Avaliando o Ambiente de Virtualização Xen Utilizando Aplicações de Bancos de Dados*

Rodrigo N. Calheiros, Guilherme Rodrigues, Tiago Ferreto, César A. F. De Rose
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Faculdade de Informática
Av. Ipiranga, 6681 – Porto Alegre, RS – Brasil
{rcalheiros,grodrigues}@inf.pucrs.br, {tiago.ferreto,cesar.derose}@pucrs.br

Resumo

Embora seja uma tecnologia utilizada há várias décadas, a virtualização ainda apresenta desafios na sua aplicação prática, como por exemplo determinar como a divisão de CPU entre máquinas virtuais do usuário e a máquina virtual de acesso ao hardware do sistema afetam o desempenho das aplicações hospedadas. Neste artigo, diferentes configurações de divisão de CPU entre máquinas virtuais Xen são testadas e o desempenho do SGBD MySQL executando o benchmark OSDB é avaliado para cada uma delas. Também é realizada uma análise da forma como recursos são consumidos pelo domínio de Driver (Domínio 0 do Xen) durante a execução da aplicação.

1. Introdução

O conceito de virtualização não é novo e tem princípios na década de 60 com o VM/370 da IBM que já utilizava tal técnica [5]. Atualmente ela é empregada em diversas áreas da computação, que vão desde serviços de suporte que utilizam vários sistemas até a área de alto desempenho. Isto ocorre porque a virtualização melhora a escalabilidade do sistema de maneira geral [15]. Esta tendência tem fomentado o interesse de grandes empresas no desenvolvimento de tecnologias específicas para implementação de sistemas virtuais, dentre as quais podemos citar os esforços da Intel para desenvolver e lançar no mercado uma família de processadores adequados a virtualização [11].

A virtualização permite que um único computador (*host*) hospede um ou mais ambientes, proporcionando vantagens como melhor utilização dos recursos, consolidação dos servidores, segurança através do isolamento entre diferentes máquinas virtuais (VM's), tolerância a falhas e balanceamento de carga [9]. Uma das formas de implementar virtualização é através da *paravirtualização*, onde a virtualização da plataforma de hardware não é completamente abstraída ao sistema operacional que roda sobre o virtualizador [2].

O intuito deste trabalho é analisar o efeito da divisão de tempo de processamento entre a máquina virtual de acesso ao hardware e a máquina virtual que roda aplicações de usuários afeta o desempenho destas aplicações. Especificamente, este efeito da divisão de CPU será analisado em um servidor de banco de dados, o MySQL Server [10], executando em um ambiente virtualizado com o Xen VMM [2]. Os testes foram realizados com o benchmark de banco de dados OSDB.

Este artigo está organizado da seguinte forma. A Seção 2 apresenta os conceitos de virtualização, paravirtualização e o Xen VMM. Na Seção 3 é descrito o benchmark utilizado para testar o sistema. Na Seção 4 são descritos os ambientes de teste, e os resultados são analisados. A Seção 5 apresenta os trabalhos relacionados e a Seção 6 conclui o trabalho e apresenta os trabalhos futuros.

2. Virtualização e Paravirtualização

A virtualização de plataformas computacionais – o mapeamento da interface de algum sistema sobre a interface do próprio ou de um outro sistema [14] – não é uma idéia nova. De fato, desde a década de 60 a técnica é empregada em *mainframes* IBM [5].

Neste artigo, o termo *virtualização* é utilizado como sinônimo de *virtualização de sistema*. Esta é a técnica que permite que um único computador (*host*) hospede um ou mais ambientes[1]. Cada um destes ambientes executa um sistema operacional completo, isto é, cada sistema operacional enxerga a sua própria área de memória, seu tempo de

*Este trabalho foi desenvolvido em colaboração com a HP Brasil P&D.

[1]No decorrer do texto, utilizaremos os termos ambiente, máquina virtual e VM como sinônimos.

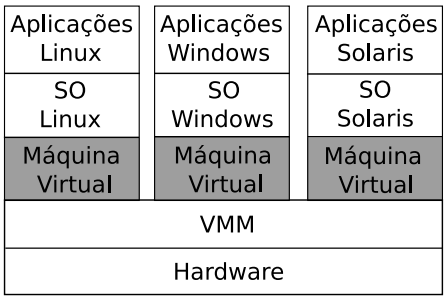


Figura 1. Exemplo de ambiente virtual.

processamento de CPU, os recursos do sistema e assim por diante. Embora cada VM perceba os recursos como sob seu controle, eles são compartilhados entre os demais ambientes hospedados no *host* e controlados por uma camada de software chamada VMM (*Virtual Machine Monitor* – Monitor de Máquina Virtual) ou *hypervisor* [14].

O VMM controla o acesso das VMs aos recursos (incluindo memória e tempo de CPU) da máquina. Esta arquitetura é mostrada na Figura 1. A figura representa um *host*: ele possui seu hardware, que está abaixo do VMM. Sobre o VMM existem um ou mais ambientes, cada um com o seu próprio sistema operacional e com o seu hardware virtual disponível às aplicações dos usuários (executando sobre o sistema operacional).

Os sistemas operacionais que executam sobre um virtualizador são sistemas operacionais desenvolvidos para uma determinada plataforma de hardware, e em princípio não são modificados para serem utilizados no ambiente virtual. A paravirtualização [2] é um tipo especial de virtualização de sistema onde a virtualização da plataforma de hardware não é abstraída ao sistema operacional que roda sobre o virtualizador: com o objetivo de aumentar o desempenho do sistema, o sistema operacional que executa sobre o paravirtualizador é modificado a fim de tornar-se ciente da presença do mecanismo e assim ser capaz de suportá-lo. No entanto, nenhuma modificação é exigida nas aplicações dos usuários.

2.1. Xen VMM

O Xen é um paravirtualizador livre desenvolvido originalmente para a arquitetura IA-32, mas que atualmente suporta também x86/64, Itanium e PowerPC. Por se tratar de um paravirtualizador, os sistemas operacionais executando nas VMs devem ser adaptados para serem compatíveis com o VMM. No entanto, recentes versões do Xen suportam, por exemplo, o Windows XP sem modificações se for utilizado, sob o virtualizador, um processador com suporte a virtualização. Os processadores "Pacifica" da AMD [1] ou os processadores Intel com tecnologia "VT" [11].

O Xen VMM coordena a execução de uma ou mais VMs. As máquinas virtuais no Xen (também conhecidas como domínios) podem ser de dois tipos. O primeiro contém máquinas virtuais que executam aplicações de usuários e são conhecidos como *DomU* – de *unprivileged*. Estas máquinas não têm acesso direto ao hardware real da máquina. O segundo (*Domain 0* ou simplesmente *Dom0*) é responsável pela gerência das demais VMs e pelo acesso ao hardware da máquina. Apenas um *Dom0* executa em um *host*. Também é possível, através do *Dom0*, monitorar o consumo de recursos de cada uma das VMs [2].

Quando uma VM é criada deve ser especificada a quantidade de memória inicial do sistema. A quantidade de memória destinada a uma VM pode ser alterada (via *Dom0*) de forma dinâmica. A quantidade de CPU destinada a VM é ajustada por meio de "processadores virtuais" delegados a esta VM. O *hypervisor* possui um escalonador que controla a distribuição de tempo de CPU entre as máquinas virtuais. A política de escalonamento utilizada pode ser determinada em tempo de *boot* do VMM. A quantidade de CPU destinada a cada VM pode ser alterada dinamicamente através do *Dom0*.

O *Dom0* é também um *Domínio de Driver Isolado* (*Isolated Driver Domain* – IDD), isto é, um domínio que possui acesso ao hardware do sistema através de *drivers* para sistemas operacionais comuns. Portanto, todo o acesso a dispositivos de entrada e saída passa pelo *Dom0*. Porém, este é também uma VM e portanto está sujeita a mesma política de escalonamento que as demais máquinas virtuais do sistema.

Se for delegado pouco tempo de CPU a este domínio, ele terá pouco tempo para processar a entrada e saída das demais VMs e o desempenho das aplicações em execução poderá piorar. Se for delegado ao IDD tempo demais, as VMs receberão menos tempo de processamento, e as aplicações também podem perder desempenho. Portanto, um ajuste preciso da divisão de processamento entre o IDD e os demais domínios é fundamental para um bom desempenho das aplicações executando sobre VMs.

Visando uma melhor compreensão das causas deste fenômeno, com o objetivo de determinar uma configuração que apresente um ótimo desempenho sem a realização de testes empíricos, foram realizados experimentos utilizando como serviço um Sistema de Gerenciamento de Banco de Dados exercitado com um benchmark para tal tipo de aplicação. Tal benchmark é descrito na seção seguinte.

3. OSDB

O OSDB [12] é um benchmark para sistemas de gerenciamento de bancos de dados (SGBD). Ele é baseado no AS^3AP, um benchmark para bancos de dados relacionais que por sua vez é baseado no padrão ANSI SQL [16]. Os testes realizados pelo OSDB incluem criação de tabelas,

criação de índices, população de bancos de dados, seleções, junções e atualizações para testes com único usuário e teste com vários usuários.

O OSDB é projetado para ser utilizado na mesma máquina que hospeda o banco de dados ou em uma máquina remota, através de dois testbeds: o *Mixed On-Line Transaction Processing* (Mixed OLTP), que avalia a vazão do servidor na realização de atualizações em um banco de dados e o *Mixed Information Retrieval* (Mixed IR), que mede a vazão do servidor do banco de dados retornando consultas. No primeiro teste, o OSDB realiza tantas consultas quanto possíveis dentro do tempo de execução do teste. No segundo, são realizadas tantas atualizações quantas possíveis dentro do intervalo de tempo de duração do teste. O OSDB fornece como métrica de saída o número de tuplas processadas em cada teste durante a duração do benchmark para os testes de multi-usuários, que duram 5 minutos, para os testes Mixed IR e Mixed OLTP.

3.1. Perfil da Aplicação

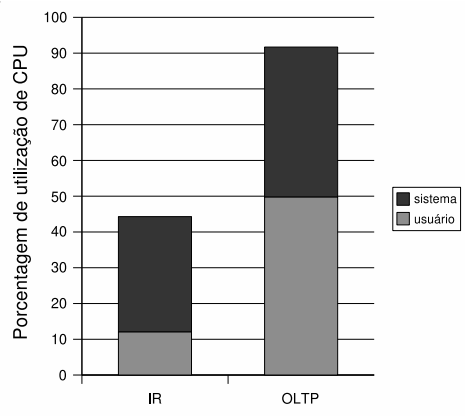
Com o intuito de permitir uma melhor compreensão dos fenômenos observados nos experimentos da seção seguinte, foi realizada uma análise da execução do OSDB. Para tanto, foi utilizado o sistema de monitoração Ganglia [8].

A máquina utilizada para hospedar o SGBD é um Pentium 4 a 2.6GHz com 256MB de memória RAM, a fim de deixá-la com a mesma quantidade de recursos disponíveis para a VM que executava o SGBD no teste com o Xen (descritos na seção seguinte). O servidor em questão executa diretamente o Debian GNU/Linux Etch e utiliza o MySQL 4.1.

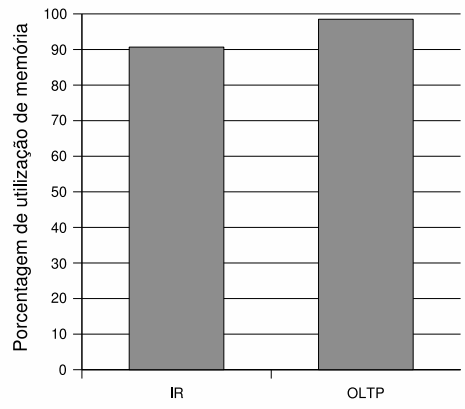
O SGBD é o único serviço em funcionamento no servidor durante os testes. A rede que interliga o servidor ao cliente é dedicada, sendo que trata-se de uma rede *Fast Ethernet* chaveada. O cliente é uma máquina Pentium 3 a 800MHz com 256MB de memória RAM executando o sistema operacional Debian GNU/Linux Sarge.

As Figuras 2(a), 2(b) e 2(c) mostram, respectivamente, o consumo de CPU, memória e o fluxo de dados pela rede realizada pela aplicação. Percebe-se, por estas figuras, que o teste Mixed OLTP é uma aplicação intensiva em uso de CPU (a utilização de CPU ultrapassa 90% durante a sua execução), mas que transfere menos dados pela rede que a Mixed IR. Esta, por sua vez, não utiliza toda a CPU a ela destinada. O OLTP também utiliza mais memória que o IR, fazendo com que o consumo de memória disponível atinja quase 100% durante a execução deste testbed.

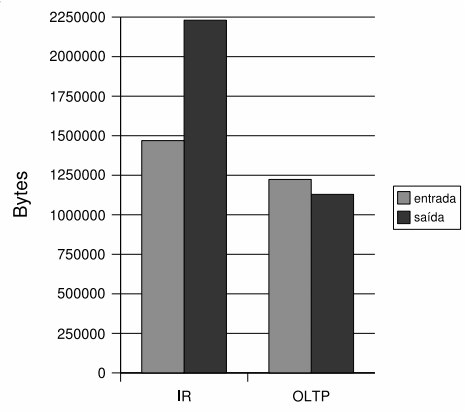
Com base neste perfil das aplicações é que são feitas as considerações sobre o desempenho da aplicação utilizando o Xen apresentadas na seção a seguir.



(a)



(b)



(c)

Figura 2. Utilização de recursos pelo OSDB. (a) CPU (%) (b) Memória real (%) (c) Rede (bytes)

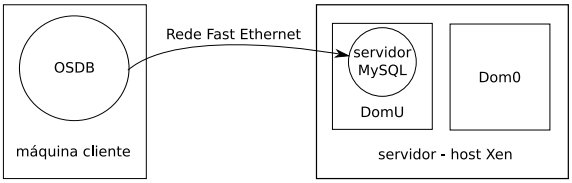


Figura 3. Ambiente utilizado nos testes.

4. Experimentos Realizados

Nesta seção serão descritos os testes que foram realizados e as conclusões que puderam ser obtidas a partir da análise dos seus resultados.

4.1. Ambiente de Testes

O ambiente de testes consiste em uma máquina cliente de onde foi executado o OSDB e uma máquina servidora rodando o Xen e hospedando o SGBD. Com o intuito de tornar os testes mais homogêneos, foram utilizadas as mesmas máquinas do teste da seção anterior (Figura 3). No entanto, o servidor teve a sua memória aumentada para 1GB e foi adicionado o Xen versão 3.0.1.

A máquina virtual que hospeda o SGBD possui 256MB de memória RAM e possui sistema operacional Debian GNU/Linux Etch. O SGBD usado é o MySQL 4.1. O *Dom0* correspondente possui também 256MB de memória RAM. As imagens dos sistemas operacionais são armazenadas em disco e carregadas durante o *boot* da VM. O escalonador utilizado no Xen é o SEDF (Simple Earliest Deadline First). Este escalonador permite que seja determinadas partições fixas de tempo de CPU, em que não é possível a uma VM utilizar tempo de processamento não utilizado de outra VM. Desta forma, foi possível assegurar que o desempenho medido é somente decorrente do tempo delegado pelo escalonador a cada VM com base na divisão de CPU sendo avaliada. Se tal ação não tivesse sido tomada, a divisão de CPU configurada não corresponderia a divisão de CPU efetivamente recebida pelos processadores, o que invalidaria os resultados aqui apresentados.

Cabe ressaltar novamente que durante os experimentos, não havia nenhum tráfego na rede que interliga as máquinas clientes e servidora a não ser o tráfego gerado pelo benchmark. Desta forma, evitou-se interferência nos resultados que poderiam ser ocasionados pela carga na rede.

Para a monitoração da utilização dos recursos do IDD também foi utilizado o Ganglia.

4.2. Testes

Os testes realizados consistiram na execução do benchmark OSDB na máquina cliente, realizando consultas sobre o SGBD executando no servidor com o Xen. O único serviço executado pelo servidor durante a realização dos testes era o próprio MySQL. Os workloads executados pelo OSDB são o Mixed IR e o Mixed OLTP, para um número de usuários igual a 5.

Antes da execução do benchmark, o tempo de CPU destinado a cada domínio era configurado, de forma a refletir uma diferente distribuição de tempo entre o *Dom0* e a VM executando o MySQL. Cada uma das configurações, com a respectiva divisão de tempo (quota) entre cada domínio, é mostrada na Tabela 1. Por exemplo, uma configuração onde a quota do *Dom0* é 2 e a quota do *DomU* é 1 significa que o primeiro recebe duas vezes mais tempo de CPU do que o segundo. A quota de 1 e 1 significa que ambos os domínios recebem o mesmo tempo de CPU e assim sucessivamente. A divisão de tempo no SEDF pode ser configurada para ser realizada exatamente desta forma (com as taxas relativas para cada domínio) através do *Dom0*.

Tabela 1. Quotas de CPU utilizada nos experimentos.

# experimento	*Dom0*	*DomU*
#1	1	1
#2	1	2
#3	1	5
#4	1	10
#5	2	1
#6	2	5
#7	5	1
#8	5	2
#9	10	1

Para monitorar o consumo de recursos pelo IDD (Dom0) durante a execução dos testes, os *daemons* do Ganglia executavam no *Dom0*, capturando os dados deste domínio. Os dados de monitoração eram obtidos em intervalos de 15 segundos. As métricas capturadas foram:

- utilização de CPU em processos de usuário;
- utilização de CPU em processos de sistema;
- tempo de CPU livre;
- quantidade de memória real livre;
- bytes recebidos;
- bytes enviados.

Os testes foram repetidos 10 vezes, e os resultados apresentados são a média dos valores obtidos em cada um dos experimentos, para cada métrica de monitoração e para cada métrica de saída do OSDB. O coeficiente de variação em

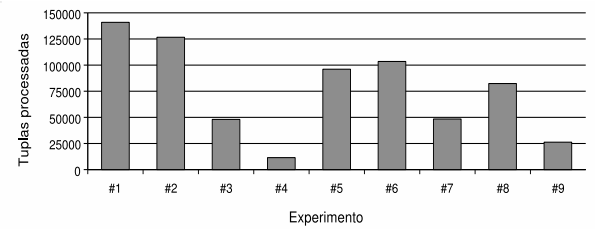


Figura 4. Tuplas processadas pelo OSDB – Mixed IR.

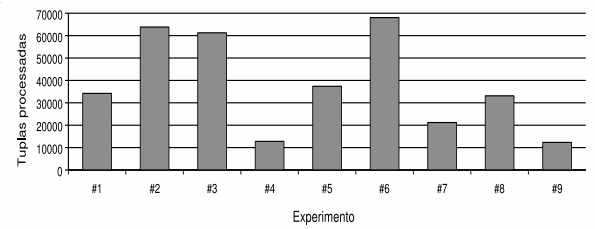


Figura 5. Tuplas processadas pelo OSDB – Mixed OLTP.

cada teste manteve-se abaixo de 25%, e na maioria deles não ultrapassou 5%, o que nos permite considerar os resultados válidos apesar do pequeno número de medições realizadas.

4.3. Análise de Desempenho

As Figuras 4 e 5 mostram a quantidade de tuplas geradas na execução do OSDB, respectivamente para os testes Mixed IR e Mixed OLTP, para cada quota de CPU.

As configurações de divisão de CPU entre domínios (descritas na Tabela 1) são apresentadas nas figuras em ordem decrescente de quantidade de tuplas geradas.

O primeiro resultado do experimento é a perceptível diferença de desempenho obtida entre a melhor e a pior configuração: no caso do Mixed IR, a diferença entre a melhor e a pior configuração foi de 93,6%, e no Mixed OLTP esta diferença foi de 81,9%.

Também é evidente, pela inspeção das Figuras 4 e 5, que o melhor desempenho da aplicação não foi obtido, para cada teste, com a mesma configuração. Embora a configuração #6 tenha apresentado um bom desempenho nos dois testes, no caso do Mixed IR as configurações #1 e #2 apresentaram resultado superior. Porém, no Mixed OLTP a configuração #1 apresentou um resultado 49,7% pior do que a melhor configuração para este teste.

Nota-se também que os casos com maior discrepância entre quotas para o IDD e para a VMM (configurações #4 e #9, que apresentam quotas na razão de 10:1, respectivamente para o *DomU* e para o *Dom0*) apresentaram desempenhos baixos em ambos os testes. Portanto, diferenças tão grandes na distribuição de tempo de processamento devem ser evitadas.

Embora estes dados revelem que a distribuição de CPU entre as VMs altera o desempenho das aplicações, e que aplicações diferentes necessitam de proporções diferentes para que apresentem um máximo desempenho, esta análise inicial não permite determinar o motivo de tal comportamento e nem se é possível a determinação prévia de uma configuração ótima sem a realização de testes empíricos. A análise da monitoração do IDD (*Dom0*), apresentada a seguir, é mais um passo em direção à obtenção destas respostas.

4.4. Análise da Monitoração

A Figura 6 mostra os valores médios do consumo de CPU executando tarefas de usuários e executando tarefas de sistema. A Figura 7, por sua vez, apresenta a quantidade média de bytes enviados e recebidos pela rede para o Mixed IR.

Analisando as métricas de CPU percebe-se que em nenhum momento a quantidade CPU destinada ao *Dom0* chegou a 100% de utilização. Portanto, a causa da perda de desempenho não pode ser creditada à falta de recursos no sistema para executar este benchmark. Os gráficos de consumo de memória não são mostrados, mas a quantidade de memória utilizada pela *Dom0* em todos os cenários foi equivalente: tanto para o Mixed IR quanto para o Mixed OLTP, a média de utilização foi de 97,8%. A maior utilização média de memória foi de 98,3% e a menor foi de 97,6%.

Para o teste Mixed IR, o pior desempenho foi observado na configuração #4. Nesta configuração, a CPU apresenta o menor tempo ocioso entre todas as configurações, e passa mais de 60% do seu tempo executando tarefas do usuário. Pode-se concluir, com estes dados, que, por receber muito pouco tempo de CPU (em comparação ao *DomU*), o IDD não tem tempo suficiente para processar todas as requisições de E/S que chegam do outro domínio, que é capaz de processar rapidamente as consultas mas precisa esperar pela consulta ao banco (operação de E/S realizada pelo IDD). Com isto, grande parte do tempo é gasto no processamento destas requisições, e o IDD não possui CPU suficiente para processá-las adequadamente, e o resultado é uma queda no desempenho do SGBD, que é evidenciado pela baixa vazão de E/S pela rede neste caso.

Quando o inverso ocorre, isto é, o IDD possui muito mais tempo que o *DomU* para processar, a CPU destinada a este domínio fica 90% do tempo ociosa: isto ocorre porque neste caso o SGBD possui pouco tempo de CPU para processar as consultas, e com isso o IDD fica com pouca atividade, pois não são gerados pedidos de E/S suficientes para ocupá-

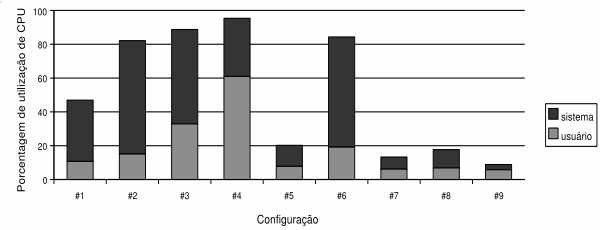


Figura 6. Utilização de CPU (%) – Mixed IR.

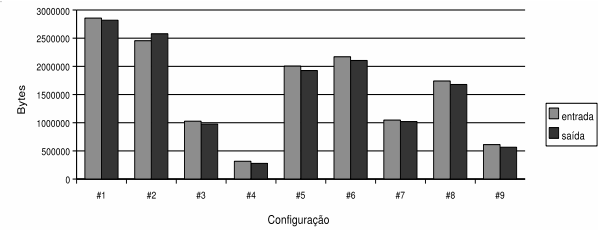


Figura 7. Bytes enviados e recebidos – Mixed IR.

lo. Novamente, neste caso é percebida uma baixa quantidade de dados enviada e recebida. Isto foi observado nas configurações #7 e #9 para o Mixed IR.

Nas configurações onde foi observado um bom desempenho do SGBD executando o Mixed IR, percebe-se um relativo equilíbrio na distribuição de quotas de CPU (configurações #1, #2, #5, #6 e #8), o que resulta em uma boa capacidade do SGBD processar as consultas e uma boa capacidade do IDD de processar a E/S gerada.

As métricas observadas para o teste Mixed OLTP são mostradas nas Figuras 8 e 9. Para este teste, é possível perceber que em algumas configurações a CPU não teve tempo ocioso. Outra relação interessante é que as configurações onde o teste apresentou pior desempenho são aquelas onde o tempo de CPU ocioso foi de 0% e para os que a CPU ficou mais de 90% ociosa.

Neste teste, as configurações #3 e #4, que privilegiam o *DomU*, mantiveram a CPU ocupada 100%. Mesmo tanto tempo ocupado, na configuração #3 o IDD recebeu pro-

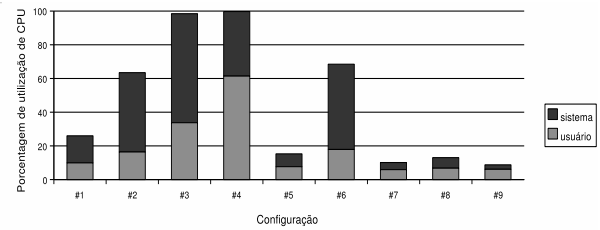


Figura 8. Utilização de CPU (%) – Mixed OLTP.

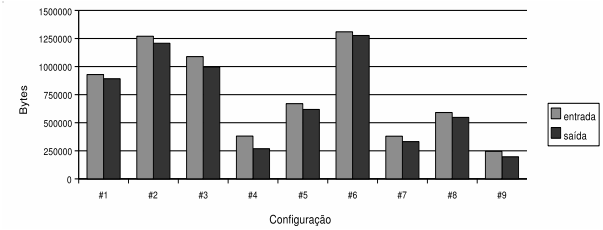


Figura 9. Bytes enviados e recebidos – Mixed OLTP.

cessamento suficiente para processar um volume de dados maior do que a quota de CPU da configuração #4 permitiu. Com isto, a quantidade de dados enviados na configuração #3 foi maior, o que refletiu em um desempenho melhor do benchmark.

A quantidade de dados recebida pelas VMs no Mixed OLTP é menor do que a quantidade recebida no Mixed IR. Porém, quantidade de dados de saída gerada é maior. Na configuração #9, onde a quantidade de CPU recebida pelo IDD é 10 vezes maior, o SGBD não conseguiu processar requisições suficientes para atender à demanda do benchmark, o que resultou em baixa utilização de CPU no IDD e pouco volume de dados trafegado. O resultado é o pior desempenho do OSDB para o Mixed OLTP. O mesmo fenômeno pode ser observado nas configurações #7 e #8. O melhor desempenho para este teste foi observado nas configurações que apresentaram um uso elevado da CPU em modo de sistema e um consumo em torno de 30% no modo usuário, com tempo de ociosidade em torno de 30% (configurações #2 e #6).

A configuração #3, que gerou o 3º melhor desempenho para o Mixed OLTP, possui um perfil de utilização de CPU diferente das que geraram os melhores resultados: o consumo de CPU em modo sistema foi elevado e a CPU não ficou ociosa. A explicação para o fato é que, neste caso, a configuração foi capaz de fornecer capacidade computacional estritamente suficiente para o IDD processar as requisições do usuário, e a quantidade de CPU no *DomU* era grande o suficiente para processar as consultas rapidamente, ocasionado um bom desempenho do servidor. De fato, esta é a característica de todas as configurações que possuem um relativo balanço entre quotas de CPU (#1, #2 e #5).

De maneira geral, podemos concluir que a simples análise da atividade de recursos no IDD não é suficiente para se fazer um diagnóstico preciso a respeito dos motivos que levam um sistema virtualizado à perda de desempenho. No entanto, a análise dos dados aqui apresentados permite que algumas conclusões sejam obtidas a respeito dos motivos da perda de desempenho.

A primeira conclusão é que o conhecimento da demanda

da aplicação hospedada no ambiente virtual pode dar uma boa pista sobre como o sistema deve ser configurado. Um tempo maior de CPU deve ser disponibilizado àquela VM que exigirá maior consumo de CPU. No entanto, a fatia de tempo não deve ser desproporcional a ponto de deixar pouco tempo para que o IDD possa processar a E/S do ambiente. Portanto, a quantidade de entrada e saída realizada pela aplicação também é um fator importante a ser considerado durante a configuração do sistema.

No entanto, na prática dificilmente uma aplicação mantém o mesmo padrão de comportamento durante toda a sua execução. Dependendo da carga submetida a um serviço, este pode apresentar em um momento um comportamento *CPU-bound* e em outro momento um comportamento *I/O-bound*.

Portanto, sistemas de configuração estáticos não serão capazes de manter o sistema em um estado que garanta o seu maior desempenho, uma vez que mudanças no perfil das requisições recebidas pelo servidor pode modificar os requisitos de CPU do *DomU* e do *Dom0*.

5. Trabalhos Relacionados

Em [7] é apresentado o resultado de um estudo que mostra que o desempenho das aplicações executando em VMs Xen pode variar dependendo da forma como a CPU é dividida entre o IDD e as VMs de usuários. Em tal estudo, foi utilizada uma versão anterior do Xen (a versão 3.0) que por sua vez utiliza o escalonador BRV (*Borrowed Virtual Time*). Nosso estudo complementa este trabalho prévio, mostrando que as versões posteriores do Xen continuam sendo influenciadas por esta divisão de CPU e, mais importante, o escalonador SEDF não é capaz de resolvê-lo. Também aqui são apresentados os resultados para mais configurações do que em [7] O software testado também é diferente em cada trabalho: enquanto aqui e apresentado o resultado do desempenho para um SGBD, o trabalho prévio avalia o desempenho de um servidor Web.

Outro estudo de desempenho do Xen é apresentado em [4], onde é avaliada a utilização de CPU do IDD durante operações de rede e de entrada e saída utilizando o SEDF. No entanto, tal estudo não considera o impacto da divisão de CPU nos seus resultados, pois o seu foco é na avaliação do tempo gasto pela CPU nas operações, na presença de uma ou de várias VMs. As únicas métricas avaliadas são utilização de CPU, tráfego na rede e quantidade de trocas de página de memória ocorridas. Nosso trabalho procura mostrar como este fenômeno influencia o desempenho das VMs, aspecto que não é considerado em [4].

Em [6] é apresentado mais um estudo sobre o desempenho de aplicações executando sobre o Xen, e complementa o estudo apresentado em [4]. Assim como o trabalho que o antecedeu, o trabalho foca no *overhead* do IDD e não considera particionamento da CPU entre as VMs. Um novo escalonador é proposto e uma ferramenta que considera o tempo que o IDD gasta com entradas e saídas de domínios específicos no tempo de CPU de tal domínio. Porém, o trabalho não considera como este particionamento afeta o desempenho do sistema como um todo.

Em [13] é proposta a utilização de técnicas de teoria de controle como uma maneira de contornar o tipo de fenômeno relatado neste artigo. O tempo de CPU destinado a cada VM é configurado dinâmica e continuamente a fim de manter uma determinada qualidade de serviço a cada VM. No entanto, o sistema não prevê a configuração do sistema de forma a extrair o máximo de desempenho do *host* a uma única aplicação, algo que um melhor entendimento sobre o impacto do tempo de CPU na performance permite que seja obtido.

Em [3] são propostos modelos de desempenho que descrevem o comportamento de sistemas virtualizados. No entanto, os modelos descrevem sistemas SMP, e não sistemas com um processadores compartilhados entre VMs. Os modelos foram validados com o Xen e o escalonador BVT. Se os modelos são ou não capazes de descrever sistemas com o escalonador SEDF, e como eles poderiam ser estendidos para sistemas com um único processador são assuntos que ainda devem ser investigados.

6. Conclusão e Trabalhos Futuros

Este artigo apresentou a relação entre recursos disponíveis às máquinas virtuais Xen e o desempenho de uma aplicação de banco de dados. Testes foram realizados com o objetivo de analisar o impacto da configuração de divisão de tempo de processamento sobre um servidor de banco de dados.

Os resultados permitiram observar que a configuração utilizada influencia diretamente no desempenho, pois a variação entre o pior e o melhor desempenhos para os experimentos realizados foi considerável nos dois tipos de testes com taxas muito expressivas.

Na continuação deste trabalho, pretende-se realizar estudos semelhantes com outros tipos de aplicações. Também propõem-se uma análise do consumo de recursos no domínio que hospeda a aplicação, a fim de se obter mais informações sobre os fatores que influenciam no desempenho de serviços executando sobre o Xen.

No entanto, os resultados aqui apresentados sugerem que um sistema virtualizado onde a configuração possa se adaptar de acordo com a aplicação poderia fornecer desempenho ótimo considerando a capacidade que o sistema virtualizado é capaz de fornecer. Portanto, o desenvolvimento de um protótipo capaz de realizar tal ação é um importante desdobramento dos resultados aqui apresentados.

Referências

[1] AMD. AMD64 virtualization codenamed "Pacifica" technology secure virtual machine architecture reference manual. Disponível em http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/33047.pdf, 2005.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauery, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proceedings of SOSP'03*, 2003.

[3] F. Benevenuto et al. Performance models for virtualized applications. In *Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops*, volume 4331 of *Lecture Notes in Computer Science*, pages 427–439. Springer, 2006.

[4] L. Cherkasova and R. Gardner. Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In *Proceedings of the 2005 USENIX Annual Technical Conference*, pages 387–390. USENIX, 2005.

[5] R. J. Creasy. The origin of the VM/370 time-sharing system. *IBM Journal of Research and Development*, 25(5):483–490, 1981.

[6] D. Gupta et al. Enforcing performance isolation across virtual machines in Xen. In *Middleware 2006*, volume 4290 of *Lecture Notes in Computer Science*, pages 342–362. Springer, 2006.

[7] D. Gupta, R. Gardner, and L. Cherkasova. XenMon: QoS monitoring and performance profiling tool. Technical Report HPL-2005-187, HP Laboratories Palo Alto, 2005.

[8] M. L. Massie, B. N. Chun, and D. E. Culler. The ganglia distributed monitoring system: Design, implementation, and experience. *Parallel Computing*, 30(7):817–840, 2004.

[9] D. A. Menascé. Virtualization: Concepts, applications, and performance modeling. In *Proceedings of the 2005 Computer Measurement Group Conference (CD-ROM)*. CMG, 2005.

[10] MySQL. Open source database. Disponível em http://www.mysql. com, 2007.

[11] G. Neiger et al. Intel virtualization technology: Hardware support for efficient processor virtualization. *Intel Technology Journal*, 10(3):167–177, 2006.

[12] OSDB. The open source database benchmark project homepage. Disponível em http://osdb.sourceforge.net/, 2007.

[13] P. Padala et al. Adaptive control of virtualized resources in utility computing environments. In *EuroSys 2007*, 2007.

[14] J. E. Smith and R. Nair. *Virtual Machines: Versatile platforms for systems and processes*. Morgan Kauffmann, 2005.

[15] A. S. Tanenbaum and A. S. Woodhull. *Sistemas Operacionais Projeto e Implementação*. Bookman, 2^a edition, 2000.

[16] C. Turbyfill, C. Orji, and D. Bitton. AS^3AP – an ANSI SQL standard scalable and portable benchmark for relational database systems. In J. Gray, editor, *The Benchmark Handbook for Database and Transaction Processing Systems*, chapter 5, pages 317–357. Morgan Kaufmann, 1993.