



Avaliação de tecnologias de virtualização para implantação de nuvem privada

Valquíria Prestes Belusso
Universidade Estadual do Oeste do Paraná
Cascavel, Brasil
valquiria.belusso@unioeste.br

Guilherme Galante
Universidade Estadual do Oeste do Paraná
Cascavel, Brasil
guilherme.galante@unioeste.br

Abstract — A fundamental aspect for the construction of private clouds is the choice of the virtualization solution. The objective of our work is to compare the performance of three open source virtualization solutions, Xen, KVM and VirtualBox, in order to choose the most appropriate, in terms of performance, for the implementation of a private research cloud.

Resumo — Um aspecto fundamental para a construção de nuvens privadas é a escolha da solução de virtualização a ser utilizada. Este trabalho tem como objetivo comparar o desempenho de três soluções de virtualização de código aberto, Xen, KVM e VirtualBox, com o intuito de escolher a mais apropriada, em termos de desempenho, para a implantação de uma nuvem privada de pesquisa.

Palavras-chave— virtualização; cloud privada; análise de desempenho.

I. INTRODUÇÃO

Considerando a realidade das universidades, diversos projetos de pesquisa baseiam-se na execução de experimentos e simulações computacionais para os mais diversos fins e áreas (engenharias, ciência da computação, biologia, farmácia, física). Embora as áreas de aplicação sejam diversas, há problemas de ordem técnica que são compartilhados por todas elas, dentre os quais destaca-se a demanda por recursos computacionais.

O que ocorre hoje é que os recursos estão dispersos nos diversos laboratórios das universidades e em grande parte do tempo estão subutilizados. Isso pode ser resolvido por meio da criação de laboratórios virtuais baseados em nuvens privadas. Uma nuvem privada pode ser implantada com os recursos da própria instituição e para o seu uso exclusivo.

Nesse caso, os computadores e servidores ociosos podem ser utilizados para fornecer o ambiente de execução necessário para os experimentos científicos. Tais recursos são disponibilizados para cada experimento na forma de máquinas virtuais.

Um aspecto fundamental para a construção dessas nuvens privadas é a escolha da solução de virtualização a ser utilizada. Neste contexto, este trabalho tem como objetivo comparar o desempenho de três soluções de virtualização de

código aberto, Xen, KVM e VirtualBox, com o intuito de escolher a mais apropriada para a implantação de uma nuvem privada de pesquisa. São utilizados dois *benchmarks*, o Sysbench e o Nbench, para análise do desempenho de CPU e memória.

II. TECNOLOGIAS DE VIRTUALIZAÇÃO

A virtualização é uma tecnologia que oferece uma camada de abstração dos verdadeiros recursos de uma máquina, provendo um *hardware* virtual para cada sistema, com o objetivo de abstrair as características físicas e à forma como os sistemas operacionais e aplicações interagem com os recursos computacionais [1].

Isso permite que as organizações particionem um único computador físico em diversas máquinas virtuais (VMs). Cada máquina virtual pode interagir de forma independente e executar diferentes sistemas operacionais ou aplicativos ao mesmo tempo em que todas compartilham os recursos de um único computador físico (*host*, ou *hospedeiro*).

A camada de virtualização propriamente dita é o hipervisor ou monitor de máquina virtual. Os hipervisores, são um intermediador entre o *hardware* e as máquinas virtuais, e controlam o acesso ao *hardware* físico para cada sistema operacional convidado (para cada VM) [2].

Neste trabalho, comparamos três hipervisores de código aberto: Oracle VirtualBox, KVM e Xen.

Oracle VirtualBox é um sistema de virtualização multiplataforma para *hardwares* Intel ou AMD compatíveis. Permite executar um sistema operacional em uma máquina virtual hospedada em sistemas Windows, Mac OS X, Linux ou Oracle Solaris. O VirtualBox é um hipervisor de tipo 2. Enquanto um hipervisor *bare-metal* ou tipo 1 seria executado diretamente no *hardware*, o VirtualBox requer a instalação sobre um sistema operacional existente. Ele pode, portanto, ser executado junto com os aplicativos existentes naquele *host* [3].

O KVM (máquina virtual baseada em *kernel*) é uma tecnologia de virtualização baseada no Linux. Seu funcionamento consiste em converter o Linux em um hipervi-



sor tipo 1, interagindo diretamente com o *hardware* da máquina física. Para executar VMs, todos os hipervisores precisam de alguns componentes em nível de sistema operacional, como gerenciador de memória, escalonador de processos, suporte para entrada/saída (E/S), *drivers* de dispositivo, gerenciador de segurança, suporte a rede e entre outros. O KVM tem todos esses componentes por fazer parte do *kernel* do Linux. Toda máquina virtual é implementada como um processo regular que é gerenciado pelo escalonador padrão do Linux. Também conta com *hardware* virtual dedicado, como placa de rede, adaptador de placa gráfica, CPU(s), memória e discos [4].

O Xen é um hipervisor tipo 1 que é executado diretamente no *hardware host*. O Xen é bem conhecido pelo uso de paravirtualização e desempenho quase nativo [5]. O hipervisor Xen é gerenciado por um convidado privilegiado específico em execução no hipervisor conhecido como Domínio-0 ou Dom0. Dom0 é um *kernel* Linux especialmente modificado que é iniciado pelo hipervisor Xen durante a inicialização do sistema. É responsável por gerenciar todos os aspectos da outra máquina virtual sem privilégios ou Domain-Us (DomU) que também estão em execução no hipervisor.

III. BENCHMARKS

Benchmarks consistem em testes padronizados que são utilizados para medir o desempenho de diferentes sistemas em tipos específicos de aplicações [6].

Para testar as tecnologias de virtualização em termos de desempenho de CPU e memória foram utilizados 2 *benchmarks* de código aberto e disponíveis para sistemas GNU/Linux, o Sysbench e o Nbench.

Sysbench [7] é uma ferramenta *multithread* com *scripts* baseada no LuaJIT, usado para banco de dados, mas também para criar cargas de trabalho arbitrariamente complexas para uso diverso. Tem como objetivo obter rapidamente informações sobre o desempenho do sistema sem a criação de pontos de medida complexos no banco de dados. A ferramenta oferece *benchmarks* para aferição de CPU, Memória e Entrada/Saída.

O *benchmark* de CPU do Sysbench é um dos mais simples, cada solicitação consiste no cálculo de números primos até um valor especificado por um parâmetro (`--cpu-max-prime`). Os cálculos são realizados usando números inteiros de 64 bits. Cada *thread* executa as solicitações simultaneamente até o número total de solicitações ou até o tempo total de execução exceder os limites especificados nas opções da linha de comando [8].

O teste de memória do Sysbench pode ser usado para comparar leituras ou gravações de memória sequencial. Nesse teste, o Sysbench aloca um *buffer* de memória e, em seguida, lê ou escreve nele, cada vez para o tamanho de um ponteiro (32 ou 64 bits) e até que o tamanho total do *buffer* tenha sido lido ou escrito. Esta atividade continuará até que o volume fornecido (`-memory-total-size`) seja alcançado. A carga pode ser aumentada ou reduzida utilizando várias *threads*, tamanho do *buffer* e tipo de solicitação [8].

Já o *benchmark* do Linux/Unix Nbench BYTEmark [9] inclui relatórios de CPU, cache e memória, desempenho de números inteiros e de ponto flutuante.

Os resultados são obtidos através de um conjunto de aplicações com características distintas.

Essas aplicações incluem: (1) *Numeric sort* - classificação de números inteiros; (2) *String sort* - classificação de *strings*, testa o desempenho da movimentação de memória; (3) *Bitfield* - manipulação de *bits*; (4) *Emulated floating-point* - emulação de ponto flutuante em *software*; (5) *Fourier coefficients* - análise numérica para o cálculo de séries aproximações de formas de onda; (6) *Assignment algorithm* - um algoritmo de alocação de tarefas; (7) *Huffman compression* - algoritmo de compressão de texto e gráficos; (8) *IDEA encryption* - algoritmo de criptografia de cifra de bloco; (9) *Neural net* - simulador de rede de propagação; e (10) *LU Decomposition* - algoritmo para resolução de equações lineares. Cada um dos testes visa coletar informações distintas sobre o desempenho de um *hardware* físico ou virtual.

Na próxima seção o resultado das comparações de desempenho para os hipervisores introduzidos na Seção II, empregando os *benchmarks* descritos nessa seção.

IV. RESULTADOS E DISCUSSÃO

Os testes de desempenho dos hipervisores foram realizados em uma única máquina física com um Intel Core i3-7100, 2 núcleos e com sistema operacional GNU/Linux Mint 18. O computador está equipado com 8 GB de RAM DDR4, bem como um disco rígido de 500 GB de 5.400 RPM. Para cada hipervisor, criou-se uma máquina virtual com 2 núcleos de processamento, 2 GB RAM e sistema operacional GNU/Linux Ubuntu 18.04 LTS.

Os *benchmarks* foram instalados na máquina física e também nas máquinas virtuais especificadas. Em seguida foram realizados vários testes para avaliação do desempenho de CPU e memória.

As Figuras 1 e 2 apresentam os resultados dos testes foram realizados com o Sysbench. A Figura 1 mostra o tempo total (em segundos) do teste com parâmetro `--cpu-max-prime` com valor 50.000.000. Os resultados com o parâmetro com o valor 50.000 é apresentado na Figura 2.

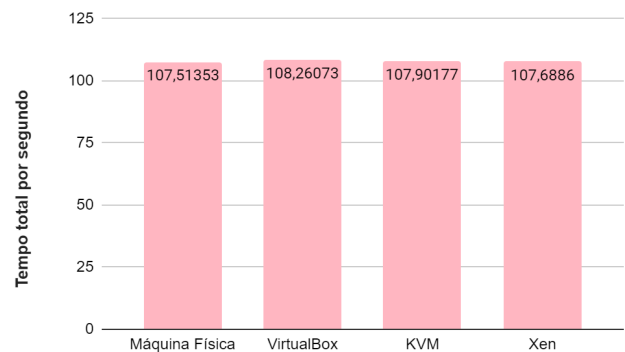


Fig 1. Teste 1 de desempenho de CPU Sysbench

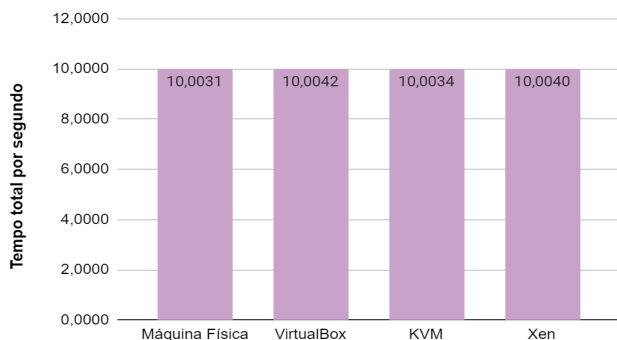


Fig 2. Teste 2 de desempenho de CPU Sysbench

Observando os resultados, pode-se notar que o desempenho dos 3 hipervisores são muito semelhantes, inclusive ao resultado da máquina física. Portanto, baseando-se nesses testes, pode-se afirmar que qualquer solução poderia ser escolhida para a implementação da nuvem privada.

O benchmark do Nbench BYTEmark produz dois índices gerais: índice inteiro e índice de ponto flutuante, para esses indicadores, quanto maior o resultado, melhor. O índice inteiro é a média geométrica dos testes que envolvem apenas processamento de inteiros – *numeric sort*, *string sort*, *bitfield*, *emulated floating-point*, *assignment*, *Huffman* e *IDEA* – enquanto que o índice de ponto flutuante é a média geométrica dos testes que requerem processamento de ponto flutuante – *Fourier*, *neural net* e *LU decomposition* [10]. A Figura 3 apresenta os resultados obtidos para estes índices.

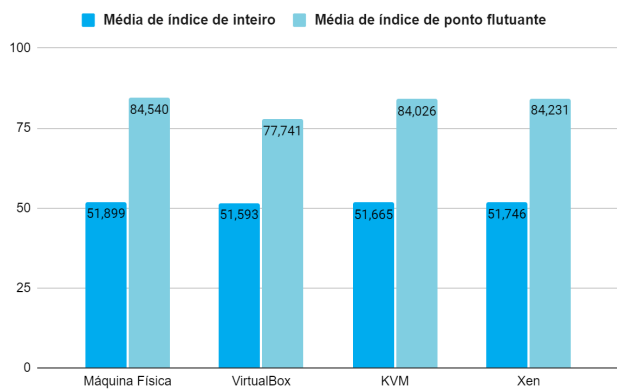


Fig 3. Teste de desempenho de CPU Nbench

Observa-se na Figura 3 que os resultados de índice para inteiros não apresentaram variações significativas. Para números de ponto flutuante, o hipervisor que mais se aproxima do resultado (ideal) é o Xen, enquanto que o VirtualBox apresentou o pior desempenho. Considerando que a boa parte das aplicações científicas que envolvem cálculos utilizam-se de ponto flutuante, a melhor alternativa é o uso do Xen.

O benchmark de memória do Sysbench é usado para comparar leituras/gravações de memória sequencial. Dependendo das opções da linha de comandos, cada *thread* pode acessar um bloco global ou local para todas as operações de memória. Nos testes realizados foram acessados

blocos globais. Os parâmetros variados nestes testes foram: *--memory-block-size* que é equivalente ao tamanho do bloco de memória a ser usado e *--memory-total-size* que é equivalente ao tamanho total dos dados a serem transferidos [8].

Foram realizados diversos testes com valores diferentes para ambos os parâmetros. Nesse caso, os resultados variaram bastante, para cada mudança de parâmetro. As figuras 4 e 5, mostram a variação dos resultados de dois destes testes.

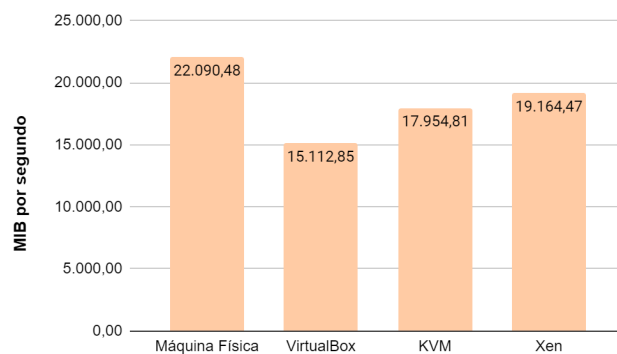


Fig 4. Teste 1 de desempenho de memória Sysbench

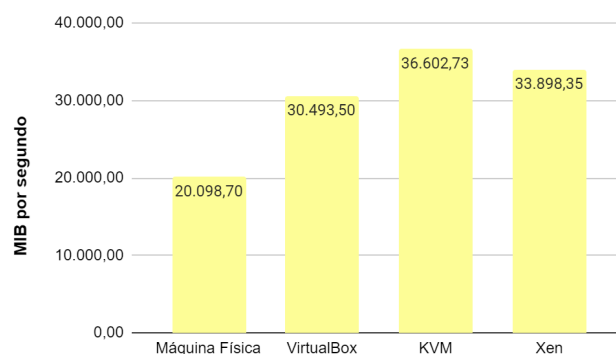


Fig 5. Teste 2 de desempenho de memória Sysbench

Os resultados apresentados na Figura 4 indicam o Xen como melhor hipervisor pois foi o que mais se aproximou do resultado da máquina física (ideal) para bloco de tamanho 1M, memória de 100M e 2 *threads*.

Por sua vez, a Figura 5 indica melhor desempenho para o hipervisor KVM, nos testes usando blocos de tamanho 1M, memória de 1G e 2 *threads*. Portanto, entende-se que este benchmark não obteve resultados conclusivos ao avaliar o desempenho de memória dos ambientes. Inclusive o fato das máquinas virtuais apresentarem desempenho melhor do que a máquina física deve ser melhor avaliado posteriormente.

O benchmark Nbench BYTEmark gera o índice de memória a partir da média geométrica dos testes *String sort*, *Bitfield* e *Assignment*, e quanto maior for o resultado obtido, melhor. Na Figura 6 encontra-se os resultados do teste de memória deste benchmark

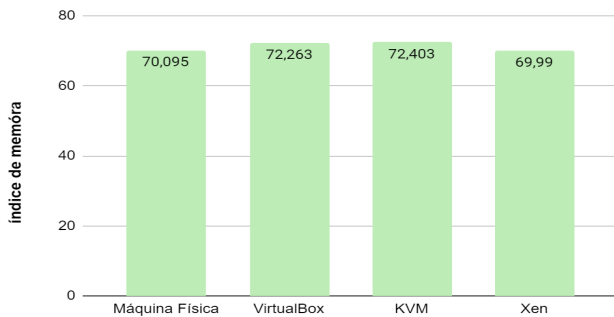


Fig 6. Teste de desempenho de memória Nbench

Nota-se que o KVM obteve o maior resultado, isso ocorre devido a sua técnica que faz com que o controle do mapeamento de memória fique no sistema operacional visitante fazendo com que o hipervisor não fique sempre rastreando o estado da tabela de memória do visitante para emular alguma alteração, reduzindo assim a sobrecarga. Assim, é o melhor hipervisor para aplicações que utilizam intensivamente recursos de memória (*memory bound*).

V. CONCLUSÃO

De acordo com os resultados pode-se concluir que para cada tipo de recurso utilizado (CPU ou memória) existe um ambiente de virtualização mais favorável e isso pode ser levado em consideração para melhorar o desempenho dos experimentos científicos.

Para as aplicações que utilizam ponto flutuante o recomendado é utilizar virtualização baseada no hipervisor Xen. Para aplicações com números inteiros os resultados mostram que o desempenho é similar para todas as soluções de virtualização, portanto, não é um fator de impacto na escolha. Considerando aplicações com uso intensivo de memória, o hipervisor KVM mostra-se a solução de virtualização.

Considerando esses resultados, a implementação da nuvem privada voltada para pesquisa pode considerar o uso das tecnologias Xen e KVM. Uma alternativa é utilizar o hipervisor mais adequado para a carga computacional de cada aplicação.

Trabalhos futuros incluem a análise da viabilidade da implantação dessa solução empregando múltiplos hipervisores.

AGRADECIMENTOS

A Fundação Araucária, pelo financiamento deste projeto de iniciação científica.

REFERÊNCIAS

- [1] A. S. Tanenbaum, H. Bos. “Modern Operating Systems”. 4th EDITION. Pearson, 2015.
- [2] D. T. Vojnak, B. S. Đorđević, V. V. Timčenko, S. M. Štrbac, “Performance Comparison of the type-2 hypervisor VirtualBox and VMWare Workstation”. 27th Telecommunications Forum (TELFOR), 2019.
- [3] Oracle Corporation. “Oracle VM VirtualBox User Manual”. Disponível em: <https://download.virtualbox.org/virtualbox/6.0.8/UserManual.pdf>.
- [4] Red Hat. “KVM: Máquina virtual baseada em Kernel”. Disponível em: <https://www.redhat.com/pt-br/topics/virtualization/what-is-KVM>.
- [5] T. Deshane, Z. Shepherd, J. N. Matthews, M. Ben-Yehuda, A. Shah and B. Rao, “Quantitative comparison of Xen and KVM”. Xen Summit, 2008.
- [6] N. A. Ramos, “Avaliação e Comparação de Desempenho de Computadores: Metodologia e Estudo de Caso”. Trabalho de Graduação, Centro de Informática da Universidade Federal de Pernambuco, 2008.
- [7] A. Kopytov, “Sysbench”. Disponível em: <https://github.com/akopytov/sysbench>.
- [8] A. Kopytov, “SysBench manual”. Disponível em: <https://imysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf>.
- [9] U. F. Mayer, “Linux/Unix Nbench”. Disponível em: <http://www.math.utah.edu/~mayer/linux/bmark.html>.
- [10] U. F. Mayer, “BYTEmark documentation”. Disponível em: <http://www.math.utah.edu/~mayer/linux/byte/bdoc.pdf>.