Análise do Comportamento de Aplicações Paralelas em Ambientes de Computação de Alto Desempenho Virtualizados

Adriano J. Ferruzzi[1] [2], **Christian E. Rothenberg**[2], **Leandro Martinez**[1]

[1]Instituto de Química
Centro de Computação em Engenharia e Ciências
Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brasil.

[2]Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brasil.

adrianof@unicamp.br, chesteve@gmail.com, leandro@iqm.unicamp.br

Abstract. *Cloud computing has become increasingly common in companies and universities. This happened because cloud computing is easier managed once your environment was completely configured. However, there is resistance in deploying high-performance cloud computing environments and it is associated with the fact that the cloud needs a hypervisor to administrate the virtual machines, resulting in a performance loss caused by the virtualization layer. On the other hand, virtualization technologies are constantly advancing and make the performance loss smaller over the years. In this article the performance of parallel applications and the infrastructure will be analyzed. In this regard, the paper seeks to evaluate the capabilities of the XenServer and KVM hypervisors in order to benchmark the performance of each environment and identify which virtualization platform is the best to create a virtual high performance computing environment.*

Resumo. *A computação em nuvem tem se tornado cada vez mais frequente em empresas e universidades, principalmente devido ao fato da nuvem computacional ser mais facilmente administrada uma vez que seu ambiente esteja devidamente configurado. Entretanto, há resistência na implantação de ambientes de computação de alto desempenho em nuvem e isso, provavelmente, está associado ao fato da nuvem precisar de um hypervisor para a administração das máquinas virtuais, resultando em uma perda de desempenho causado pela camada de virtualização. Por outro lado, as tecnologias de virtualização estão em constante avanço, tornando essa perda de desempenho menor com o passar dos anos. Sendo assim, o trabalho analisará o desempenho de aplicações paralelas e da infraestrutura utilizada e também irá avaliar as características dos hypervisors XenServer e KVM com o propósito de avaliar comparativamente o desempenho de cada ambiente e identificar qual a melhor hypervisor a ser utilizada na criação de um ambiente de computação de alto desempenho virtualizado.*

1. Introdução

Com a disponibilidade de alto processamento, memória e armazenamento, a virtualização e a computação em nuvem se tornaram cada vez mais acessíveis e utilizadas pelo mercado. Usando computação em nuvem as empresas podem deixar de investir em grandes servidores para investir em suas aplicações de negócio, possibilitando a economia de recursos e custos com infraestrutura.

Entretanto, esse cenário não ocorre quando o assunto é computação de alto desempenho. Uma das razões é o fato da nuvem computacional necessitar de hypervisors para fazer a virtualização das máquinas, gerando mais uma camada para processamento e, consequentemente, diminuindo a performance dos cálculos científicos.

Este projeto foca precisamente no comportamento de uso de cpu, memória e disco feito pelos hypervisors e pretende analisar seu comportamento se comparado a um ambiente nativo, ou seja, sem virtualização. Além de mapear o comportamento e o impacto da virtualização voltada a ambientes de computação de alto desempenho.

O estudo será desenvolvido com preceitos na literatura, pesquisa bibliográfica e estudo de caso com ambientes reais. Para atingir tal propósito o trabalho será dividido em sete partes, onde a primeira se trata deste preâmbulo que apontou a justificativa da temática, objetivos e metodologia. A seção 2 apresentará uma revisão bibliográfica que dissertará a respeito da computação de alto desempenho, virtualização de uma maneira breve. A Seção 3 mostrará uma revisão dos trabalhos relacionadas com o mesmo tema. Na Seção 4 será apresentada a metodologia com a proposta de desenvolvimento dos ambientes e seus softwares de análise, assim como suas métricas. Em seguida, na Seção 5, serão analisados os resultados obtidos e verificado qual hypervisor obteve melhor desempenho. Finalmente, na Seção 6 será feita a conclusão do trabalho que ressaltará os resultados alcançados e a Seção 7 discorrerá sobre possíveis trabalhos futuros.

2. Revisão Bibliográfica

2.1. Computação de Alto Desempenho

Computação de alto desempenho, ou ambiente de processamento de alto desempenho, se refere ao uso de um conjunto de computadores para resolução de grandes tarefas. Esse ambiente é caracterizado por realizar tarefas paralelas e complexas. Essas tarefas normalmente são simulações numéricas que necessitam de grande volume de recursos computacionais para sua resolução [Colvero 2005]. Além disso, existem outros tipos de clusters como o cluster de alta disponibilidade e o cluster de balanceamento de carga. Neste trabalho será utilizado apenas a arquitetura de cluster de alto desempenho.

Os clusters de computação de alto desempenho são voltados a simulações que necessitam de muito processamento, como sistemas de previsão meteorológica, simulações de dinâmica molecular, análise de genômica entre outros. Os nós do cluster de alto desempenho são interligados com utilização de redes, seja rede Ethernet ou Infiniband de alta performance e o seu principal objetivo é permitir a realização de cálculos numéricos da maneira mais rápida possível. Normalmente os clusters de computação de alto desempenho são configurados com sistema operacional Linux ou Unix e utilizam algumas bibliotecas para computação paralela como OpenMP e MPI, além de normalmente serem configurados de forma homogênea.

2.2. Virtualização

A ideia de virtualização surgiu na década de 1960 quando a IBM utilizava essa tecnologia em mainframes. Desde 2001 essa tecnologia está disponível para plataforma x86 [Caciato 2009]. A virtualização é uma tecnologia que possibilita a execução de vários sistemas operacionais e aplicativos simultaneamente no mesmo servidor. Na teoria, pode-se criar um ou vários servidores virtuais dentro de um único servidor físico. Algumas das soluções mais famosas na virtualização são a VMWare, Microsoft Hyper-V, XenServer, KVM e Virtualbox.

Cada máquina virtual oferece todos os recursos de computação e trabalha de forma transparente, dessa forma, é possível utilizar todos os recursos do sistema operacional instalado. Além disso, com a virtualização é possível executar dois ou mais sistemas distintos no mesmo servidor físico, aproveitando possíveis recursos ociosos e evitando a compra de novos equipamentos [Caciato 2009].

A respeito dos tipos de vitualização, as máquinas virtuais podem ser distinguidas de acordo com a localização da camada de virtualização. Se a camada de virtualização estiver diretamente sobre a camada de hardware ela é chamada de Tipo 1 ou Nativa. Caso a camada de virtualização estiver em cima de um sistema operacional, é chamada de Tipo 2 ou Modell Hosted Virtualization [Carissimi 2008] e [Santos 2009]. Além desses 2 tipos de virtualização ainda existem dois tipos básicos de virtualização: a virtualização completa e a paravirtualização. Trata-se de virtualização completa aquela onde toda a plataforma física é virtualizada, como: CPU, memória e disco. No caso da paravirtualização, as máquinas virtuais podem se comunicar diretamente com o hardware sem passar pelo hospedeiro, teoricamente ganhando um maior desempenho e maior performance em I/O do que as máquinas completamente virtualizadas por outro lado, isso requer sistemas operacionais hóspedes modificáveis, praticamente eliminando o sistema operacional MS Windows.

3. Trabalhos Relacionados

Em [Xavier 2013] o autor propõe a implementação de ambientes HPC com a utilização de containers como Linux-VServer, OpenVZ e Linux Containers (LXC). Em sua conclusão os containers apresentaram uma performance semelhante ao ambiente nativo, contudo o OpenVZ obteve a pior performance e o Xen obteve uma boa performance quanto ao uso intensivo de CPU. Nos testes com utilização de vários nós as diferenças se tornaram mais evidentes e o Xen ficou com a pior performance devido ao uso da rede. No artigo [Beserra 2016] o objetivo era descobrir qual hypervisor possui a melhor performance para fazer a virtualização com foco em computação de alto desempenho e em todos os casos o KVM se mostrou melhor para se aplicar ao uso de computação de alto desempenho.

No trabalho [Younge 2014], o autor se propõe a analisar pontos chave na virtualização para computação de alto desempenho, visando minimizar a sobrecarga causada pelo hypervisor. Os testes com MPI e FFT mostraram que o ambiente com KVM e Virtualbox ficam muito próximo do ambiente real enquanto que o Xen perde uma performance considerável. Neste outro artigo, [Younge 2011] faz uma análise entre diferentes tipos de hypervisors, com aplicabilidade em computação de alto desempenho, para verificar as vantagens e desvantagens de cada um. Como conclusão os autores dizem que é

possível criar um ambiente de computação de alto desempenho virtualizado, pois escolhendo o hypervisor certo a perda de performance seria muito baixa.

Em [Pickartz 2017], os autores apresentam um estudo sobre o estado da arte da virtualização aplicada à computação de alto desempenho, que não só considera os aspectos de workloads, mas também discute aspectos qualitativos como o desenvolvimento da virtualização nesse contexto. Em sua conclusão os autores reconhecem que a virtualização completa pode chegar a resultados próximos aos resultados de desempenho de programas executados em máquinas nativas, mas que o uso das máquinas virtuais depende das características de cada aplicação.

4. Metodologia

O objetivo dos benchmarks foi gerar uma medição de performance de CPU, memória RAM e leitura/escrita de disco. Além do estudo de caso com NAMD e Gamess. Ademais, os benchmarks foram feitos com cargas e granularidades diferentes de máquinas virtuais, pois segundo [Pickartz 2017], é possível que ocorra uma diferença de performance quando a granularidade da máquina virtual é diferente, mas que isso é gerenciado pelo isolamento da VM. Sendo assim, foram construídas três categorias de máquinas virtuais com as seguintes configurações:

Máquina Virtual	Threads	Memória	Disco
vm-1	40	50GB	100GB
vm-2	20	30GB	100GB
vm-3	10	15GB	100GB

Tabela 1. Configurações das Máquinas Virtuais.

O hardware utilizado para a montagem dos ambientes possui a seguinte configuração: 2 processadores Intel Xeon E5-2670 v2 de 2.50GHz de 40 threads, 64 GB de memória RAM DDR4 2133 MHz (8x8 GB), 1 disco SATA 1000 GB 7200 RPM e conexão de rede Gigabit Ethernet. Além de ser configurado com sistema operacional Suse SLES 12.1 em todas as máquinas.

As ferramentas utilizadas para a realização deste trabalho foram: O software `Sysbench` que foi utilizado em todos os ambientes para obter a uma estimativa de desempenho dos processadores. O software `dd`, utilizado em todos os ambientes para obter uma estimativa de desempenho de memória RAM e disco com variações de tamanho dos arquivos e ,finalmente, os softwares `NAMD` e `Gamess` para realização do estudo de caso. Todos os benchmarks foram executados 30 vezes e, em seguida, o software R foi utilizado para a análise dos resultados dos cálculos de desempenho, assim como, para a criação dos gráficos.

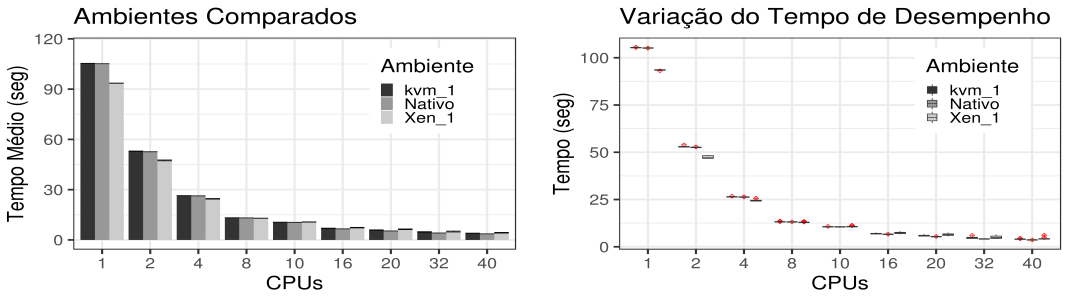
Para garantir a reprodutibilidade dos experimentos e dos dados obtidos, todos os dados e códigos usados para os testes iniciais estão disponíveis de forma pública no repositório: https://github.com/adrianoferruzzi/vmdata/ Os diretórios incluem os scripts utilizados, dados obtidos e os gráficos gerados.

5. Avaliações e Benchmarks

5.1. Avaliação de CPU

O software `Sysbench` foi utilizado em todos os ambientes para obter a estimativa de desempenho dos processadores. O parâmetro de comparação foi o tempo necessário, em segundos, para completar 100000 eventos, onde cada evento é o resultado do cálculo para descoberta de números primos entre 0 e 10000. Referente aos CPUs, o parâmetro `turbo boost` foi desativado para garantir a mesma frequência durante todos os experimentos. Além disso, o `cache` foi levado em consideração, mas os primeros experimentos tiveram um tempo similar a dos demais, ou seja, o `cache` não impactou no resultado obtido.

Ao analisar os resultados foi possível perceber que não houve uma diferença significativa entre os diferentes `flavors` de máquinas virtuais, portanto o ambiente Nativo foi comparado apenas com um `flavor` de cada máquina virtual. A Figura 1 apresenta a comparação entre o ambiente kvm-1, Xen-1 e nativo.



(a) Desempenho Comparado

(b) Dispersão no Boxplot

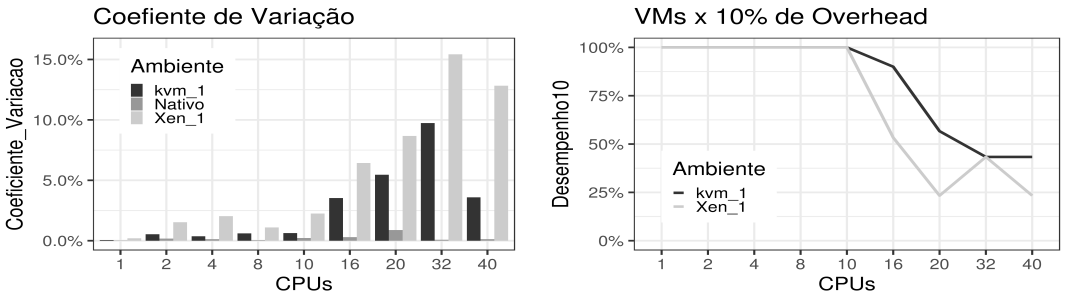
Figura 1. Desempenho e Dispersão dos Resultados

Ao analisar a Figura 1 foi possível perceber que ao utilizar poucas threads o Xen obteve melhor desempenho enquanto que o KVM executa o benchmarck sempre com uma performance próxima a do ambiente nativo e ao utilizar todas as threads disponíveis o ambiente nativo obteve melhor desempenho. Isso já era esperado, pois no ambiente nativo os processadores não precisam processar a camada de virtualização.

O próximo passo foi calcular a eficiência de processamento entre 1 thread e 40 threads. Assim, pode-se verificar que o Xen-1 obteve uma eficiência de 54.58%, enquanto que o KVM-1 conseguiu 65.29%, ficando mais próxima ao resultado do ambiente nativo que foi 72.42%. Por outro lado, levando em consideração a média de desempenho final e o intervalo de confiança é possível notar que ambos os hypervisors obtiveram um resultado estatístico semelhante.

Após o cálculo de eficiência de processamento o passo seguinte foi ver a dispersão dos dados obtidos que também está apresentado na Figura 1. Com a análise do `boxplot` foi possível identificar alguns outliers, ou anomalias. Essas anomalias são pontos fora da curva e representam os cálculos que não atingiram o desempenho esperado. Além disso, é possível ver o impacto do hyperthreading nos ambientes virtualizados, pois ao utilizar essa tecnologia foi observado que o coeficiente de variação sobe para mais de 15% no ambiente Xen-1 com 32 threads. Essa variação é causada pela distribuição entre núcleos físicos e virtuais.

Nesse sentido, a Figura 2 ilustra o coefiente de variação e ao analisar a variação de aproximadamente 15% uma nova dúvida surgiu: Quantos processos de fato atingiram um desempenho razoável? E após a criação de uma tabela com as comparações de tempos dos experimentos realizados foi possível verificar que o melhor comportamento, ao utilizar 40 threads, foi o KVM. Ele atingiu em 45% das vezes um desempenho comparável ao ambiente nativo com mais 10% de overhead, conforme apresentado na Figura 2



(a) Coeficiente de Variação

(b) Desempenho Atingido com 10% de Overhead no Ambiente Nativo

Figura 2. Coeficiente de Variação e Desempenho Atingido

O gráfico com o desempenho atingido mostra que ao utilizar o máximo do nó, 40 threads, o tempo de processamento terá mais que 10% de `overhead` em pelo menos 55% das vezes e isso pode acontecer mesmo com o melhor hypervisor utilizado no trabalho.

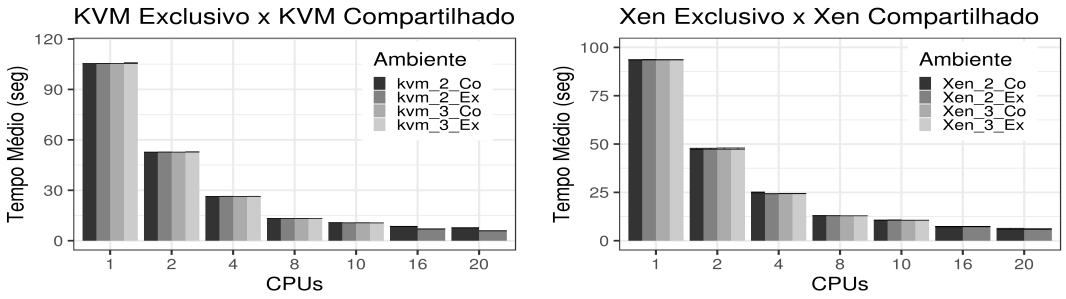
É importante salientar que foi utilizado a tecnologia de hyperthreading nos benchmarks com máquinas virtuais. Portanto, nota-se um pequeno ganho de desempenho ao comparar os testes com 20 threads e 40 threads e esse fenômeno também pode ser observado nos resultados com 16 threaeds e 32 threads. Isso acontece porque o hyperthreading cria núcleos virtuais permitindo dobrar a quantidade de threads e, obviamente, os núcleos virtuais não têm o mesmo desempenho dos núcleos físicos, por isso o ganho de desempenho é pequeno, mesmo quando é usado o dobro de threads.

Na segunda etapa os benchmarks foram feitos de forma simultânea, onde duas máquinas virtuais estavam hospedadas no mesmo hospedeiro e ambas executavam os testes ao mesmo tempo. O objetivo foi verificar se existe alguma mudança de comportamento quando comparadas com as máquinas virtuais que executaram de forma exclusiva no hospedeiro.

Com a análise da Figura 3 foi possível perceber que o desempenho se mantem semelhante ao desempenho das máquinas virtuais que utilizaram os hospedeiros de forma exclusiva.

Nesses benchmarks o Xen se mostrou levemente mais performático que o Kvm, entretanto, essa diferença não é tão significativa e, além disso, o Kvm é mais confiável em seus resultados, pois a variação dos resultados obtidos é bem menor. Isso torna o Kvm uma boa escolha caso seja utilizado em um ambiente onde os cálculos demandem majoritariamente processamento.

Finalmente, o resultado de desempenho compartilhado se mostrou dentro do esperado, pois o desempenho se manteve estável, isso acontece devido ao fato da virtualização



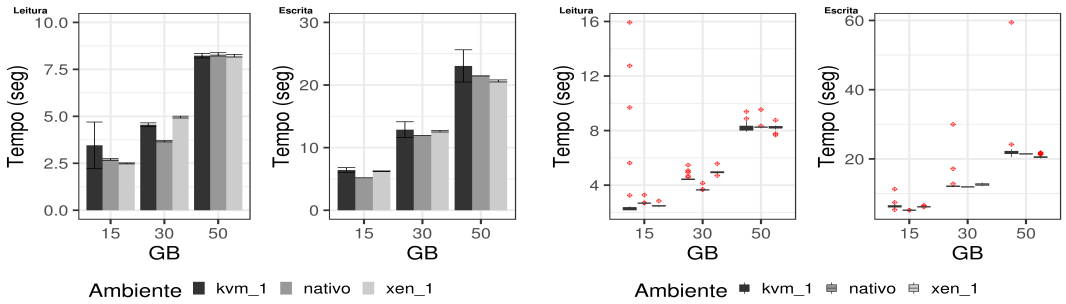
(a) KVM - Exclusivo x Compartilhado (b) Xen - Exclusivo x Compartilhado

Figura 3. Comparação dos Resultados Atingidos

garantir o isolamento dos processos e processadores virtualizados. Desta forma, uma máquina virtual não afeta o desempenho da outra, desde que estas não ultrapassem os limites do próprio hospedeiro.

5.2. Avaliação de Memória

Nessa etapa o software `dd` foi utilizado em todos os ambientes para obter uma estimativa de desempenho de memória no quesito de leitura e escrita sequencial com variações de tamanho dos arquivos. O parâmetro de comparação foi o tempo necessário, em segundos, para ler ou escrever determinado arquivo em memória.



(a) Leitura/Escrita (b) Boxplot

Figura 4. Leitura/Escrita e BoxPlot

A Figura 4 apresenta os gráficos com a comparação de velocidade de leitura e escrita, onde quanto menor o tempo mais eficiente é o sistema. Percebe-se que o Xen possui um melhor desempenho de leitura que os demais ambientes, principalmente nos testes de leitura de 15 GB, contudo quando o arquivo lido possui 30GB o ambiente nativo se torna melhor e com 50GB os ambiente se tornam estatisticamente semelhantes. Algo que chamou a atenção foi o tamanho da barra de intervalo de confiança do ambiente kvm durante a leitura de 15GB. A princípio esse caso foi interpretado como uma amostra com problemas, mas o mesmo fenômeno foi encontrado ao refazer as simulações. Isso quer dizer que a variação nos tempos de leitura do kvm são grandes e que normalmente ocorrem.

Além disso, ao identificar esses `outliers` no ambiente Kvm, foi percebido que eles representam os primeiros cálculos realizados e mostram como o hypervisor trabalha

com `cache` de memória, pois conforme os experimentos eram feitos os tempos diminuiam e esse comportamento é acentuado quando se observa o desempenho com 15GB.

De forma semelhante, é possível verificar que o desempenho de escrita de todos os ambientes são bem próximos e somente nos testes de 15G que o ambiente nativo conseguiu um desempenho estatisticamente melhor que os ambientes virtualizados. Nestes testes o Xen se mostrou um hypervisor com menor variação nos resultados, além de manter um bom índice de desempenho, principalmente na escrita.

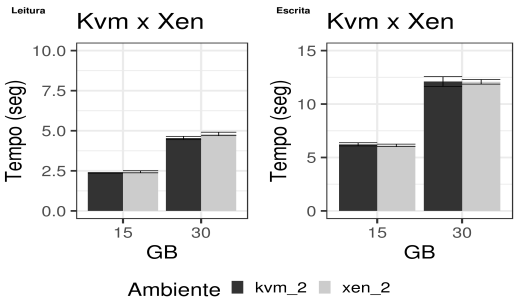


Figura 5. Desempenho de Leitura/Escrita de Memória em uso Compartilhado

Por sua vez, os hypervisors apresentaram um desempenho muito semelhante nos testes com uso de memória de forma compartilhada, mas dessa vez sem tantas anomalias nos resultados. A Figura 5 apresenta os gráficos que exemplificam o bom uso de memória de ambos os hypervisors, onde ambos trabalham de forma consistente e estatisticamente semelhantes.

Uma observação a ser considerada é que o hypervisor Xen não permitiu o provisionamento de máquinas virtuais com 60GB de memória, mesmo com o servidor hospedeiro possuindo 64GB. Um dos motivos é a necessidade de uma parte da memória do hospedeiro ser resevada para gerenciar seus processos e para executar o `Control Domain`. Devido a esse fator, os benchmarks foram feitos com no máximo 50GB. Também é importante mencionar que esse problema não ocorreu no caso do Kvm, pois sua arquitetura de virtualização não necessita de uma máquina virtual como `Control Domain`.

5.3. Avaliação de Disco

O software `dd` foi utilizado novamente em todos os ambientes para obter uma estimativa de desempenho de leitura e escrita no disco e também foram utilizadas variações de tamanho dos arquivos. O parâmetro de comparação foi o tempo necessário para ler ou escrever determinado arquivo. Antes da execução dos testes foi feita uma avaliação dos discos, cujos parâmetros de controle foram listados utilizando o `HDparm`. Dessa forma, foi possível garantir os mesmos valores em todos os ambientes, tornando os resultados mais confiáveis. Os parâmetros de controle avaliados e seus respectivos valores são: `CNominal Media Rotation Rate`: 10000; `R/W multiple sector transfer`: Max = 16 Current = 16; `Advanced power management level`: Max = 128 Current = 128; `DMA`: *udma6.

A primeira bateria de testes foi feita nos ambientes de forma exclusiva, ou seja, com apenas um processo gravando um arquivo. A Figura 6 apresenta os gráficos com a comparação do tempo necessário para leitura e escrita no disco.

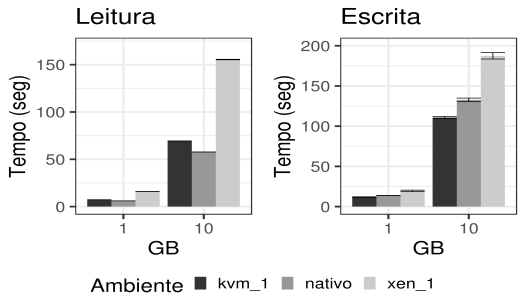


Figura 6. Desempenho de Leitura/Escrita de Disco

Nesse experimento, o Xen apresentou um desempenho pior que os outros ambientes, enquanto que o Kvm obteve um resultado mais próximo ao ambiente nativo, conseguindo até mesmo um desempenho melhor nos testes de escrita. Para explicar esse fenômeno que houve no ambiente com kvm foi necessário avaliar os parâmetros de cache dele, nesse caso, o parâmetro de cache constava `writeback`. Esse parâmetro habilita o uso de `cache` durante escrita em disco, ou seja, o convidado faz um `cache` no hospedeiro antes de gravar no disco. Além disso, o parâmetro `IO mode` também foi avaliado. Por padrão o `IO mode` usa modo `default (threads)`, cuja caracteristica é o uso de `I/O` assíncrono com opções de `I/O` direto.

A recomendação para ambiente com alta densidade de `I/O` em disco é o `cache=none` e o `IO mode=native`. O parâmetro `cache=none` permite, ao sistema convidado, escrever direto no disco sem a necessidade de fazer `cache` no sistema hospedeiro, dessa forma, o sistema hospedeiro reduz a cópia de dados. De fato, isso melhora o desempenho ao mesmo tempo que aumenta a chance de perda de dados, pois em caso de queda de energia a probabilidade de corrupção de arquivos é muito grande. [Jiri Herrmann 2019]

Por sua vez, o Xen apresentou um desempenho pior que os outros ambientes. Sua recomendação para obter um melhor desempenho é o uso de `storage` com RAID (Redundant Array of Independent Disks) e `LVM (Logical Volume Manager)` com vários discos. No entanto, o ambiente utilizado possuia apenas um disco local e não permitiu essa configuração. Em seu manual de ajuste de performance não havia opções de parâmetros de disco, mas havia recomendações para a alteração da memória alocada no `Control Domain`, pois a memória livre disponível é utilizada para `cache` de leitura. [Citrix 2019]

Na segunda etapa, os testes de leitura e escrita foram feitos com duas máquinas virtuais sendo executadas simultaneamente no sistema hospedeiro. Isso gera concorrência no uso de disco, pois o servidor hospedeiro possui apenas um disco físico. A Figura 7 apresenta os resultados obtidos.

Nestes gráficos da Figura 7, pode-se perceber que o uso compartilhado apresenta uma vantagem muito grande do Kvm sobre o Xen. Notadamente, o Kvm apresenta maturidade quanto ao uso das funcionalidades do `kernel` do Linux.

No final dos experimentos o resultado esperado era que o ambiente nativo fosse mais performático, pois o Kvm usa a técnica de virtualização total, onde todos os dispositivos são virtualizados e a máquina virtual é tratada como um processo regular do Linux

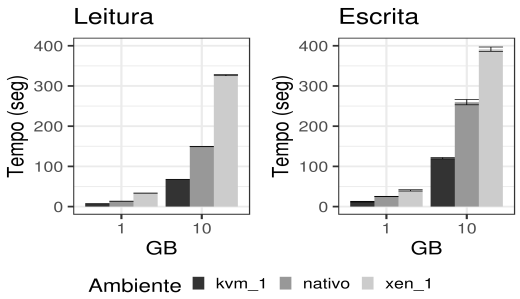


Figura 7. Desempenho de Leitura/Escrita de Disco Compartilhado

hospedeiro. Por isso o resultado foi bem surpreendente, pois demonstrou a maturidade do Kvm quanto ao uso de recursos e funcionalidades do `kernel` do Linux hospedeiro e, portanto, sendo uma melhor escolha para ambientes que demandam mais uso de discos em seus cálculos e processos.

Finalmente, é importante salientar que, neste trabalho, foram mantidos os parâmetros de forma padrão em ambos os hypervisors, pois a intenção é permitir que o ambiente atenda as difenrentes necessidades do ambiente de computação de alto desempenho. Além disso, a configuração de hardware utilizada no Xen foi a mesma do Kvm.

5.4. Estudo de Caso com NAMD - Nanoscale Molecular Dynamics

O primeiro estudo de caso foi feito utilizando o software `NAMD 2.13 - Nanoscale Molecular Dynamics` [James C. Phillips 2005]. Trata-se de um software que usa processamento paralelo projetado para simulação de alto desempenho de sistemas biomoleculares grandes. Neste trabalho, o benchmark com NAMD foi executado levando em consideração as melhores práticas descritas pelos criadores do software, portanto a métrica a ser utilizada foi a quantidade de nanossegundos de cálculo realizados por dia. [James C. Phillips 2005]

A proteína utilizada na realização desse benchmark foi a Apolipoproteína A-I. E ela foi escolhida por ser utilizada nos testes de benchmark realizados pelos desenvolvedores do software. [James C. Phillips 2005] O sistema simulado possui 92.224 átomos e o cálculo é feito até completar 500 ciclos de integração das equações de movimento.

O comportamento do software se caracteriza pelo alto consumo de processamento e baixo uso de memória e disco. Isso o torna um software com alto potencial de ser usado em ambientes virtualizados, pois sua perda de desempenho pode ser baixa tornando as vantagens do ambiente virtualizado maiores que as desvantagens. Após os testes foi observado que seu consumo de memória foi por volta de 350MB com 1 thread até 2933MB com 40 threads e o uso de disco foi de 186KB, comprovando o pouco uso de memória e disco.

Ao analisar os gráficos da Figura 8, percebe-se que a perda de desempenho é ínfima e que existe uma semelhança de desempenho dos ambientes, portanto, trata-se de um ótimo sinal para quem utiliza esse software em ambientes virtualizados. Além disso, é possível perceber a influência do hyperthreading nos resultados, pois pode-se observar um aumento de aproximadamente 10% no desempenho se comparado com os cálculos

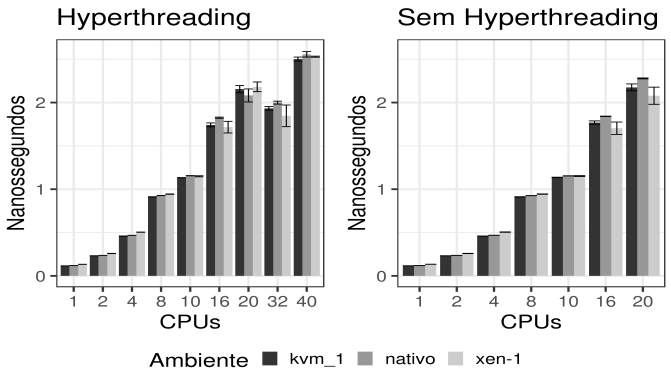


Figura 8. Desempenho do NAMD

realizados sem essa tecnologia.

Da mesma forma, é possível ver comportamento do hyperthreading ao observar os resultados de 20 threads e 32 threads, pois nesses resultados mencionados nota-se a queda de desempenho ao aumentar o número de threads. Esse comportamento é muito semelhante ao ocorrido no benchmark de CPUs e pode ser explicado pelo fato de que na verdade são 16 threads utilizando os núcleos físicos e 16 threads utilizando núcleos virtuais.

Obviamente essa distribuição fica a critério do sistema operacional e nem sempre os núcleos são distribuidos de forma exata, ou seja, 16 núcleos físicos e 16 núcleos virtuais, podendo ocorrer uma variabilidade nessa distribuição. Portanto, essa distribuição também explica a barra com o intervalo de confiança ser maior nos ambientes sem hyperthreading. Por esse motivo o correto é observar os resultados entre 16 e 32 threads, assim como, os resultados de 20 threads devem ser comparados aos de 40.

Ao mesmo tempo, o desempenho sem hyperthreading se mostrou menos variável no ambiente nativo e no ambiente Kvm, assim como o tempo de cálculo se manteve similar.

Em seguida, é possível perceber que a eficiência de escalabilidade do NAMD é muito boa, pois há um aumento de 21 vezes no desempenho ao usar 40 threads e aproximadamente 19 vezes sem hyperthreading. Finalmente, percebe-se que ao utilizar 40 threads todos os ambientes se tornam estatisticamente semelhantes e isso comprova a viabilidade do uso da virtualização para essa aplicação.

5.5. Estudo de Caso com Gamess - General Atomic and Molecular Electronic Structure System

No segundo estudo de caso foi utilizado o software `Gamess (General Atomic and Molecular Electronic Structure System)` [M.W.Schmidt 1993]. Trata-se de um software de processamento paralelo utilizado para pesquisas de dinâmica quântica. No trabalho, foram feitos cálculos teóricos utilizando resultados não publicados e a versão GAMESS-2014R1.

O comportamento do software se caracteriza pelo alto consumo de processamento e disco, tornando um software passível de ter seu desempenho bastante afetado em ambientes virtualizados. A medida de análise de desempenho usada foi o tempo, em se-

gundos, que o cálculo demorou para realizar 1000000 ciclos de integração das equações de movimento. Durante a execução do cálculo percebeu-se que foi utilizado 1.38GB de armazenamento em disco.

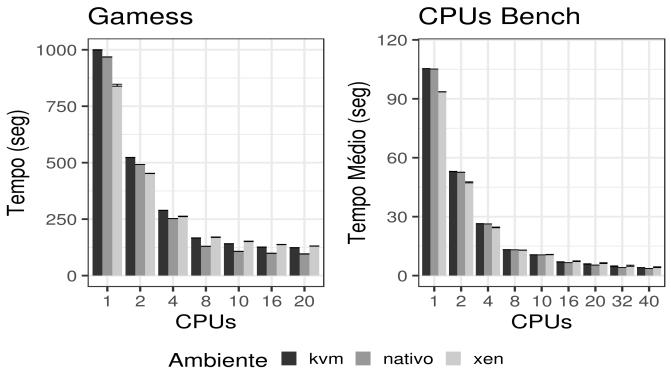


Figura 9. Desempenho do Gamess sem hyperthreading.

Ao analisar o gráfico de desempenho na Figura 9 foi possível verificar que o ambiente nativo conseguiu realizar os cálculos de forma mais performática ao utilizar 20 threads, seguido do Kvm e do Xen, respectivamente. Entretanto, fica nítido que o software não consegue um bom escalonamento, pois ao comparar o desempenho com 10 threads e 20 threads é possível notar que quase não há ganho mesmo sendo utilizado o dobro de recurso computacional.

Além disso, o Gamess também foi comparado ao comportamento do benchmarck de CPUs e apresenta uma certa diferença de comportamento, mas que pode ser explicado devido a necessidade de leitura e escrita em disco.

Neste caso, os ambiente virtuais conseguiram desempenhos próximos, embora com o Kvm sendo ligeiramente melhor. É importante notificar que o Gamess não funcionou de forma correta ao utilizar hyperthreading e ao utilizar somente os núcleos físicos o problema não ocorreu. Além disso, é importante salientar que o Gamess foi compilado da mesma forma e com os mesmos parâmetros em todos os ambientes.

Com a análise realizada ficou clara a possilidade da utilização do GAMESS em ambiente virtualizado, desde que o usuário esteja disposto a perder um pouco de desempenho e utilize um sistema de processamento paralelo fortemente acoplado.

6. Considerações Finais

Neste trabalho foi apresentado uma comparação de desempenho entre dois hypervisors gratuitos, onde o objetivo foi determinar a possibilidade de virtualização de um ambiente computacional para alto desempenho e qual o melhor hypervisor a ser utilizado de acordo com a aplicação apresentada. Assim, através dos resultados obtido foi possível perceber o impacto da virtualização nas aplicações utilizadas.

Nesse sentido, foi analisado o desempenho no uso de CPU, memória e disco de forma exclusiva e compartilhada, em ambientes com o hypervisor Kvm e Xen. Além de ter sido testadas diferentes aplicações em diferentes máquinas virtuais. No entanto, é importante salientar que os benchmarks foram feitos em um ambiente de processamento

paralelo fortemente acoplado, ou seja, um único sistema com grande capacidade computacional.

Por sua vez, os resultados mostraram que ao utilizar o número total de CPUs os dois hypervisors performam de forma semelhante, tanto no uso de forma exclusiva quanto no uso de forma compartilhada, além disso, ambos obtiveram um desempenho satisfatório se comparado ao ambiente sem virtualização, tornando ambos aptos para ambientes que demandam alto processamento de CPU. De forma semelhante, o uso de memória de ambos os ambientes foi satisfatório, embora a variação tenha sido alta o desempenho ficou com uma boa média, tanto no uso exclusivo quanto no uso compartilhado e nesse caso o Xen conseguiu uma pequena vantagem.

Nos testes de desempenho de leitura e escrita em disco, o Kvm obteve um desempenho consideravelmente maior que o Xen, tanto no uso exclusivo quanto compartilhado. Além disso, o Xen foi fortemente impactado nos testes de uso compartilhado e sua performance reduzida em torno de 50%. Sendo assim, previa-se que o Xen poderia ser utilizado para aplicações que demandassem alto uso de memória, enquanto que o Kvm seria uma melhor opção para aplicações que demandassem leitura e escrita em disco.

No entanto, na última etapa do trabalho o Kvm conseguiu um melhor desempenho nos benchmarks realizados com o NAMD e com o GAMESS, tornando o Kvm uma melhor escolha para ambientes de computação de alto desempenho que utilizem softwares de dinâmica molecular com comportamento semelhante ao NAMD e de softwares de dinâmica quântica com comportamento similar ao GAMESS.

Finalmente, o trabalho consequiu mapear os comportamentos distintos das aplicações e dos hypervisors permitindo a escolha dos ambientes virtuais de forma orientada a dados e resultados.

7. Trabalhos Futuros

Trabalhos futuros podem incluir benchmarks em ambientes de processamento paralelo fracamente acoplados, ou seja, um conjunto maior de memória e processamento distribuido em vários nós de computação. Além disso, pode-se criar benchmarks para testar a eficiência das GPUs que potencializam muito o desempenho de cálculos científicos. Finalmente, esses resultados podem ajudar a comunidade científica a planejar seu ambiente de computação de alto desempenho de forma consciente e orientada a dados.

8. Agradecimentos

Os autores gostariam de agradecer à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo 2013/08293-7 e ao CEPID-CCES pelos recursos computacionais e suporte a esta pesquisa.

Referências

Beserra, D.; Moreno, E. D. E. P. T. B. J. (2016). Performance Evaluation of Hypervisors for HPC Applications. *IEEE International Conference on Systems, Man, and Cybernetics*, (14).

Caciato, L. E. (2009). Virtualização e Consolidação dos Servidores do Datacenter. *Centro de Computação da Universidade Estadual de Campinas – São Paulo*, (23).

Carissimi, A. (2008). Virtualização: da teoria a soluções. *26º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.

Citrix (2019). Citrix XenServer 7.1 Administrator's Guide.

Colvero, T. A.; Dantas, M. e. C. D. P. (2005). Ambientes de Clusters e Grids Computacionais: Características, Facilidades e Desafios. *Congresso Sul Brasileiro de Computação*.

James C. Phillips, Rosemary Braun, W. W. J. G. E. T. E. V. C. C. R. D. S. L. K. e. K. S. (2005). Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*.

Jiri Herrmann, Yehuda Zimmerman, D. P. e. S. R. (2019). Virtualization Tuning and Optimization Guide.

M.W.Schmidt, K.K.Baldridge, J. S. M. J. S. N. K. S. T. M. J. J. (1993). General Atomic and Molecular Electronic Structure System GAMESS . *Journal of Computational Chemistry*.

Pickartz, S. e. a. (2017). *Virtualization in HPC - An Enabler for Adaptive Co-Scheduling?*, volume 28nd of *Advances in Parallel Computing*. IOS Press, Lansdale, USA, 1 edition.

Santos, D. L. (2009). Migração de servidores físicos para virtuais – P2V usando ferramentas Open Source. *Curso de Especialização em Redes e Segurança de Sistemas*.

Xavier, M. G.; Neves, M. V. R. F. D. F. T. C. L. T. e. R. C. A. F. (2013). Performance Evaluation of Container Based Virtualization for HPC Computing Environments. *IEEE - 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, (27).

Younge, A. J. e. a. (2011). Analysis of Virtualization Technologies for High Performance Computing Environments. *IEEE - 4th International Conference on Cloud Computing*, (01).

Younge, A. J. ; Fox, G. C. (2014). Advanced Virtualization Techniques for High Performance Cloud Cyberinfrastructure. *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, (29).