

Carlos Alexandre Silva de Melo

# Tecnologias de Virtualização para HPC

– Monografia –

10 de setembro de 2014

Springer

**Resumo** A Computação em Nuvem tem encontrado uma série de aplicações para a Computação de Alto Desempenho (HPC). Sendo a virtualização uma tecnologia básica e um dos principais pilares para a computação em nuvem, torna-se necessária a realização de análises e a avaliação de seu desempenho, bem como de seu comportamento diante de aplicações que utilizam de maneira intensiva os recursos computacionais disponíveis. Este artigo apresenta e discute o estado da arte das tecnologias de virtualização aplicadas a HPC.

---

## Sumário

<b>1</b>	<b>Introdução.....</b>	<b>1</b>
<b>2</b>	<b>A Computação em Nuvem, A Virtualização e os Virtualizadores .....</b>	<b>3</b>
2.1	Computação em Nuvem .....	3
2.2	Virtualização .....	4
2.2.1	Virtualização em Nível de Sistema Operacional .....	4
2.2.2	Virtualização Total .....	4
2.2.3	Paravirtualização.....	7
2.2.4	Virtualização Assistida por Hardware .....	8
2.2.5	Envelhecimento de Software .....	9
<b>3</b>	<b>Os Virtualizadores e a Computação de Alto Desempenho ..</b>	<b>11</b>
<b>4</b>	<b>Considerações Finais e Trabalhos Futuros .....</b>	<b>15</b>
	<b>Referências .....</b>	<b>17</b>



## Introdução

Computação de Alto Desempenho (HPC) é um termo genérico utilizado para se referir a uma gama de aplicações que fazem uso intensivo de um ou mais tipos de recursos computacionais e são de natureza técnico-científica [Simons e Buell 2010]. Exemplos de aplicações de HPC são a resolução de sistemas lineares de grandes dimensões [Costa, Matsunaga e Brancher 2013] e a simulação de estruturas físicas de larga escala, como o Universo [Vogelsberger et al 2014]. Atualmente, muitas aplicações de HPC são hospedadas em nuvens computacionais, existindo soluções de computação em nuvem orientadas a aplicações científicas de alto desempenho, tais como Nimbus, Neblina e N-Clusters ([Keahey 2007], [Fernandes et al 2012], [Barbosa et al 2014]).

É consenso na academia que a virtualização é a tecnologia-chave para a computação em nuvem ([Vaquero et al 2008], [Armburst et al 2009], [Buyya et al 2009], [Endo et al 2010], [Karman et al 2014]). A virtualização é uma abstração que permite que uma ou mais instâncias de Máquinas Virtuais (VM) sejam hospedadas em um ou mais hospedeiros físicos ([Simons e Buell 2010], [Beserra et al 2014]).

Alguns requisitos fundamentais devem ser atendidos quando se emprega virtualização para HPC. O primeiro requisito é que a sobrecarga da virtualização não deve provocar impactos significativos no desempenho do sistema, sendo este requisito o “ponto crítico” para HPC [Ye et al 2010]. A virtualização deve maximizar o uso dos recursos físicos disponíveis [Nusbaum et al 2009], impedindo que recursos sejam subutilizados ([Jhonson et al 2011], [Beserra et al 2012]). O terceiro requisito é que a virtualização deve melhorar a administração do ambiente, permitindo a criação e destruição rápida de VMs e distribuição flexível de recursos de hardware. O quarto requisito é que a virtualização deve aumentar a confiabilidade do ambiente, provendo mecanismos de redundância e recuperação de falhas ([Ye et al. 2010], [Simons e Buell 2010]). Outro requisito é que a virtualização deve aumentar, ou ao menos não diminuir, a segurança do ambiente mediante isolamento de aplicações em VMs [Ye et al 2010]. Essa lista de requisitos não é exaustiva, mas pode ser

considerada como um bom ponto de partida para a análise de ferramentas de virtualização em contextos de HPC.

Apesar de promissor, o uso de virtualização em HPC possui limitações, pois nenhuma técnica de virtualização conhecida atende a todos os requisitos elencados no parágrafo anterior. Algumas abordagens e ferramentas de virtualização, para atender melhor um dos requisitos, acabam reduzindo o suporte a algum outro requisito. Além disso, despontaram outros desafios a serem vencidos para seu uso adequado em HPC, como a minimização dos efeitos da concorrência por recursos de um mesmo servidor físico por diversas máquinas virtuais [Beserra et al 2014], uma melhor adaptação a arquiteturas computacionais multicore e many-core ([Cheng e Chen 2013], [Walters et al 2014]), a necessidade de uma melhor gestão de recursos energéticos, focando na economia [Verma, Ahuja and Neogi, 2009] e o envelhecimento de software virtualizador [Thein and Park, 2009], este último já sendo verificada sua ocorrência em ambientes de cloud hospedadas sob nuvens Eucalyptus [Araujo, Matos and Maciel 2011] .

Este artigo apresenta e discute o estado da arte das tecnologias de virtualização aplicadas a HPC. O restante desse artigo é organizado como segue. A próxima seção apresenta as principais abordagens para provisionamento de virtualização. A seção 3 mostra como a virtualização tem sido utilizada na computação de alto desempenho. Encerrando o artigo, a Seção 4 apresenta as conclusões obtidas e as sugestões para trabalhos futuros.

## A Computação em Nuvem, A Virtualização e os Virtualizadores

Esta seção tem por objetivo o esclarecimento de grande parte dos questionamentos referentes às tecnologias que foram utilizadas para o desenvolvimento deste projeto. Focando-nos inicialmente apenas nos materiais teóricos destinados ao embasamento da pesquisa, trazendo consigo uma série de definições e suas principais analogias.

### 2.1 Computação em Nuvem

A Computação em Nuvem é um paradigma que consiste na alocação dinâmica de recursos virtualizados de hardware e software com o objetivo de fornecer os mais diversos serviços através da Internet (VAQUEIRO et al., 2009). Este modelo computacional vem sendo cada vez mais adotado por empresas ao redor do mundo que buscam a redução dos elevados gastos com a manutenibilidade da infraestrutura de um datacenter próprio, através da terceirização desse ambiente. Uma das principais características deste modelo está em prover ao usuário a sensação de possuir uma quantidade ilimitada de recursos computacionais, através da sua disponibilidade sob demanda e da sua elasticidade que, por sua vez, permitem que o serviço seja expandido praticamente em tempo real à medida que as requisições sejam realizadas (ARMBRUST et al., 2009). De maneira análoga FOSTER et al., (2008) definem Cloud Computing como um poderoso paradigma proveniente da computação distribuída que é caracterizado pela abstração de recursos de hardware e software, sendo então virtualizados, dinamicamente alocados, de grande capacidade de gerenciamento e onde o armazenamento, as plataformas e os serviços são oferecidos sob demanda aos clientes das provedoras através da Internet. Muitos dos serviços que fazem parte do dia-a-dia de centenas de milhões de pessoas funcionam ou estão hospedados em um ambiente de computação em nuvem. Desde a checagem de e-mails, ou a visualização de vídeos pelo maior site do gênero no mundo, o YouTube; Até a navegação entre páginas de amigos no Facebook, bem como o compartilhamento de fotos e vídeos por aplicativos como Instagram e Vine. A

Computação nas Nuvens não é um paradigma recente, pois se utiliza de tecnologias, abordagens, conceitos e práticas já há muito tempo definidos na área da computação (SUN MICROSYSTEMS, 2009). A primeira das referências a este termo é datada do ano de 1996, e teria surgido em um dos escritórios da HP por parte de um até então funcionário da empresa, George Favarolo, mas só depois de passada quase uma década é que este conceito tornou-se popular, graças, principalmente à participação de empresas como Google e Amazon que passaram a fazer uso do termo Cloud Computing, isto acarretou numa grande quantidade de informações referentes a este termo, que podemos encontrar em dezenas de milhões de páginas e referências a este paradigma na Internet (REGALADO, 2011). Para Hauck et al., (2010), o principal valor proposto pelo paradigma de Computação nas Nuvens é definido pela capacidade de prover aos clientes uma boa relação custo-benefício, de modo que o cliente consuma apenas os recursos de tecnologia da informação que sejam realmente necessários. Sendo assim, a provedora de serviços poderá fazer uso de uma arquitetura multi-tenant, que possibilitará a utilização de um mesmo recurso de hardware por inúmeros clientes, provendo mais economia. Um dos alicerces para a atual popularização da computação em nuvem é a virtualização, que proporcionou grandes mudanças no atual cenário de tecnologia da informação (MICROSOFT, 2012).

## 2.2 Virtualização

Existem várias abordagens diferentes para prover a virtualização de recursos, com implicações diretas positivas ou negativas na garantia de proporcionar os requisitos apresentados na seção anterior.

### 2.2.1 Virtualização em Nível de Sistema Operacional

Na virtualização em nível de sistema operacional, um único kernel hospedeiro é compartilhado com vários contêineres de máquinas virtualizadas. Tende a ser o tipo mais eficiente de virtualização, em termos de desempenho, porém é mais difícil de gerenciar e oferece pouco isolamento entre as VMs, consequentemente impactando na segurança dos ambientes.

### 2.2.2 Virtualização Total

A virtualização total permite a execução de sistemas operacionais convidados (SO) em um sistema operacional hospedeiro sem necessidade de modificar o kernel de nenhum dos sistemas. Entretanto, o provisionamento de uma interface completa para os sistemas convidados acarreta em uma grande sobrecarga no desempenho das VMs, indo em oposição ao requisito mais importante que a virtualização deve atender para ser empregada em HPC (não provocar impactos significativos no sistema). Embora não satisfaça o critério de desempenho



**Tabela 2.1.** Principais características dos virtualizadores de código aberto.

	<b>KVM</b>	<b>VirtualBox</b>	<b>Xen</b>
<b>Última versão</b>	Embutido no Kernel de Sistemas Operacionais baseados em Linux.	4.3.14	4.4.1
<b>Para-virtualização</b>	Não	Sim	Sim
<b>Virtualização Total</b>	Sim	Sim	Sim
<b>CPU Hospedeira</b>	x86, x86-64, IA-64, PPC	x86, x86-64	x86, x86-64, IA-64
<b>CPU Convidada</b>	x86, x86-64, IA-64, PPC	x86, x86-64	x86, x86-64, IA-64
<b>SO Hospedeiro</b>	Linux	Windows, Linux, Unix, Solaris, Mac OS.	Linux, Unix
<b>SO Convidado</b>	Windows, Linux, Unix	Windows, Linux, Unix, Solaris	Linux, Windows, NetBSD
<b>VT-x/AMD-v</b>	Requerido	Opcional	Opcional
<b>Núcleos Suportados</b>	64	128	128
<b>Memória Suportada</b>	4Tb	1Tb	5Tb
<b>Aceleração 3D</b>	VMGL	Open-GL, Direct3D	Xen-GL
<b>Live Migration</b>	Sim	Sim	Sim
<b>Licença</b>	GPL	GPL/Proprietária	GPL

a contento, essa abordagem de virtualização é bastante segura, devido ao completo isolamento que proporciona entre o ambiente hospedeiro e os ambientes virtuais [Nusbaum et al. 2009]. Os custos de desempenho provocados por essa abordagem de virtualização podem ser mitigados com o uso de virtualização assistida por hardware (Hardware-assisted virtualization) [Nussbaum et al 2009]. Existem várias soluções de virtualização disponíveis que utilizam virtualização total, como o VMware ESXi, o KVM, QEMU e o VirtualBox. A Figura 2.1 apresenta a hierarquia e uma representação gráfica da relação entre virtualização total e os recursos de hardware e software que são disponibilizados por um ambiente computacional que detenha ou ofereça suporte a tal tecnologia, vale salientar que em ambientes que utilizem de virtualizadores bare-metal não há necessidade de um sistema operacional na máquina hospedeira, implicando em uma leve mudança na representação apresentada.

### VMWare ESXi

O VMware vSphere ou “ESXi”, nome dado em virtude do tipo de arquitetura de hypervisor utilizada pela mesma, é a plataforma de virtualização da VMware. O vSphere é um hypervisor bare-metal, do tipo que interage diretamente com o hardware sem a necessidade de um sistema operacional para a máquina hospedeira, permite a criação e execução de diversas partições dispostas em várias máquinas virtuais que compartilham entre si os mes-



**Figura 2.1.** Virtualização Total

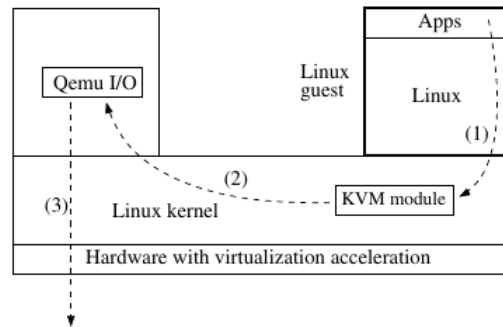
mos recursos físicos e o seu gerenciamento pode ser feito de maneira remota (VMWare, 2014).

### **KVM**

Kernel-based Virtual Machine é uma solução de virtualização completa para sistemas Linux em hardware x86 que contém extensões de virtualização (Intel-VT ou AMD-V). Graças ao KVM, é possível executar várias instâncias de máquinas virtuais. O KVM foi desenvolvido em máquinas AMD que utilizam a tecnologia AMD-V ou AMD Virtualization (KVM, 2013).

### **QEMU**

QEMU é o acrônimo para Quick Emulator ou Emulador Rápido. Trata-se de um emulador de máquinas virtuais e de um virtualizador opensource e que trabalha em conjunto com o Hypervisor para executar a sua função como emulador, os Hypervisors mais comuns que tendem a ser combinado a QEMU são o Xen e o já citado KVM (QEMU, 2013).



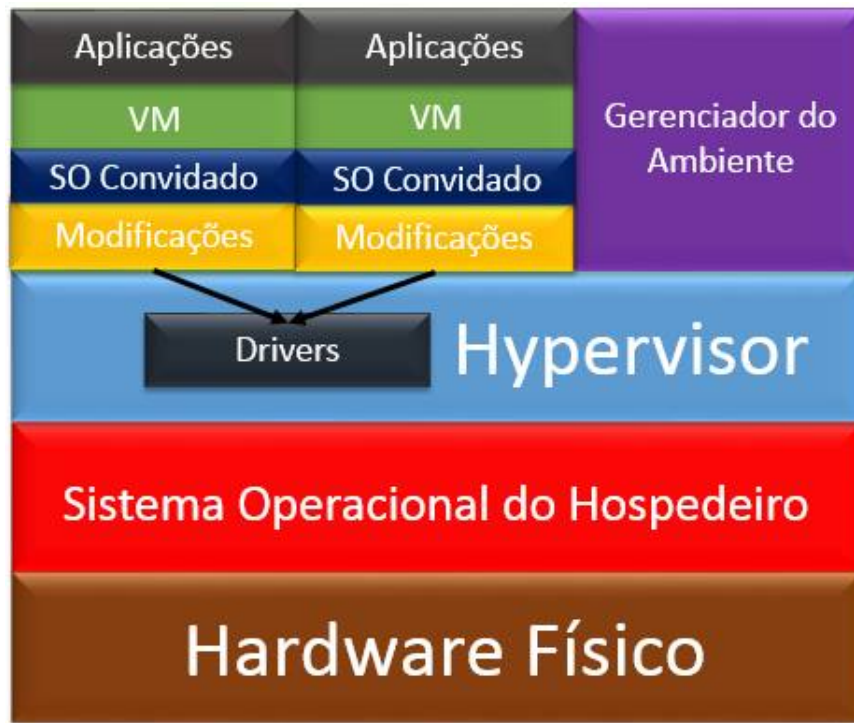
**Figura 2.2.** Mecanismo de I/O do KVM

### VirtualBox

VirtualBox é um virtualizador opensource destinado tanto a arquiteturas x86 quanto para AMD64 e Intel x64, é conhecido por ser a única solução profissional de virtualização que está disponível gratuitamente tanto para empresas quanto para usuários comuns (Oracle, 2014). Inicialmente, o suporte a virtualização assistida por hardware foi desenvolvido para os dispositivos de processamento e não para os dispositivos de entrada e saída (E/S), como interfaces de rede (NIC) e unidades de processamento gráfico (GPU) [Waters et al 2014]. A ausência de virtualização suportada diretamente pelo hardware nesses dispositivos leva a uma maior utilização da CPU hospedeira e tem impactos diretos no desempenho. Para mitigar isto, foram desenvolvidas abordagens híbridas onde drivers paravirtualizados para os dispositivos de E/S são empregados em conjunto com a virtualização assistida por hardware no processador e as soluções baseadas em virtualização total [Nusbaum et al 2009].

#### 2.2.3 Paravirtualização

Na paravirtualização os dispositivos de hardware não são emulados e sim acessados mediante o emprego de drivers paravirtualizados especiais, obtendo um melhor desempenho quando comparado com o obtido pela virtualização total, mesmo quando assistida por hardware [Beserra et al 2014]. A principal diferença em relação a virtualização total é que os kernels dos sistemas operacionais convidados devem ser modificados para providenciar novas chamadas de sistemas [Nusbaum et al. 2009]. Aumenta o desempenho, reduz a segurança, a confiabilidade e dificulta a administração do ambiente, por sempre ter que modificar os sistemas convidados. A Figura 2.3 apresenta a hierarquia e uma representação gráfica da relação entre paravirtualização e os recursos de hardware e software que são disponibilizados por um ambiente computacional que detenha ou ofereça suporte a tal tecnologia.



**Figura 2.3.** Paravirtualização

#### 2.2.4 Virtualização Assistida por Hardware

As técnicas virtualização totalmente baseadas em software, que capturam cada instrução proveniente dos sistemas operacionais convidados e as emulam no sistema operacional hospedeiro, são bastante custosas em termos de uso de CPU. Para mitigar esse problema foram desenvolvidos processadores com conjuntos de instruções específicas para virtualização. Em um processador sem instruções de virtualização, o kernel do SO hospedeiro e hypervisors são executados em um nível privilegiado dentro da hierarquia do processador, denominado Anel 0 (Ring 0). Entretanto, as máquinas virtuais, no melhor caso, são executadas no Anel 1 do processador, com menos acesso a instruções avançadas de processador [Nussbaum et al 2009]. Já em um processador com suporte a virtualização, um hypervisor passa a ser executado em um novo nível privilegiado na hierarquia do processador, denominado Anel -1 (Ring -1), com as máquinas virtuais sendo agora executadas no Anel 0, obtendo assim mais privilégios de acesso ao processador e melhorando o desempenho global. Conforme ilustrado na seção 2.1, os dispositivos de E/S tem tanta importância quanto o processador em aplicações de HPC e prover virtualização assistida por hardware apenas no processador causa impactos no desempe-

nho [Nusbaum et al 2009]. O auxílio de hardware para a virtualização de dispositivos de E/S surgiu com as Unidades de Administração de Memória de Entrada/Saída (IOMMUs) [Johnson et al 2011]. Uma IOMMU proporciona Acesso Direto à Memória (DMA) para dispositivos de E/S, mapeando endereços de dispositivos virtuais para endereços de dispositivos físicos [Walters et al 2014]. Com uso de IOUMMUs o desempenho de alguns dispositivos pode chegar a 98% do obtido por um sistema não virtualizado, conforme verificado para interfaces de rede [Johnson et al. 2011] e GPUs [Walters et al. 2014]. A virtualização assistida por hardware atende a muitos dos requisitos destacados na Seção 1. Ela não dificulta a administração do ambiente; proporciona segurança mediante o isolamento dos espaços virtuais quando em conjunto com um hypervisor de virtualização total; colabora a reduzir os gargalos de desempenho. O principal virtualizador que utiliza paravirtualização é o Xen [Ye et al 2010].

### 2.2.5 Envelhecimento de Software

A introdução ao termo Software Aging (SA) ou Envelhecimento de Software foi feita por (HUANG et al., 1995). Passados dezenove anos, o interesse de toda a comunidade, seja da indústria ou da academia apenas cresceu, (GROTTKE, MATIAS JR.; TRIVEDI, 2008). De acordo com Grottke, Matias JR. E Trivedi (2008), SA é um fenômeno que consiste principalmente na crescente degradação do estado interno de um software durante a sua vida operacional. Devido a sua propriedade acumulativa, o SA ocorre com mais intensidade em sistemas que estão sendo executados por um longo período de tempo. Tais sistemas possuem também uma maior probabilidade de entrarem em estado crítico, por terem sua integridade avariada pela vasta quantidade de falhas de software que ocorram durante a sua execução (MATIAS et al., 2010). Em contrapartida ao envelhecimento de software existem as estratégias que visam à redução dos impactos causados pela degradação dos ambientes e sistemas computacionais, que são intituladas de estratégias de Rejuvenescimento de Software. A mais comum delas foi à mencionada no estudo realizado por (HUANG et al., 1995) e aqui enfatizado. Nela uma rotina que realiza o reinício do sistema sempre que o mesmo alcança determinados níveis de degradação foi implementada, mas existem certos tipos de sistemas que são considerados críticos, e esta abordagem não seria a mais viável. Uma abordagem que pode ser utilizada neste caso é tratar os efeitos do envelhecimento de software através de um gerenciamento proativo, ou seja, não esperar que as mesmas ocorram e sim prevê-las com o máximo de antecedência possível (VAIDYANATHAN; TRIVEDI, 2005) e (IBM, 2012).



## Os Virtualizadores e a Computação de Alto Desempenho

A academia produziu uma série de trabalhos destinados principalmente a avaliação e a análise de desempenho dos virtualizadores de código aberto mais utilizados, como VirtualBox, XEN e KVM. Estes trabalhos em sua maioria observaram o impacto da virtualização no desempenho geral do sistema, no consumo de recursos, na degradação, justiça e na qualidade do serviço ofertado, e por fim na gerencia, usabilidade e flexibilidade desses virtualizadores. Em [Beserra et al. 2011] é observado que, quando em condições similares, o desempenho de processamento e a capacidade de comunicação de infraestruturas virtualizadas destinadas a atividades de HPC possui desempenho equipotente a uma infraestrutura implementada em sistemas de hardware nativos direcionados ao mesmo fim. Já [Beserra et al. 2012] avaliou o desempenho dos virtualizadores (VMWare Workstation, Virtual PC e VirtualBox) para implementação de clusters virtualizados que funcionavam em máquinas com o sistema operacional da Microsoft, o Windows, como hospedeiro. O VirtualBox foi o que apresentou um melhor desempenho global. Um aspecto importante que deve ser levado em conta na implementação de uma infraestrutura virtualizada destinada ao uso em HPC é o impacto do SO hospedeiro do desempenho do sistema [Beserra et al. 2014]. Em [Beserra et al. 2014] é realizado um estudo comparativo entre dois clusters virtualizados com o virtualizador VirtualBox. Os clusters tiveram as mesmas configurações de hardware e software, com o SO de ambos os clusters virtuais sendo o NACPI Rocks Clusters [Papadopoulos, Katz e Bruno 2001]. A única diferença entre ambos os clusters foi o SO hospedeiro, onde um dos clusters estava hospedado sob o SO Windows 7 e o outro cluster estava hospedado sobre o Kubuntu Linux, ambos sistemas que trabalham com palavras de 64 bits. O estudo conduzido por [Beserra et al 2014] indicou que o SO Windows não é adequado como hospedeiro de clusters virtuais para HPC em aplicações de dependam da rede de comunicação, por apresentar grande latência nas comunicações. Entretanto, para ambientes de ensino de programação paralela e tópicos de HPC, clusters virtuais criados com o virtualbox e hospedados em Linux são adequados [Beserra et al 2013]. Um outro trabalho que abordou o uso do VirtualBox para HPC é o de [Mello

et al. 2010], que avaliou o desempenho de distribuições Linux tanto de 32 quanto de 64 bits, utilizando do VirtualBox como virtualizador em máquinas hospedeiras, descobriu-se que as distribuições de 32 bits apresentaram um melhor desempenho. Outra descoberta realizada está diretamente relacionada a distribuição justa dos recursos de hardware, mais especificamente a dos recursos de rede, entre as máquinas virtuais, o VirtualBox não apresentou esta característica de qualidade e justiça de serviços, deste modo o mesmo serviço oferecido em duas máquinas virtuais distintas e que estivessem presentes no mesmo cluster apresentariam desempenhos também distintos para aqueles que estivessem a acessar este serviço, o que poderia gerar insatisfação de clientes, por exemplo. Já em [Younge et al. 2011] tem-se uma tabela comparativa entre os principais recursos de alguns dos virtualizadores de código aberto mais utilizados, XEN, KVM e VirtualBox foram avaliados. A tabela foi atualizada em [Beserra et al 2014] e a Tabela 1 deste apresenta uma versão atualizada desta tabela, em relação a apresentada em [Beserra et al 2014]. Em [Younge et al 2011] também foi analisada a viabilidade da virtualização para HPC, bem com o desempenhos dos virtualizadores de código aberto com o uso das ferramentas de testes High Performance Computing Benchmark (HPCC) e Standard Performance Evaluation Corporation (SPEC). Com os resultados obtidos foi possível observar uma ampla vantagem tanto por parte do KVM quanto do VirtualBox com relação ao XEN (embora o trabalho não mencione se os resultados obtidos com Xen são referentes ao seu modo de paravirtualização ou de virtualização total), apresentando um melhor desempenho e uma maior facilidade de gerenciamento. Pode-se acrescentar que o KVM apresentou melhor desempenho em aplicações para sistemas de memória compartilhada e o VirtualBox maior facilidade de gerenciamento. Outros trabalhos já abordaram o uso do KVM em CAD, como em [Karman et al 2014], que realiza um estudo comparativo de um cluster virtualizado em comparação com um cluster instalado em hardware nativo, sendo observadas perda de desempenho em relação ao ambiente original, principalmente nas tarefas que demandem por uso intensivo de recursos de rede. Foi analisado também em [Karman et al 2014] o efeito da concorrência na utilização dos recursos de hardware hospedeiros pelas VMs no desempenho dessas VMs. Foi observado que os recursos virtuais alocados primeiro mantêm um desempenho superior que os alocados depois, o que indica que o SO hospedeiro é incapaz de balancear os recursos entre múltiplos hospedeiros de uma maneira equipotente. Isso tem impactos em um contrato de serviços em uma cloud, onde usuários distintos que pagam por um determinado serviço devem ter um serviço de desempenho similar. Um trabalho similar é apresentado em [Johnson et al 2011], que implementa um cluster virtualizado para uso em aplicações de CAD e no ensino de programação paralela, entretanto não verifica os efeitos da concorrência no desempenho. Um ponto positivo em [Johnson et al 2011] é identificar que o principal gargalo na rede é a ausência de interfaces de rede IOMMU na maioria dos computadores. Em [Ye et al. 2010] é realizado um estudo comparativo entre clusters virtualizados com o Xen, com um dos clusters utilizando para-



virtualização e o outro utilizando virtualização total. Os resultados indicam que, no caso do Xen, a paravirtualização é a tecnologia mais adequada. Alguns trabalhos abordaram o uso dos virtualizadores em atividades de HPC, porém, focando no uso de Unidades de Processamento Gráfico (GPU), como [Walters et al 2014], que indica que todos os principais virtualizadores disponíveis atualmente são adequados para o uso em HPC com GPUs, desde que usem tecnologias que permitam o acesso direto ao hardware de GPU.



## Considerações Finais e Trabalhos Futuros

Neste trabalho foram apresentadas as principais técnicas de virtualização atualmente disponíveis, bem como suas vantagens e desvantagens individuais em contextos de HPC. De uma maneira geral o autor acredita que a virtualização com o uso de containeres Linux é a mais benéfica quando se busca extrair o desempenho máximo de um servidor hospedeiro e que a virtualização total é a mais indicada em casos onde a administração, confiabilidade do sistema e segurança da informação sejam mais necessárias.



---

## Referências

1. Costa, E. B. B., Matsunaga, F. T., & Brancher, J. D. (2013) Análise de escalabilidade e eficiência da fatoração LU usando CPU x GPU.
2. Beserra, D., Borba, A., Souto, S., Andrade, M. e Araújo A., (2012). "Desempenho de Ferramentas de Virtualização na Implementação de Clusters Beowulf Virtualizados em Hospedeiros Windows." In X Workshop em Clouds, Grids e Aplicações-SBRC 2012.
3. Grottke, M., Matias, R., Trivedi, K. S. (2008). The Fundamentals of Software Aging. 1<sup>o</sup> International Workshop on Software Aging and Rejuvenation/19<sup>o</sup> International Symposium on Software Reliability Engineering, Seattle.
4. Huang, Y., Kintala, C., Kolletis, N., Fulton, Dudley. Software Rejuvenation: Analysis, module and applications. 25th Symposium on Fault Tolerant Computing, Pasadena, 1995. 381-390.
5. IBM. IBM Smart Cloud. IBM, (2012). Disponível em: <http://www.ibm.com/cloud-computing/br/pt/>. Acesso em: 15 ago. 2014.
6. Johnson, E., Garrity, P., Yates, T., & Brown, R. A. (2011). Performance of a Virtual Cluster in a General-Purpose Teaching Laboratory. In Cluster Computing (CLUSTER), 2011 IEEE International Conference on (pp. 600-604). IEEE.
7. KVM, 2013. Disponível em: <http://www.linux-kvm.org/page/MainPage>. Acesso em: 15 ago. 2014.
8. Matias, R., Beicker, I., Leitao, B., e Maciel, P. (2012) Measuring Software Aging Effects Through OS Kernel Instrumentation. Software Aging and Rejuvenation (WoSAR), IEEE Second International Workshop, 2010.
9. Microsoft. Technet Microsoft. Technet. Disponível em: <http://technet.microsoft.com/en-us/magazine/hh855066.aspx>. Acesso em: 25 ago. 2014.
10. Oracle. Virtual Box. Wiki, (2014). Disponível em: <https://www.virtualbox.org/>. Acesso em: 17 ago. 2014.
11. QEMU. Quick Emulator. Wiki QEMU, 2013. Disponível em: <http://wiki.qemu.org/MainPage>. Acesso em: 17 ago. 2014.
12. Simons, J. E., & Buell, J. (2010). Virtualizing high performance computing. ACM SIGOPS Operating Systems Review, 44(4), 136-145.
13. Vaidyanathan, K.; Trivedi, K. S. (2005) A Comprehensive Model. Transactions on Dependable and Secure Computing, v. 2, n. 2, p. 124-137.

14. VMWare. VMWare vSphere. products, (2014). Disponível em: <http://www.vmware.com/br/products/vsphere/features/esxi-hypervisor/>. Acesso em: 17 ago. 2014.
15. Ye, Kejiang, et al. "Analyzing and modeling the performance in xen-based virtual cluster environment." High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on. IEEE, 2010.
16. Younge, Andrew J., et al. "Analysis of virtualization technologies for high performance computing environments." Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011.
17. Zhao, Tiezhu, et al. (2009). "Research on the performance of xvm virtual machine based on hpcc." ChinaGrid Annual Conference, 2009. ChinaGrid'09. Fourth. IEEE.
18. GROTTKE, M.; MATIAS JR., R.; TRIVEDI, K. S. The Fundamentals of Software Aging. 1<sup>o</sup> International Workshop on Software Aging and Rejuvenation/19<sup>o</sup> International Symposium on Software Reliability Engineering, Seattle, 2008.
19. Huang, Y., Kintala, C., Kolettis e N., Fulton, N. D. (1995) "Software Rejuvenation: Analysis, Module and Applications", In: Proceedings of the 25th Symposium on Fault Tolerant Computer Systems, p. 381-390, Pasadena, CA.
20. VAIDYANATHAN, K.; TRIVEDI, K. S. A Comprehensive Model.(2005). TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, v. 2, n. 2, p. 124-137.
21. Ian Foster, Yong Zhao, Ioan Raicu, Shylong Lu.(2008). Cloud Computing and Grid Computing 360-Degree Compared. Grid Computing Environments Workshop, Austin.
22. SUN MICROSYSTEMS.(2009) Introduction to Cloud Computing Architecture. Sun. [S.l.], p. 40.
23. REGALADO, A. MIT Technology Review. Technology Review, 2011. Disponível em: <http://www.technologyreview.com/news/425970/who-coined-cloud-computing/>. Acesso em: 30 dez. 2013.
24. HAUCK M., HUBER, M., KLEMS, M., KOUNEV, S. MULLER-QUADE, J., PRETSCHNER P., REUSSNER R., TAI, S. (2010) Challenges and Opportunities of. Karlsruhe Institute of Technology. Karlsruhe, p. 31.
25. Younge, A. J., Henschel, R., Brown, J. T., von Laszewski, G., Qiu, J., Fox, G. C., (2011) "Analysis of virtualization technologies for high performance computing environments." 2011 IEEE International Conference on Cloud Computing (CLOUD). IEEE.
26. Zhao, Tiezhu, "Research on the performance of xvm virtual machine based on hpcc." ChinaGrid Annual Conference, 2009. ChinaGrid'09. Fourth. IEEE, 2009.
27. Johnson, E., Garrity, P., Yates, T., & Brown, R. A. (2011, September). Performance of a Virtual Cluster in a General-Purpose Teaching Laboratory. In Cluster Computing (CLUSTER), 2011 IEEE International Conference on (pp. 600-604). IEEE.
28. Beserra, D. W. S. C., Souto, S. C. R. A., José, M., de Andrade, P., & de Araujo, A. E. P. (2011). Comparativo de desempenho de um cluster virtualizado em relação a um cluster convencional sob condições equipotentes. In IX Workshop em Clouds, Grids e Aplicacoes-SBRC 2011.
29. Beserra, D., Karman, R., Camboim, K., Araujo, J., Borba, A., & Araújo, A. (2014) Análise do Desempenho de Sistemas Operacionais Hospedeiros de Clus-

- ters Virtualizados com o VirtualBox. In XII Workshop em Clouds, Grids e Aplicacoes-SBRC.
30. VAQUEIRO, L. M. et al. A Break in the Clouds: Towards a Cloud Definition. *Computer Communication Review*, v. 39, n. 1, p. 50-55, Janeiro 2009.
  31. Nussbaum, L., Anhalt, F., Mornard, O., & Gelas, J. P. (2009). Linux-based virtualization for HPC clusters. In *Montreal Linux Symposium*.
  32. Cheng, Y., & Chen, W. (2013, October). Evaluation of Virtual Machine Performance on NUMA Multicore Systems. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on* (pp. 136-143). IEEE.
  33. Walters, J. P., Younge, A. J., Kang, D. I., Yao, K. T., Kang, M., Crago, S. P., & Fox, G. C. GPU Passthrough Performance: A Comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL Applications.
  34. Karman, R., Beserra, D., Endo, P., & Galdino, S. (2014). Análise de Desempenho do Virtualizador KVM com o HPCC em Aplicações de CAD. In *XII Workshop em Clouds, Grids e Aplicacoes-SBRC*.
  35. Papadopoulos, P. M., Katz, M. J., Bruno, G. (2013). NPACI Rocks: Tools and techniques for easily deploying manageable linux clusters. *Concurrency and Computation: Practice and Experience*, 15(7-8), 707-725.
  36. M. Vogelsberger, S. Genel, V. Springel, P. Torrey, D. Sijacki, D. Xu, G. Snyder, S. Bird, D. Nelson and L. Hernquist. Properties of galaxies reproduced by a hydrodynamic simulation. *Nature*, 509 (7499), pp. 177-182, 2014.
  37. Keahey, K., Freeman, T., Lauret, J., & Olson, D. (2007, July). Virtual workspaces for scientific applications. In *Journal of Physics: Conference Series* (Vol. 78, No. 1, p. 012038). IOP Publishing.
  38. Fernandes, Barbosa, J., Mury, A. R., & Schulze, B. (2012, October). Integration of cloud services in support to tests, simulations and knowledge dissemination with cluster environments. In *Computer Systems (WSCAD-SSC), 2012 13th Symposium on* (pp. 72-79). IEEE.
  39. Barbosa, J., Oliveira, V., Bandini, M., Schulze, B., & Mury, A. N-Clusters: Ferramenta para a gerência de ambientes de Computação Massivamente Paralela e Distribuída.
  40. ARMBRUST, M. et al. (2009) Above the Clouds: A Berkeley View of Cloud Computing. UC Berkeley Reliable Adaptive Distributed Systems Laboratory, Berkeley, 28.
  41. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6), 599-616.
  42. Endo, P. T., Gonçalves, G. E., Kelner, J., & Sadok, D. (2010, May). A survey on open-source cloud computing solutions. In *Brazilian Symposium on Computer Networks and Distributed Systems*.
  43. Verma, A., Ahuja, P., & Neogi, A. (2008, June). Power-aware dynamic placement of hpc applications. In *Proceedings of the 22nd annual international conference on Supercomputing* (pp. 175-184). ACM.
  44. Thein, T., & Park, J. S. (2009). Availability analysis of application servers using software rejuvenation and virtualization. *Journal of computer science and technology*, 24(2), 339-346.

45. Mello, T. C., Schulze, B., Pinto, R. C., & Mury, A. R. (2010). Uma análise de recursos virtualizados em ambiente de HPC. In IX Workshop em Clouds, Grids e Aplicações.
46. D. Beserra, M França, C. Melo, Y. Sousa, S. Souto, E. Sousa, M. Andrade and A. Araújo. (2013) “Ambiente Virtualizado para Ensino de Programação Paralela e Computação em Cluster. In: XXXIII Congresso da Sociedade Brasileira de Computação / XXI Workshop sobre Educação em Computação”, Maceió.