

## Comparativo de desempenho de um cluster virtualizado em relação a um cluster convencional sob condições equipotentes

David Willians S.C Beserra<sup>1</sup>, Samuel Carlos Romeiro Azevedo Souto<sup>1</sup>, Mariel José Pimentel de Andrade<sup>1</sup>, Alberto Einstein Pereira de Araujo<sup>1</sup>

<sup>1</sup>Unidade Acadêmica de Garanhuns – Universidade Federal Rural de Pernambuco (UFRPE)  
Garanhuns – PE – Brazil

david.beserra@linuxmail.org, samuel.romeiro@gmail.com,  
{mariel,aepa}@uag.ufrpe.br

**Abstract.** *In this work we presents a comparative study of the performance of a cluster of real machines relative to a cluster of virtual machines. Both structures have theoretical equality in your hardware and software configurations. The work was accomplished to evaluate if the structures present similar performance in similar conditions. Our goal is verify the possibility of implement a Beowulf cluster through virtual structures in work stations which are underused, using the maximum of its capacity of processing.*

**Resumo.** *Neste trabalho nós apresentamos um estudo que compara um cluster de máquinas reais com um cluster de máquinas virtuais. Ambas as estruturas têm igualdade teórica em configurações de hardware e de software. O trabalho foi realizado para avaliar se as estruturas apresentam desempenho semelhante em condições semelhantes. Nossa meta é verificar a possibilidade do uso de um cluster Beowulf por estruturas virtuais em estações de trabalho que são subutilizadas, usando o máximo de sua capacidade de processamento.*

### 1. Introdução e Trabalhos Relacionados

Na atualidade, diversos setores da atividade humana têm demandado por elevado poder de processamento e armazenamento de dados. Embora setores econômicos e governamentais demandem cada vez mais por tais recursos, ainda é na ciência em que esse tipo de demanda se concentra. Os Supercomputadores foram à primeira iniciativa para prover alto poder de computação, todavia são onerosos e pouco escalonáveis. Como alternativa aos Supercomputadores convencionais surgiu o *Cluster Beowulf*, que provê computação de alto desempenho (HPC) a baixo custo já que utiliza commodities como componentes [Sterling et al. 1995]. Um *Cluster Beowulf* é constituído por um agrupamento de máquinas (nodos) interligadas por uma rede, tendo um nodo a função de gerente do ambiente (*frontend*) e os demais sendo apenas escravos que executam instruções provenientes do *frontend*.

Alem da necessidade da concepção de arquiteturas de computadores direcionadas para HPC, existe à necessidade de se aproveitar melhor os recursos físicos já disponíveis. Uma maneira encontrada para se prover essa otimização surgiu com o advento da virtualização, que é uma tecnologia que permite executar mais de um

Sistema Operacional (SO) em um mesmo hardware [National Instruments 2010]. Existem basicamente duas formas de realizar a virtualização: A Virtualização Total e a Paravirtualização. Atualmente em alguns ambientes como laboratórios de informática com fins educacionais ou mesmo ambientes de escritórios, a capacidade de processamento de máquinas atuais é subaproveitada, surge então um questionamento a respeito da sua utilização em HPC através da virtualização.

Na Virtualização Total, o SO convidado não sabe que está sendo executado em um ambiente virtualizado, o que também é denotado por Máquina Assistida por Hardware Virtual (HVM). A virtualização completa exige que o Gerenciador de Máquinas Virtuais (VMM) intercepte muitas operações que um SO costuma executar diretamente no hardware. A vantagem desse método é a garantia de que o convidado não interferirá no sistema hospedeiro, lendo ou modificando a memória alocada para ele, nem desligando a CPU que está utilizando [Zhao et al. 2009]. Na Paravirtualização, um SO convidado está ciente da camada de virtualização e modifica-se para apoiá-la, resultando em maior desempenho do sistema virtual. O SO convidado paravirtualizado é portado para rodar em cima do VMM, usando rede virtual, disco e dispositivos de console. Ele deve trabalhar colaborativamente com a camada do VMM. Domínios do cliente podem ser parcialmente ou totalmente virtualizados, e um sistema pode ter os dois tipos de execução simultaneamente [Zhao et al. 2009].

Logo, uma alternativa para economia de custos e amplificação de desempenho para HPC é a virtualização de *Clusters Beowulf*, que podem conviver dentro de outras infraestruturas de computação, inclusive na nuvem, tendo a implementação de um destes descrita por [Ivica et al. 2009]. Todavia, segundo [Mello et al. 2010] uma questão importante que não pode ser ignorada é o impacto da virtualização sobre o ambiente de HPC. Uma forma de medir o desempenho de um *Cluster Beowulf* é mediante o uso de ferramentas de benchmarking [Silva et al. 2009], que podem ser também aplicadas em seus correspondentes virtuais e são de grande auxílio no estudo dos impactos da virtualização no desempenho.

Em [Huang et al. 2006] temos um benchmarking de um *cluster* virtual realizado com enfoque no desempenho de rede e de bibliotecas de paralelização de código, onde o SO nativo e o virtualizado concorrem por todos os recursos disponíveis, obtendo bom desempenho, mas com perdas devido ao compartilhamento de recursos de rede, já [Ranadive et al. 2008] verifica os efeitos da partilha de recursos sobre o desempenho de aplicações de HPC. Especificamente, para várias máquinas virtuais (VMs) em execução em plataformas *multicore*, foi avaliada a extensão em que suas comunicações são afetadas pelo fato de que eles compartilham um único recurso de comunicação, utilizando uma interconexão *Infiniband* como o exemplo concreto de tal recurso. Os resultados experimentais apresentados demonstram que um alto nível de compartilhamento, ou seja, um número significativo de máquinas virtuais implantadas para cada nó é viável sem degradação de desempenho notável.

Também foram realizados estudos para verificar se um sistema para HPC virtualizado e tendo como infraestrutura hospedeira à nuvem poderia fazer parte do ranking dos 500 mais poderosos sistemas computacionais existentes, o TOP 500 [Napper and Bientinesiy 2009]. Eles indicaram que ainda não existe maturidade suficiente na tecnologia de nuvem que permita que ela seja usada em aplicações HPC

com boa relação custo-benefício (RCB), sendo muito ineficientes em relação a sistemas HPC tradicionais, mas que, com o avanço das tecnologias de interconexão de computadores, este cenário pode ser alterado e em um futuro próximo poderão ser ofertados serviços HPC na infraestrutura da nuvem com qualidade comparável aos tradicionais e com RCB aceitável [Napper and Bientinesiy 2009].

A proposta deste trabalho é verificar a viabilidade da implementação de um *Cluster Beowulf* dentro de outra infraestrutura, observando a ocorrência de impactos no desempenho da infraestrutura hospedeira e se existem diferenças notáveis de desempenho entre um *cluster* real e um virtual sobre condições equiparáveis. O restante do trabalho está subdividido pela convenção a seguir: Na próxima seção são expostos os objetivos e a metodologia de análise. Na seção subsequente tem-se a exposição e análise dos resultados e na seção 4 são dispostas as conclusões e os trabalhos futuros.

## 2. Objetivos e Metodologia de Análise

Os testes realizados tentam determinar valores de desempenho de processamento sustentado em termos de Bilhões de Operações de Ponto Flutuante por Segundo (GFLOPS) e vazão média em Mega Bits por Segundo (Mbps), bem como a vazão média em função do tamanho do pacote de dados utilizado, com tamanho especificado em Bytes. Para a realização dos testes de desempenho de processamento sustentado foi o *High Performance Linpack* (HPL), já para os testes de capacidade de comunicação foi empregado o NetPIPE (Network Protocol Independent Performance Evaluator).

### 2.1. Objetivos

Os testes realizados foram estruturados de acordo com os seguintes objetivos:

1. Verificar a diferença de desempenho de um *cluster* de máquinas virtuais em relação a um *cluster* de máquinas reais com configurações de hardware e software similares.
2. Verificar o impacto da execução de uma máquina virtual em capacidade máxima sobre o desempenho do SO da máquina hospedeira, observando a percentagem de uso de CPU pelo hospedeiro no momento da execução da VM.

### 2.2. Infraestrutura de Equipamentos e Opções Arquiteturais

Neste experimento foram empregados 7 computadores padrão IBM-PC, com arquitetura de processador do tipo x86, sendo 3 destes equipados com processador Q8200 e 4 com processador E6550. Mais detalhes sobre a configuração individual de cada computador são apresentados na Tabela 1.

**Tabela 1. Configuração dos computadores utilizados**

Processador	Qtde. núcleos	Frequência	Memória	Frequência
E6550	2	2.33 GHz	1 GB	667 MHz
Q8200	4	2.33 GHz	2 GB	667 MHz

## 2.2.1. Cluster de Máquinas Reais

### 2.2.1.1. Nodos computacionais

O *cluster* com máquinas reais foi implementado com o uso de quatro computadores, todos com processador E6550 e com a quantidade de 1 GB para a memória principal.

### 2.2.1.2. Rede de interconexão

Para a interconexão entre o *frontend* e os nos escravos foi utilizada uma rede *Fast Ethernet*, de alta latência e baixo custo, com banda de 100 Mbps. Todos os nós possuem uma placa de rede com chip RTL8139 e estão interconectados por um roteador Intelbras WRG 240 E.

## 2.2.2. Cluster de Máquinas Virtuais

### 2.2.2.1. Nodos Computacionais

Neste *cluster*, o *frontend* também é um computador real com processador E6550. Já os escravos são máquinas virtuais instaladas em 3 computadores reais equipados com o processador Q8200, pertencentes a um laboratório de ensino. Cada máquina real utilizada (máquina hospedeira) tem como SO nativo o *Windows Vista Ultimate* de 32 bits.

Em cada máquina hospedeira foi instalada uma máquina virtual configurada com duas *vCPUs* (processadores virtuais) e 1 GB de memória principal alocada para si. Observe que cada máquina virtual possui configuração teórica similar as máquinas constituintes do *cluster* real, possuindo a mesma quantidade de núcleos de processamento e memória principal, onde cada núcleo e a memória possuem também a mesma frequência que a do *cluster* real.

Os núcleos e a memória restante ficam para o SO das máquinas hospedeiras e seus aplicativos e, embora o SO do hospedeiro continue a enxergar os núcleos alocados para a VM, não os utiliza (o escopo de visualização dos núcleos nas máquinas hospedeiras é apresentado graficamente na Figura 1). Todavia, se uma máquina virtual estiver ociosa, os recursos alocados a ela podem ser gradativamente liberados para o uso pelo SO nativo da máquina real.

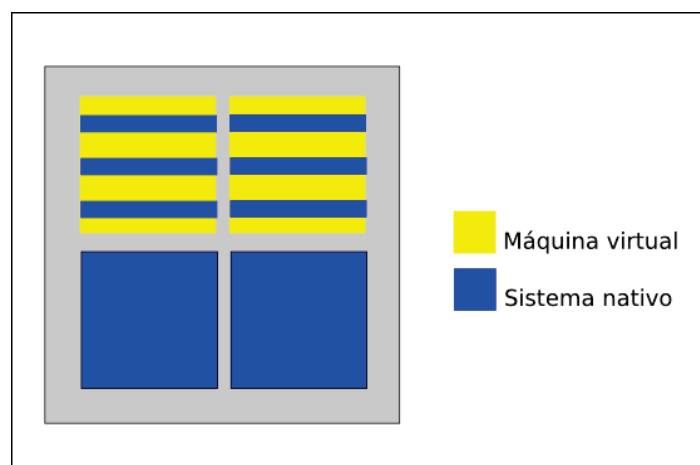


Figura 1. Escopo de visualização dos núcleos do processador

### 2.2.2.2. Rede de interconexão

Para assegurar a igualdade de condições entre os dois *clusters*, os mesmos padrões e equipamentos de rede utilizados no *cluster* real foram empregados no *cluster* de máquinas virtuais.

Vale salientar que, como as máquinas hospedeiras possuem apenas uma placa de rede, o seu SO nativo e a máquina virtual, (também através de seu SO), acabam concorrendo pelos recursos de rede. Para evitar que isso ocorra e continuar mantendo a igualdade teórica de configurações entre as duas plataformas de *cluster*, não foi configurada uma rede local *Windows* entre as máquinas hospedeiras, existindo apenas a rede do *cluster* virtual, que não é enxergada pelo SO nativo dos hospedeiros.

### 2.3. Ferramenta de Cluster e Monitoração

O sistema operacional *Rocks Cluster 5.4* [NSF 2011], em sua versão de 32 bits, foi o escolhido para a implementação de ambos os clusters. Sua principal meta é auxiliar na implementação rápida de *Clusters Beowulf*, possuindo assim um processo de instalação simplificado para *frontend* e escravos. Uma vez instalado o *frontend*, os escravos podem ser adicionados mediante um simples comando de terminal.

A instalação do *frontend* é feita com o DVD do *Rocks* e a dos escravos via PXE (boot remoto via rede). Ao contrario de outras ferramentas de *cluster*, como o *PelicanHPC* [Creel 2011], o *Rocks* necessita que as máquinas hospedeiras do sistema possuam disco rígido, o que, embora consuma mais espaço, torna o sistema mais seguro e estável, em contrapartida a sistemas RAMDISK (que tratam a memória alocada para seu funcionamento como um disco), que são mais instáveis, tendo como exemplo o supracitado *PelicanHPC*.

Para o gerenciamento de carga de processamento dos nós, fluxo de rede e outras atribuições importantes, foi utilizado o visualizador gráfico de recursos de *cluster* *Ganglia*, disponível no *Rocks* como um de seus *Rolls* (pacotes do *Rocks*). Ele é automaticamente instalado quando selecionado no processo de instalação do *frontend*.

### 2.4. Ferramenta de Virtualização

Para a configuração e criação das máquinas virtuais empregadas no experimento foi utilizado o virtualizador proprietário *VMware Workstation* [VMware 2011]. Sua escolha foi motivada pelo fato de outros virtualizadores disponíveis (notadamente o *VirtualBox* e o *Virtual PC*) para plataforma *Windows* não permitirem a criação de máquinas virtuais *multicore*, exceto se o hardware dispor de instruções específicas para virtualização. O *VMware Workstation* é um sistema de Virtualização Total, muito embora disponibilize como opção a habilitação de um conjunto de instruções de Paravirtualização, que não foram empregadas.

### 2.5. Ferramentas de Benchmarking

#### 2.5.1. Desempenho de transmissão de dados

O NetPIPE é um avaliador de desempenho para uma rede local e possui total independência do tipo de protocolo empregado na rede, permitindo monitorar a sobrecarga (overhead) proveniente de camadas protocolares superiores. Ele permite

medir o desempenho com maior profundidade de diversas tecnologias de rede, transferindo tamanhos de blocos que podem ser posteriormente analisados com maior detalhamento [Pitanga 2008]. Como pode ser utilizado com diferentes protocolos, basta especificar o protocolo durante o processo de compilação. Neste trabalho, o protocolo escolhido foi o MPI (Interface de Passagem de Mensagens).

Em sua execução, o NetPIPE gera por padrão um arquivo nomeado np.out que armazena os resultados obtidos. O arquivo contém 3 colunas: o número de bytes por pacote, a vazão (throughput) em Mbps e o tempo de ida e volta das mensagens de teste dividido por 2. As duas primeiras colunas são empregadas para a obtenção de um gráfico da vazão pelo tamanho do pacote e a segunda para se obter a vazão média da rede. [Yowa University 2011]

### 2.5.2.1 Desempenho de processamento sustentado

Para a obtenção do desempenho de processamento sustentado de ambos os *clusters* foi utilizado o pacote HPL, que é a ferramenta padrão para a realização de medições de desempenho de processamento de supercomputadores utilizada pelo projeto TOP 500 [TOP500 2011].

#### 2.5.2.1.1 Funcionamento, Configuração e Execução do HPL

O HPL [Silva et al. 2009] resolve um sistema de equações lineares do tipo  $A.x=b$ , onde  $A$  é a matriz dos coeficientes que é gerada estocasticamente pelo programa e possui tamanho  $N \times N$ .  $x$  e  $b$  possuem dimensão  $N$ . O primeiro passo para a resolução do sistema a ser aplicado pelo HPL é a fatoração da matriz  $A$  como sendo o produto  $A=L.U$ , onde  $L$  e  $U$  representam respectivamente as matrizes triangulares inferior e superior. A fatoração é realizada mediante pivotamento parcial de linha, por ser um método mais estável. Por fim, o algoritmo encontra a solução  $x$  através da aplicação sucessiva de passos de solução triangular,  $L.z=b$  e por fim  $U.x=z$ .

A matriz  $A$ , componente do sistema em questão, tem seus elementos distribuídos por uma grade bidimensional de processos  $P \times Q$  de maneira cíclica. Por sua vez, a matriz de coeficientes (dimensão  $N \times N+1$ ) é particionada em blocos de tamanho  $NB \times NB$ , também distribuídos ciclicamente na grade de processos supracitada. Na Figura 2 é possível ver essa distribuição para 6 processos, que foi a quantidade utilizada nos testes. Cada elemento  $A_{ij}$  corresponde a uma submatriz de dimensão  $NB \times NB$ . Esse procedimento é executado em todas as dimensões da matriz para assegurar principalmente a escalabilidade algorítmica e um bom balanceamento de carga.

	0			1			2		
0	$A_{11}$	$A_{14}$	$A_{17}$	$A_{12}$	$A_{15}$	$A_{18}$	$A_{13}$	$A_{16}$	
	$A_{31}$	$A_{34}$	$A_{37}$	$A_{32}$	$A_{35}$	$A_{38}$	$A_{33}$	$A_{36}$	
	$A_{51}$	$A_{54}$	$A_{57}$	$A_{52}$	$A_{55}$	$A_{58}$	$A_{53}$	$A_{56}$	
	$A_{71}$	$A_{74}$	$A_{77}$	$A_{72}$	$A_{75}$	$A_{78}$	$A_{73}$	$A_{76}$	
1	$A_{21}$	$A_{24}$	$A_{27}$	$A_{22}$	$A_{25}$	$A_{28}$	$A_{23}$	$A_{26}$	
	$A_{41}$	$A_{44}$	$A_{47}$	$A_{42}$	$A_{45}$	$A_{48}$	$A_{43}$	$A_{46}$	
	$A_{61}$	$A_{64}$	$A_{67}$	$A_{62}$	$A_{65}$	$A_{68}$	$A_{63}$	$A_{66}$	
	$A_{81}$	$A_{84}$	$A_{87}$	$A_{82}$	$A_{85}$	$A_{88}$	$A_{83}$	$A_{86}$	

Figura 2. Distribuição dos blocos lógicos da matriz na grade de processos

O HPL deve ser configurado em função da arquitetura dos processadores utilizados nos nodos do *cluster* (daí é extremamente importante todos os nós possuírem o mesmo tipo de processador). Essa configuração é feita em um arquivo a parte com o nome da arquitetura, seguindo padrões estabelecidos pelo HPL. Neste experimento, para ambos os *clusters*, o HPL foi compilado através do arquivo ia32.arch. Para a correta execução do HPL do ponto de vista de avaliação é necessário configurar um arquivo nomeado HPL.dat, que fica localizado no diretório onde foi compilado o HPL para o *cluster* em questão. Neste arquivo devem ser descritos o valor da ordem  $N$  da matriz, o valor de  $P$  e  $Q$ , que devem ser dois números cujo produto resulte na quantidade total de processadores, e o valor de  $NB$ , entre outros parâmetros. [Advanced Clustering 2011] oferece um mecanismo em uma pagina web que gera automaticamente um arquivo HPL.dat adequado para qualquer *cluster*. As informações necessárias, que devem ser passadas como parâmetro na página web, são a quantidade de nodos, o número de cores por nodo e a quantidade de memória principal por nodo. Na configuração do HPL para este experimento foram considerados apenas os nós escravos, ficando o *frontend* responsável unicamente pelo gerenciamento e monitoramento do *cluster*. Uma visualização do arquivo HPL.dat utilizado é fornecida na Figura 3.

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
8           device out (6=stdout,7=stderr,file)
1           # of problems sizes (N)
25344      Ns
1           # of NBs
128        NBs
0          PMAP process mapping (0=Row-,1=Column-major)
1          # of process grids (P x Q)
2          Ps
3          Qs
16.0       threshold
1          # of panel fact
2          PFACTs (0=left, 1=Crout, 2=Right)
1          # of recursive stopping criterium
4          NBMINs (>= 1)
1          # of panels in recursion
2          NDIVs
1          # of recursive panel fact.
1          RFACTs (0=left, 1=Crout, 2=Right)
1          # of broadcast
1          BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1          # of lookahead depth
1          DEPTHS (>=0)
2          SWAP (0=bin-exch,1=long,2=mix)
64         swapping threshold
0          L1 in (0=transposed,1=no-transposed) form
0          U in (0=transposed,1=no-transposed) form
1          Equilibration (0=no,1=yes)
8          memory alignment in double (> 0)
##### This line (no. 32) is ignored (it serves as a separator). #####
0          Number of additional problem sizes for PTRANS
1200 10000 30000      values of N
0          number of additional blocking sizes for PTRANS
40 9 8 13 13 20 16 32 64      values of NB
```

Figura 3. Arquivo HPL.dat utilizado

## 2.6. Testes e Medidas Realizados

Nessa análise foram realizadas 30 medições por teste. Sendo apresentados o cômputo da média e o intervalo de confiança para os testes de desempenho de processamento sustentado e de desempenho de transmissão de dados.

1. Teste 1 – Teve como objetivo obter a medida de desempenho sustentado de processamento do *cluster* implementado unicamente com máquinas físicas. Os resultados desse foram empregados como comparação aos resultados do teste 2.

2. Teste 2 – Esse teste teve como objetivo medir o desempenho de processamento do *cluster* de máquinas virtuais.

3. Teste 3 - Teve como objetivo medir a capacidade de transmissão de dados no cluster real. Os resultados deste teste serão comparados com os do teste 4.

4. Teste 4 – O objetivo é medir a capacidade de transmissão de dados no cluster virtual.

### 3. Analise dos Resultados

As medidas de desempenho médio de processamento sustentado realizadas com o HPL para os testes 1 e 2 são apresentadas na Figura 4, com um intervalo de confiança de 95%. Para o *cluster* de máquinas virtuais foi obtido um desempenho de processamento sustentado médio de 13,02 GFLOPS, ficando 1,53% superior ao *cluster* de máquinas reais de configuração análoga, que obteve um desempenho médio de 12,81 GFLOPS. Já as medidas relacionadas à capacidade de comunicação relativas aos testes 3 e 4 são apresentadas nas figuras 5,6 e 7, também com intervalo de confiança de 95%. Foram analisadas a vazão média (Figura 5) e a vazão média em função do tamanho do pacote de dados utilizado, sendo apresentados os resultados individuais para cada cluster na Figura 6 e um comparativo entre os desempenhos de ambas as estruturas na Figura 7. Na vazão média foi observada uma diferença de 21% em favor do cluster real. No quesito vazão média em função do tamanho do pacote de dados, o cluster real apresenta um crescimento mais elevado da capacidade média de vazão em relação ao seu correspondente virtual (Figura 7), tendo melhor desempenho para pequenas quantidades de dados. Todavia, à medida que a quantidade de dados dos pacotes aumenta, seus desempenhos vão se aproximando, de forma que, com uma grande quantidade de dados, se equiparam.

Quanto ao fato da diferença entre os dois *clusters* ser levemente a favor do *cluster* de máquinas virtuais, quando em vários outros experimentos, como nos realizados por [Zhao et al. 2009] e [Mello et al. 2010] serem bruscamente menores devido ao uso de Virtualização Total, pode ser atribuído ao fato de neste experimento o *cluster* virtual ser executado utilizando apenas 50% da capacidade do hospedeiro, sem necessitar competir por recursos com o mesmo e, ocasionalmente, como o hospedeiro não estava executando aplicativos extras, como antivírus, suítes de escritório e ferramentas de desenvolvimento, apenas seu próprio SO e a máquina virtual, é possível que o SO do hospedeiro tenha alocado mais recursos para o processamento das instruções da máquina virtual, ampliando levemente seu desempenho. As pequenas divergências verificadas no desempenho de rede para os pacotes pequenos são ocasionadas pelo compartilhamento de uma única interface de rede entre o hospedeiro e o sistema convidado, mesmo que o hospedeiro não esteja conectado a outra rede.

Porém, de uma maneira geral, como a diferença de desempenho não foi muito significativa, é conveniente afirmar que os dois *clusters* obtiveram desempenho real equivalente sobre as mesmas configurações, apresentando na prática o que era esperado em teoria.



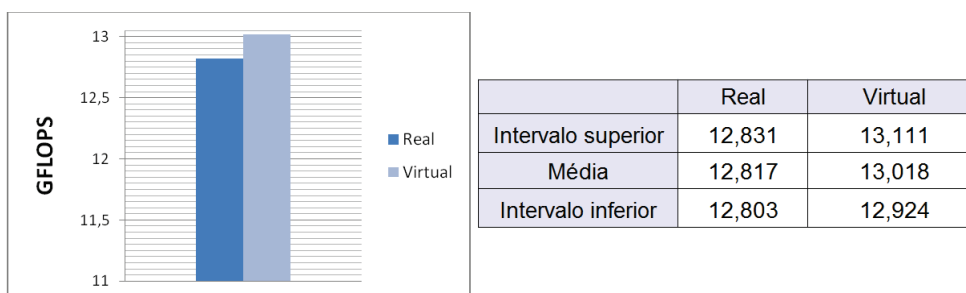


Figura 4. Desempenho de processamento sustentado

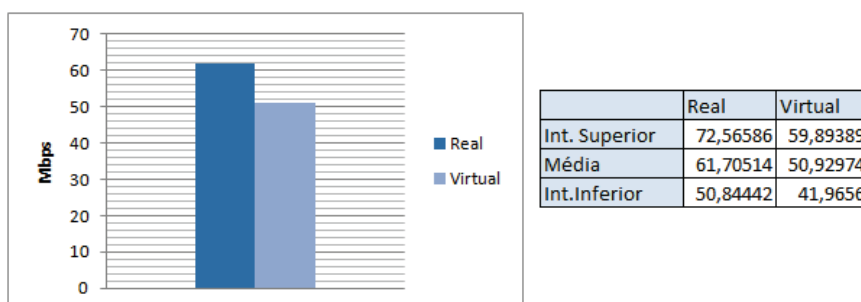


Figura 5. Vazão média da rede

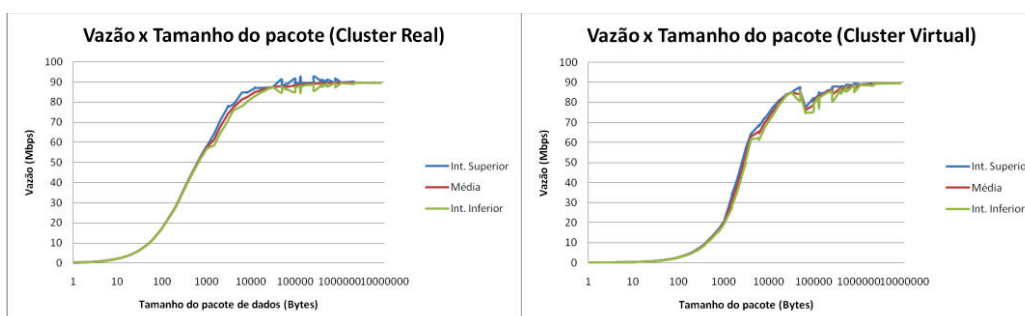


Figura 6. Vazão média x tamanho do pacote de dados

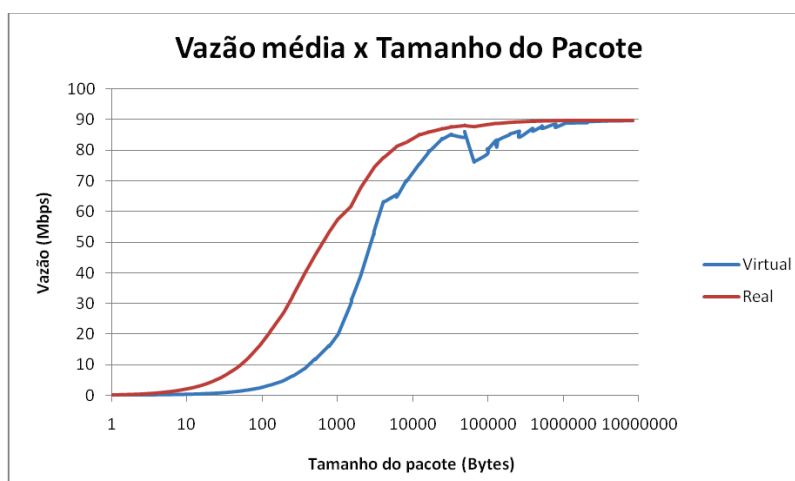
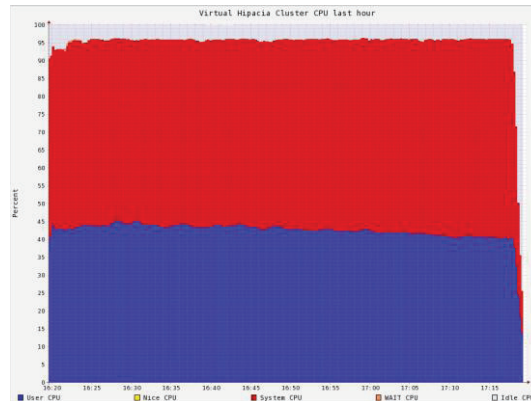


Figura 7. Comparativo da vazão média x tamanho do pacote de dados entre os clusters real e virtual

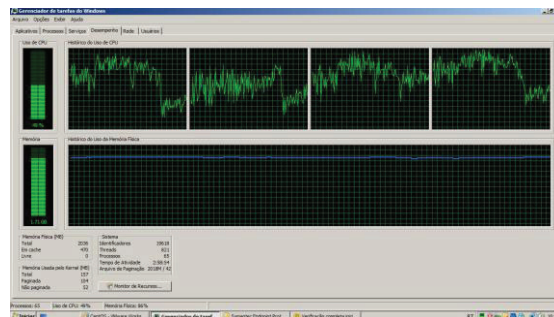
Durante a execução do HPL no *cluster* virtual foi observado através do *Ganglia* que todos os nodos faziam uso médio de 95% de suas CPUs, como indicado na Figura 8. Durante a execução do experimento foi visualizada, mediante o gerenciador de

tarefas do *Windows*, em todas máquinas hospedeiras dos nodos do *cluster* virtual a percentagem de uso de CPU, sendo constatado que de fato, o SO do hospedeiro sempre se encontrava fazendo uso de aproximadamente 50% dos recursos de CPU, oscilando um pouco entre 49-52%, tal como era esperado, já que praticamente não haviam softwares aplicativos extras em execução no hospedeiro, tendo apenas o nodo virtual em execução justamente para ser observado o seu impacto no uso de CPU.



**Figura 8. Uso de CPU pelo cluster virtual**

Foi observado também que a carga de processamento não se concentrou nos dois núcleos teoricamente reservados a VM, sendo distribuído entre todas as CPUs de maneira aproximadamente equivalente, com cada uma apresentando um uso próximo dos 25% (vide Figura 9). Esse particionamento também pode ter colaborado para o leve *superavit de performance* a favor do *cluster* de máquinas virtuais.



**Figura 9. Uso de CPU no host hospedeiro**

#### 4. Conclusões e Trabalhos Futuros

A pesquisa acima descrita levantou um conjunto de questionamentos relacionados ao uso de ambientes virtualizados para HPC e, de uma maneira mais específica, sobre os impactos da virtualização no desempenho quando ambas as estruturas possuem configurações similares.

Foi mostrado nesse estudo que, para as condições acima descritas é possível implementar um ambiente de *cluster* dentro de uma infraestrutura com propósito original diferenciado que não seja o uso para um ambiente HPC, *cluster* esse que mantém um desempenho similar ao de seu correspondente real. Isso faz com que possa ser aproveitada a capacidade ociosa das máquinas ou que ambas as estruturas possam coexistir em uso simultâneo, uma vez que, conforme observado, o *cluster* virtual só

utiliza os recursos que lhe foram alocados, podendo o restante dos recursos serem utilizados livremente.

As principais vantagens obtidas no emprego de múltiplas infraestruturas nos mesmos equipamentos físicos são a economia espacial obtida com a redução da quantidade de máquinas necessárias e a economia de recursos financeiros, uma vez que o custo proporcional por núcleo é reduzido em função do aumento da quantidade de núcleos por máquina.

Como trabalhos futuros a serem realizados serão realizados testes para verificar o impacto do virtualizador empregado no desempenho de processamento sustentado e no desempenho de rede de um *cluster* virtual, bem como em relação a *clusters* reais equivalentes. Outro estudo em pauta é o do impacto do uso de outros SOs nas máquinas hospedeiras, como o Linux, no desempenho de processamento do *cluster* virtual. Permutações dessas variáveis também serão realizadas com o propósito de se descobrir qual a melhor combinação possível de SO e virtualizador para a implementação desse modelo de *cluster*. Serão observados os efeitos do emprego do uso das instruções de Paravirtualização no desempenho e, em todos esses casos, pretende-se observar também o efeito do uso simultâneo das duas estruturas sobre o mesmo hardware para observar a ocorrência, ou não, de impactos no desempenho da execução das aplicações dos hospedeiros ou de impactos no desempenho dos nodos virtuais, para determinar mais precisamente o grau de viabilidade da adoção de *clusters* virtuais como forma de aproveitar o desempenho ocioso das grandes estruturas computacionais disponíveis nos ambientes corporativos e acadêmicos.

Os autores agradecem o apoio financeiro das agências de fomento: Fundação de Amparo a Ciência e Tecnologia de Pernambuco – FACEPE, ao CNPq e a UAG/UFRPE pelo uso do laboratório de informática.

## Referencias

- Advanced Clustering. (2011) “How Tune my HPL.dat file?”, <http://www.advancedclustering.com/faq/how-do-i-tune-my-hpldat-file.html>, March.
- Becker, D. J., Sterling, T., Savarese, D., Dorband, J. E., Ranawak, U. A., Packer, C. V. (1995) “Beowulf: A parallel Workstation for Scientific Computation”, In: *Proceedings of the 1995 International Conference on Parallel Processing*, August 14-18, 1995, Urbana-Champaign, Illinois, USA. Volume I: Architecture. pp. 11-14 CRC Press, ISBN 0-8493-2615-X
- Creel, M. (2011) “PelicanHPC tutorial”, <http://pareto.uab.es/mcreel/PelicanHPC/Tutorial/PelicanTutorial.html>, January.
- Huang, W., Liu, J., Abali, B., Panda, D.K. (2006) “A case for high performance computing with virtual machines”, In: *Proceedings of the International Conference on Supercomputing*, pp. 125-134
- Iowa State University. (2011) “NetPIPE README” <http://www.scl.ameslab.gov> February.

- Ivica, C., Riley, J.T., Shubert, C. (2009) “StarHPC - Teaching parallel programming within elastic compute cloud”, In: *Proceedings of the International Conference on Information Technology Interfaces, ITI*, pp. 353-356
- Mello, T. C., Schulze, B., Pinto, R. C. G., Mury, A. R. (2010) “Uma análise de recursos virtualizados em ambiente de HPC”, In: *Anais do VIII Workshop em Clouds, Grids e Aplicações, em conjunto com o XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 17-30
- Napper, J., Bientinesiy, P. (2009) “Can cloud computing reach the TOP500?”, In: *Proc. Combined Workshops on UnConventional High Performance Computing Workshop Plus Memory Access Workshop, UCHPC-MAW '09, Co-located with the 2009 ACM Int. Conf. on Computing Frontiers, CF'09*, pp. 17-20
- National Instruments. (2011) “Introdução à Virtualização”, [ftp://ftp.ni.com/pub/devzone/pdf/tut\\_9897.pdf](ftp://ftp.ni.com/pub/devzone/pdf/tut_9897.pdf), February.
- NSF. (2011) “Base Roll: Users Guide”, <http://www.rocksclusters.org/roll-documentation/base/5.4/>, January.
- Alves, M., J., P. (2008) “Construindo Supercomputadores com Linux”, 3º Edição, Brasport, pp. 245-248
- Ranadive, A., Kesavan, M., Gavrilovska, A., Schwan, K. (2008) “Performance implications of virtualizing multicore cluster machines”, In: *2nd Workshop on System-level Virtualization for High Performance Computing, HPCVirt 2008, held in conjunction with EuroSys 2008*, pp. 1-8
- Silva, V., Bentes, C., Guedes, S., Silva, G. P. (2009) “Arquitetura e Avaliação do Cluster de Alto Desempenho Netuno”, In: *Anais do WSCAD-SSC 2009 – X Simpósio em Sistemas Computacionais, em conjunto com o Simpósio Brasileiro de Arquiteturas Computacionais e Processamento de Alto Desempenho 2009*, pp. 52-59
- TOP500. (2011) “The Linpack Benchmark”, <http://www.top500.org/project/linpack>, March.
- VMware. (2011). “Guest Operating System Installation Guide”, [http://www.vmware.com/pdf/GuestOS\\_guide.pdf](http://www.vmware.com/pdf/GuestOS_guide.pdf), February.
- Zhao, T., Ding, Y., March, V., Dong, S., See, S. (2009) “Research on the performance of xVM virtual machine based on HPCC”, In: *4th ChinaGrid Annual Conference, ChinaGrid 2009*, pp. 216-221