não somente sob o aspecto do desempenho, mas também sob a ótica do desenvolvimento de testes e principalmente para o aprendizado e disseminação do uso da computação paralela na solução de problemas complexos. A capacidade de compartilhamento e a segmentação do uso dos recursos, mesmo com perda de desempenho, podem ser vistas como atraente a partir do momento em que estes recursos, por vezes ociosos, poderiam ser efetivamente utilizados. Há ainda vantagem se for levado em conta que o tempo necessário para a criação deste ambiente virtual é menor que o necessário para um ambiente real, além de apresentar um menor número de problemas relacionados a incompatibilidade em termos de *hardware*.

O estudo aqui apresentado, conquanto levante vários questionamentos pertinentes à lógica do uso de virtualização em ambientes de HPC, não esgota o assunto. Na realidade, o presente trabalho levantou, tanto durante os testes quanto ao final, uma série de questões ainda a serem respondidas. Concordamos que para o uso da virtualização em um ambiente desse nível são necessários mecanismos para a coleta de informações, sobretudo de desempenho e de carga, que não apenas permitam a transparência para seus usuários, mas que venham a fornecer subsídios para a melhoria dos serviços. Entretanto, sob o ponto de vista do seu uso para teste, ensino e desenvolvimento, em que pese a perda de desempenho, apontada nesse artigo, se mostra como uma ferramenta eficaz e que merece ser estudada, sobretudo ao poder utilizar de recursos ociosos e não dedicados.

Dentre os questionamentos levantados e que abrem novos cenários de testes, destacam-se: como implementar mecanismos de balanceamento que, em conjunto com o uso de migração em tempo real, possam minimizar o efeito da concorrência entre usuários; como os mecanismos de recuperação de falhas existentes nos virtualizadores podem auxiliar os usuários aumentando a confiabilidade na execução de suas aplicações; como se dá o relacionamento entre o uso de uma determinada aplicação em um ambiente virtual e a associação deste a um ou mais núcleos; e, finalmente, a contribuição do SO/VMM no balanceamento de carga.

O estudo até o momento realizado permite, no entanto, vislumbrar que, com o aperfeiçoamento das técnicas de virtualização, seja no nível do *hardware*, por meio da incorporação de instruções dedicadas no processador e aumento da complexidade do gerenciador de memória, seja no aperfeiçoamento do *Kernel* dos SO das máquinas hospedeiras ou por meio do desenvolvimento de aplicações dedicadas, será possível, em um futuro próximo, diminuir ainda mais as limitações existentes em termos de desempenho, permitindo difundir sua utilização em ambiente de HPC.

Referências

Bookman, C. (2002). *Linux Clustering: Building and Maintaining Linux Clusters*. Sams.

Brandt, J., Gentile, A., Mayo, J., Pebay, P., Roe, D., Thompson, D., and Wong, M. (2009). Resource monitoring and management with ovis to enable hpc in cloud computing

environments. In *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, Washington, DC, USA. IEEE Computer Society.

Creel, M. (2008). Pelicanhpc tutorial. UFAE and IAE Working Papers 749.08, Unitat de Fonaments de l'Analisi Economica (UAB) and Institut d'Analisi Economica (CSIC).

Dittner, R. and Rule, D. (2007). *The Best Damn Server Virtualization Book Period: Including Vmware, Xen, and Microsoft Virtual Server*. Syngress Publishing.

Evangelinos, C. and Hill, C. N. (2008). Cloud computing for parallel scientific hpc applications: Feasibility of running coupled atmosphere-ocean climate models on amazon's ec2. Cloud Computing and Its Applications.

Foster, I., Zhao, Y., Raicu, I., and Lu, S. (2008). Cloud computing and grid computing 360-degree compared. In *2008 Grid Computing Environments Workshop*. IEEE.

Gavrilovska, A., Kumar, S., Raj, H., Schwan, K., Gupta, V., Nathuji, R., Niranjan, R., Ranadive, A., and Saraiya, P. (2007). Abstract high-performance hypervisor architectures: Virtualization in hpc systems. In *HPCVirt '07: Proceedings of the 1st Workshop on System-level Virtualization for High Performance Computing*.

Keahey, K., Foster, I., Freeman, T., and Zhang, X. (2005). Virtual workspaces: Achieving quality of service and quality of life in the grid. *Sci. Program.*, 13(4):265–275.

Kredel, H. and Merz, M. (2004). The design of the ipacs distributed software architecture. In *ISICT '04: Proceedings of the 2004 international symposium on Information and communication technologies*, pages 14–19. Trinity College Dublin.

Mergen, M. F., Uhlig, V., Krieger, O., and Xenidis, J. (2006). Virtualization for high-performance computing. *SIGOPS Oper. Syst. Rev.*, 40(2):8–11.

Nussbaum, L., Anhalt, F., Mornard, O., and Gelas, J.-P. (2009). Linux-based virtualization for HPC clusters. In *Montreal Linux Symposium*, Montreal Canada.

Ranadive, A., Kesavan, M., Gavrilovska, A., and Schwan, K. (2008). Performance implications of virtualizing multicore cluster machines. In *HPCVirt '08: Proceedings of the 2nd workshop on System-level virtualization for high performance computing*, pages 1–8, New York, NY, USA. ACM.

REDHAT (2010). Kernel samepage merging. Technical report, KVM-Kernel Based Virtual Machine.

Riedel, M., Eickermann, T., Habbinga, S., Frings, W., Gibbon, P., Mallmann, D., Wolf, F., Streit, A., Lippert, T., Schiffmann, W., Ernst, A., Spurzem, R., and Nagel, W. E. (2007). Computational steering and online visualization of scientific applications on large-scale hpc systems within e-science infrastructures. In *eScience*, pages 483–490. IEEE Computer Society.

Russell, R. (2008). virtio: towards a de-facto standard for virtual i/o devices. *SIGOPS Oper. Syst. Rev.*, 42(5):95–103.

Stanoevska-Slabeva, K., Wozniak, T., and Ristol, S. (2009). *Grid and Cloud Computing: A Business Perspective on Technology and Applications*. Springer, 1 edition.

SUNMicrosytems (2010). Virtualbox. Technical report, SUN Microsytems.