

# Performance Evaluation of Virtualization Technologies for Databases in HPC Environments

Timoteo Alberto Peters Lange, Paolo Cemim, Fábio Diniz Rossi, Miguel Gomes Xavier,  
Rafael Lorenzo Belle, Tiago Coelho Ferreto, Cesar A. F. De Rose  
Pontifícia Universidade Católica do Rio Grande do Sul  
Av. Ipiranga, 6681 - Porto Alegre/RS, Brasil  
timoteo.lange@acad.pucrs.br  
tiago.ferreto, cesar.derose {@pucrs.br}

**Resumo**—Modern database systems are applications that have a variable workload demanding a great share of the available resources during peak periods. To avoid subutilization of resources during periods of low load, virtualization technology allows this type of application to be provisioned dynamically. This technique has improved in the last decades and new approaches are emerging to improve performance, like operating system level database virtualization. This paper analyzes the characteristics of this approach compared to traditional virtualization techniques.

**Keywords**—Performance Analysis; Database; Virtualization; TPC-C; HPC;

## I. INTRODUÇÃO

Aplicações de alto desempenho (HPC) têm alta demanda por recursos computacionais [1]. No entanto, os recentes avanços em *hardware* e *software* e a crescente importância dos repositórios de dados, introduziram outros recursos de computação, como por exemplo a utilização de sistemas de banco de dados (SGBD) como recurso de HPC. Neste contexto, os SGBDs passam a integrar o conjunto de recursos das aplicações de HPC. No entanto, estes possuem os seus recursos de *hardware* limitados e há uma constante demanda de mais recursos para a execução dos SGBDs. Para estes sistemas, precisamos otimizar a gerência para a alocação eficiente dos recursos.

Para prover a crescente demanda de desempenho, os centros de dados com modelos tradicionais de gerência ampliavam sua estrutura tecnológica de *hardware* para suprir a demanda requisitada. Este modelo apresenta um alto custo de aquisição, tempo de gerência e conhecimento, que por muitas vezes não representa vantagens competitivas ao centro de dados, pois essa estrutura, em determinados períodos, recai em ociosidade ou subutilização de recursos.

Uma tecnologia que permite utilizar estes recursos de maneira mais equilibrada, é a virtualização [2]. Esta tecnologia oferece compartilhamento de recursos em um sistema computacional com a utilização de máquinas virtuais, e mantém em cada uma, um sistema computacional completo executando de forma concomitante e concorrente, mas sem interferência entre elas. A virtualização tem a capacidade de, dependendo da demanda computacional das aplicações, alocar de maneira dinâmica a quantidade de recursos necessários à aplicação ou mesmo, migrar esta

aplicação para equipamentos com mais ou menos recursos. Devido a essas características, pode ser considerada a infraestrutura ideal para execução de aplicações que demandam alto desempenho [3].

Portanto, SGBDs podem aproveitar todas as características inerentes aos ambientes virtualizados. Em modelos que utilizam virtualização, o usual é existir um banco de dados executando sobre uma máquina virtual, e este pode gerenciar a utilização dos recursos dependendo da demanda da aplicação [4]. Porém, com a tentativa de diminuir o ônus computacional causado pela camada de virtualização, surge uma nova abordagem em que o banco de dados não é mais executado sobre uma máquina virtual [5]. Nesta nova abordagem, existe a virtualização da instância do banco de dados, que faz referência ao banco de dados em execução. Assim sendo, este novo modelo utiliza conceitos de *containers*, em que o próprio sistema de gerenciamento de banco de dados executa como um *hypervisor*.

Neste contexto, o presente trabalho tem como objetivo avaliar o desempenho de execução de um sistema de banco de dados em um ambiente virtualizado de provisionamento. Para isso, foram realizadas medições e comparações de desempenho em ambientes que utilizam banco de dados sobre uma máquina virtual (BDVM) e banco de dados virtual (BDV). Como parâmetro de comparação, também foram executadas avaliações de desempenho em ambientes de banco de dados sobre um sistema operacional nativo (BDSO). Outra análise realizada é a utilização de recursos, como CPU, memória, rede e consumo de energia das abordagens estudadas.

Algumas pesquisas avaliam o desempenho do SGBD em ambientes virtualizados (BDVM) comparando com a execução sobre sistema operacional (BDSO) porém, em complemento às pesquisas existentes, no presente trabalho é realizado uma comparação dos ambientes tradicionais com a nova abordagem de utilização onde somente a instância de banco de dados é virtualizada (BDV).

Este artigo está organizado da seguinte forma: Na Seção II apresentamos o conceito de virtualização de banco de dados. Na Seção III apresentamos os experimentos realizados. Na Seção IV apresentamos os trabalhos relacionados. A conclusão é apresentada na Seção V.

## II. VIRTUALIZAÇÃO DE BANCO DE DADOS

SGDB é uma classe de sistema que tem características bem específicas e estas devem ser consideradas antes de virtualizar um servidor de banco de dados [6]. Esta Seção aborda uma visão geral sobre banco de dados em ambientes virtualizados (BDVM) e banco de dados virtual (BDV).

### A. Banco de dados

O banco de dados foi originalmente projetado com o objetivo de separar o armazenamento físico dos dados da sua representação conceitual e prover uma fundamentação matemática para identificar o esforço necessário no processo de acesso aos dados. Essa abordagem de implementação introduziu as linguagens de consulta de alto nível, que são uma alternativa às interfaces para as linguagens de programação e conseqüentemente, ficou mais rápido e fácil escrever novas consultas e obter o resultado extraído do banco de dados [6].

Um sistema de banco de dados é composto basicamente por dois componentes: um conjunto de programas responsáveis pelo gerenciamento do acesso aos dados e os dados. O primeiro possui interfaces para a criação e manipulação dos dados, além de outras funcionalidades, como o controle de segurança e integridade. Este conjunto de programas é chamado de Sistema Gerenciador de Banco de Dados (SGDB). A segunda parte de um sistema de banco de dados são os dados propriamente ditos, que normalmente estão dispostos em um ou mais arquivos seguindo uma estrutura física proprietária do SGDB [5].

SGDBs atuais tem como uma de suas características a independência das estruturas lógicas de dados, como tabelas, exibições e índices das estruturas de armazenamento físicas (estrutura de arquivos). Como as estruturas físicas e lógicas são separadas, o armazenamento físico de dados pode ser armazenado sem influenciar o acesso às estruturas lógicas. Uma analogia é renomear um arquivo físico de banco de dados e não renomear as tabelas armazenadas nele e acessadas a partir de uma instância de banco de dados.

A definição de instância é muito importante para a correta compreensão deste trabalho. Instância é a estrutura de acesso aos registros armazenados nos arquivos de dados pelos programas que formam o SGDB. Os valores dos parâmetros de uma instância podem ser definidos no momento de sua criação. Esses parâmetros irão alocar a quantidade de memória e recursos do sistema operacional que foi configurado e iniciar os processos responsáveis por manipular o banco de dados [6].

### B. Virtualização

A virtualização de recursos computacionais é uma técnica aplicada em várias áreas da ciência da computação, tais como tolerância a falhas, alto desempenho e banco de dados [7], [8], [9]. A virtualização de recursos normalmente adiciona uma camada de software intermediária entre as aplicações e o *hardware*. Tradicionalmente, essa camada intermediária é provida por um Monitor de

Máquinas Virtuais (MMV), também chamado de *hypervisor*, que permite a execução de múltiplas MVs (máquinas virtuais). Cada máquina virtual possui seu próprio SO (sistema operacional) e executa de forma isolada das demais.

Neste contexto, a utilização de banco de dados em ambientes virtualizados é tipicamente realizada através da execução de um SGDB no topo de uma máquina virtual. Essa técnica impõe um sobrecusto computacional devido as camadas de *software* utilizadas para prover a abstração de uma máquina virtual completa. Uma alternativa para diminuir o ônus dessa abordagem pode ser a virtualização em nível de banco de dados. A Figura 1 apresenta as diferenças entre virtualização em nível de máquina virtual e em nível de banco de dados. Em vez de virtualizar uma máquina inteira (com seu próprio sistema operacional hospedeiro), a virtualização em nível de banco de dados virtualiza apenas uma instância do banco de dados.

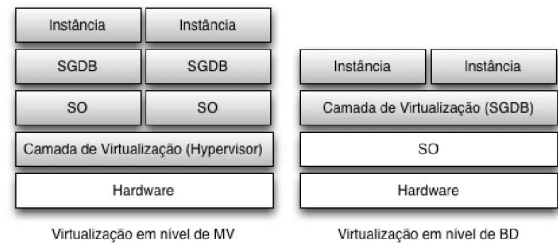


Figura 1. Comparação de virtualização em nível de MV e em nível de BD.

Esta técnica é similar à virtualização baseada em *containers* disponíveis em alguns sistemas operacionais modernos (por exemplo, Linux Containers [10], FreeBSD jails [11] e Solaris Zones [12]). A virtualização baseada em *containers* possibilita a criação de múltiplos ambientes virtuais, que funcionam de forma isolada, em nível de usuário e compartilhando o mesmo núcleo do sistema operacional. Este tipo de virtualização normalmente apresenta um desempenho bastante próximo ao nativo, já que não replica toda a pilha do sistema operacional [13]. De maneira análoga, a virtualização em nível de banco de dados permite a criação de múltiplas instâncias de banco de dados, que são isoladas entre si, sem a necessidade de ter uma máquina virtual que replica toda a pilha de sistema operacional e o próprio SGDB.

## III. EXPERIMENTOS REALIZADOS

Nesta seção serão descritos a estrutura do ambiente de avaliação, as avaliações que foram realizadas com suas respectivas observações que puderam ser obtidas a partir da análise desses resultados.

### A. Ambiente de teste

O ambiente de teste é formado por um servidor DELL, observado na Tabela I, interligado com *storage* através do protocolo ISCSI dedicado. A rede está separada em sub-redes distintas, uma rede pública, para acesso externo, e uma rede privada para comunicação direta entre *storage* e

host. O sistema operacional utilizado é OEL 5.6 (Oracle Enterprise Linux 5 update 6) com kernel 2.6.18-238.el5. O SGBD utilizado neste trabalho é Oracle® 11g r2. A arquitetura pode ser observada na Figura 2.

Tabela I  
AMBIENTES DE TESTES

Nome	Modelo/Característica
Servidor	DELL PowerEdge 810
Processador	2 Intel® Xeon® 6500 series
Memória RAM	64Gb
Storage	DELL EqualLogic PS400
Rede	GigaBit Ethernet

O ambiente de avaliação do BDSO foi configurado utilizando o OEL 5.6 sem suporte a virtualização. O SGBD foi configurado diretamente sobre o SO.

Durante a configuração do BDMV, foi instalado o sistema de virtualização VMware ESXi 5. A MV foi criada sobre o sistema de virtualização usando 100% dos recursos de *hardware* e instalado o sistema operacional OEL 5.6. O SGBD foi configurado diretamente sobre o SO convidado.

De forma semelhante, o BDV utilizou o OEL 5.6. O SGBD foi configurado diretamente sobre o SO e a instância foi criada usando o suporte à virtualização disponível no SGBD.

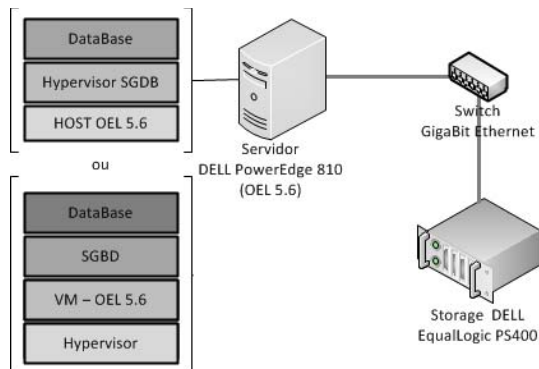


Figura 2. Estrutura do ambiente de avaliação

## B. Testes

Os testes foram realizados através da execução de *benchmark* específico (apresentado na Seção III-B1) para o domínio de banco de dados. O ambiente de teste foi configurado de maneira que apenas o SGBD estivesse em execução em cada caso de teste, além de que o sistema operacional estivesse sempre no mesmo estado inicial.

Foram realizadas 10 execuções do *benchmark* para cada caso de teste, com intervalo de confiança de 95%. Em todos os casos foi utilizada apenas uma instância do banco de dados de teste. Optamos por executar nossos testes com apenas uma instância, pois execuções com várias instâncias concomitantes e concorrentes são utilizadas em testes de isolamento ou escalabilidade, e o foco de nosso artigo é avaliar o *overhead* das camadas de virtualização de cada ambiente.

Em todos os testes, os recursos da arquitetura foram utilizados em sua totalidade, ou seja, considerando que as execuções dos testes foram realizados com apenas uma instância, essa instância utilizava 100% dos recursos de processador e memória.

1) *Benchmark TPC-C*: Avaliar o desempenho de um sistema de banco de dados não é uma tarefa naturalmente fácil, considerando a diversidade dos sistemas existentes e características de implementação [14]. Devido a essa diversidade, foi necessário a padronização dos métodos para avaliar o desempenho de um sistema de banco de dados, contemplando os mais diversos sistemas computacionais, dentre eles podemos citar OLTP (*On-Line Transaction Processing*) [15] e DSS (*Decision Support System*) [15].

Sistemas OLTP caracterizam-se por dar suporte a múltiplos usuários simultâneos, executando transações concorrentes em um SGBD, como por exemplo, operações de consulta, atualização e exclusão de dados. Avaliar este tipo de sistema permite identificar as operações típicas de ambientes que exigem grande disponibilidade de acesso simultâneo aproximando assim a execução às operações diárias executadas pelas empresas, como por exemplo: sistema bancário, comércio eletrônico e mineração de dados.

Definimos como métrica de desempenho para avaliar o *overhead* das camadas de virtualização, transações por minuto (TPM). Esta métrica é utilizada para medir desempenho em ambientes de banco de dados, permitindo que seja avaliada sua capacidade de processamento. Portanto, permite que seja avaliado o *throughput* diante de um *workload* pré-definido, que consiste em tamanho do banco de dados e o tempo de resposta desejado. O tamanho do banco de dados utilizado foi de 10Gb, criado pela própria ferramenta de *benchmark*, e com um tempo de resposta de 0.5 segundos para cada transação.

Com o objetivo de avaliar o desempenho de sistemas do tipo OLTP a *Transaction Performance Council* [15] definiu um *Benchmark* específico para isso, o *Benchmark TPC-C*. O TPC-C fornece a informação de quantas transações são concluídas em um intervalo de tempo. Existem 5 transações básicas que representam o comportamento de um sistema OLTP. Estas transações estão listadas na Tabela II, juntamente com qual é sua complexidade de execução no SGBD e qual é a porcentagem de requisição feita por cada cliente virtual instanciado na execução do TPC-C. As transações são realizadas concorrentemente. O número de clientes virtuais a ser utilizado e quantas transações cada cliente virtual vai realizar são parâmetros configuráveis do TPC-C.

O TCP-C pode ser implementado por qualquer *software*, desde que obedeça aos requisitos de implementação adotados pela TPC. Neste trabalho foi utilizado o software de *Benchmark Hammerora* [16].

## C. Resultados

Este trabalho aborda avaliação de desempenho de um sistema de banco de dados sobre uma máquina física (BDSO), sistema de banco de dados virtual (BDV) e

Tabela II  
TRANSAÇÕES TPC-C

Nome	Característica	Porcentagem
Novo Pedido	leitura-escrita, complexidade média	45%
Pagamento	leitura-escrita, complexidade baixa	43%
Estado do Pedido	leitura, complexidade média	4%
Entrega	leitura-escrita	4%
Nível de Estoque	leitura	4%

sistema de banco de dados sobre uma máquina virtual (BDVM) usando o *benchmark* TPC-C, que também pode ser definido como um conjunto de instruções para avaliar um sistema *OLTP* (abordado na Seção III-B1). Como resultado é esperado o total das execuções de transações por unidade de tempo, sendo que no contexto deste trabalho vamos utilizar transações por minuto (*TPM*). Os testes executados simulam a carga de 30 usuários realizando 100.000 transações cada um. Através dos testes realizados foram observadas diversas métricas de desempenho de cada ambiente, como CPU, memória, rede, consumo de energia e acesso ao disco.

1) *Análise de Desempenho*: A Figura 3 mostra a execução do *Benchmark* TPC-C para 30 usuários com 100.000 transações cada. Pode-se observar na Figura 3 a quantidade de transações executadas por minuto. No monitoramento de desempenho foram usados os relatórios de desempenho gerados pelo *Benchmark* Hammerora.

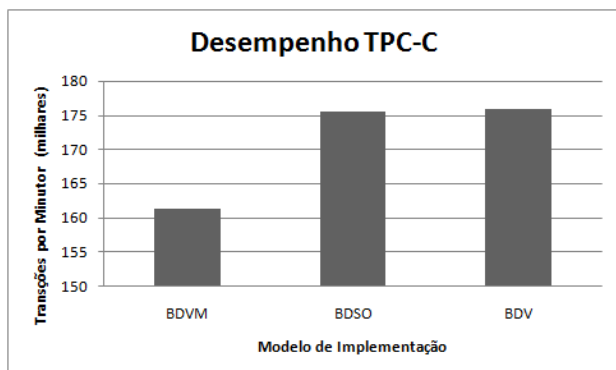


Figura 3. Transações por minuto

O desempenho do banco de dados executado diretamente sobre o *hardware*, neste contexto identificado como BDSO, em comparação com o mesmo banco de dados executado sobre uma MV é superior em 8%. Esse comportamento é esperado, como visto em [8] devido a inclusão da uma camada *hypervisor* necessária para a execução da MV. Além da inclusão da camada de virtualização, a quantidade de *Page-Faults* (páginas não encontradas) também afeta o desempenho da MV em até dois dígitos percentuais [7].

Também é mostrado o resultado da execução do banco de dados virtual, aqui identificado como BDV. É possível visualizar que o resultado de execução do BDV é equivalente à execução nativa de banco de dados (BDSO).

Esta análise inicial permite determinar que as execuções de um banco de dados virtual tem um desempenho próximo a de um banco de dados nativo. Tanto o BDV como o BDSO tem um desempenho superior em relação a execução de um banco de dados executado sobre uma MV. No entanto, apenas com esses dados ainda não é possível verificar o motivo da perda de desempenho do banco de dados sobre MV. A análise de recursos que é apresentada a seguir é mais um passo em direção à obtenção dessas respostas.

2) *Análise de Recursos*: Esta análise tem como objetivo avaliar a utilização de recursos, como CPU, memória, rede e energia. Comparando a utilização desses recursos, é possível identificar um dos motivos da diferença de desempenho. Nas Figuras 4, 5, 6 e 7 são apresentados a utilização de CPU, consumo de energia, tráfego de dados e a utilização de memória respectivamente. No monitoramento dos recursos foram utilizados as ferramentas *dstat* em conjunto com *sysstat* disponíveis à versão do SO utilizado.

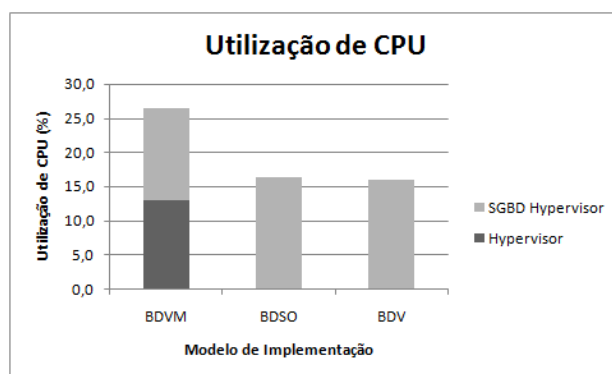


Figura 4. Utilização de CPU

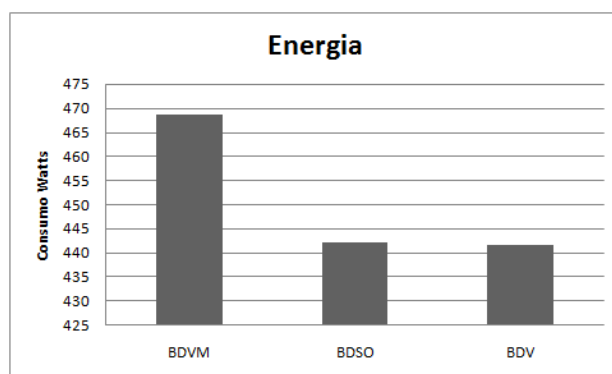


Figura 5. Consumo de energia

O primeiro resultado mostrado na Figura 4 representa uma diferença de 39% a mais de utilização de CPU entre o BDVM, comparado com a implementação do banco de dados virtual (BDV) e, a implementação do SGBD diretamente no *hardware* (BDSO).

Observamos que a utilização de CPU entre o BDV e BDSO é equivalente e se mantém na mesma proporção durante todo o monitoramento da execução da avaliação.



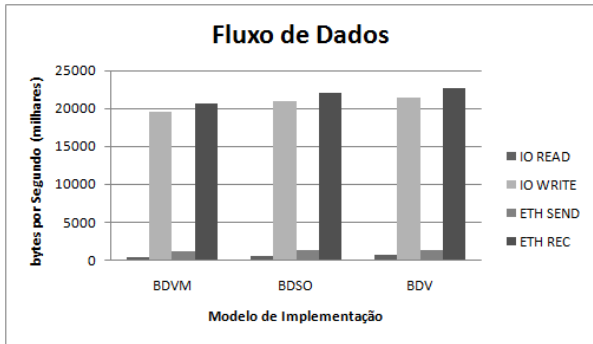


Figura 6. Fluxo de dados

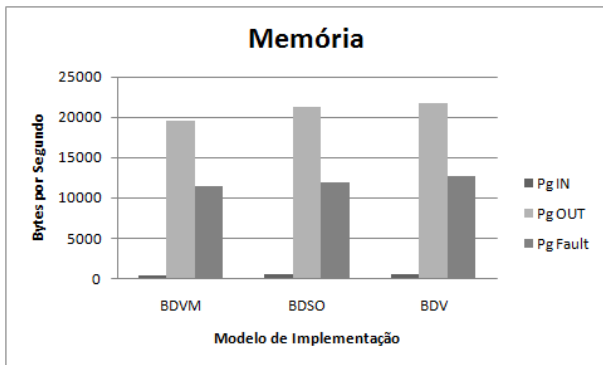


Figura 7. Utilização de memória

Comparando a execução dos três ambientes, o BDVM utilizou a mesma proporção de CPU que o BDV e o BDSO (identificada como SGBD *hypervisor*), porém, monitorando a utilização de CPU no *host* hospedeiro (identificada como *Hypervisor*) a utilização de CPU do BDVM é maior em relação a implementação do BDV e do BDSO devido a soma de utilização da MV (SGBD *Hypervisor*) com o *host* hospedeiro (*Hypervisor*). Considerando o processamento adicional devido a inclusão de uma camada extra entre a MV e o *hardware*, é possível identificar o esforço adicional executado pelo *hypervisor*, que converte as instruções da MV para o *hardware* [7].

O monitoramento do consumo de energia foi realizado através de um multímetro ligado diretamente na entrada de energia do servidor. Os dados registrados pelo multímetro eram repassados para uma estação que armazenava todas as entradas a cada segundo.

Este esforço adicional também pode ser observado na Figura 5 apresenta o consumo de energia. O consumo de energia do BDSO e BDV é nitidamente inferior ao consumo do BDVM. A média de consumo foi 9% inferior a mesma execução sobre a MV. Este comportamento é explicado devido a alta utilização dos recursos de *hardware*, como por exemplo CPU.

A utilização de memória é mostrada na Figura 7. Esse monitoramento foi realizado na camada do SGBD, ou seja, sobre a máquina nativa e sobre a MV. São mostrados, a quantidade de Páginas Carregadas (*Page-In*), de Páginas Descarregadas (*Page-Out*) e Páginas não Encontradas (*Page-*

*Fault*). Este último também pode ser considerado um dos motivos responsáveis pelo baixo desempenho de uma aplicação sobre VM [7]. Pode-se notar que a utilização de memória do BDVM é inferior em 9% comparado ao BDSO e BDV, porém a relação de utilização de memória entre BDSO e BDV é equivalente. A proporção de *Page-In*, *Page-Out* e *Page-Fault* é idêntica nas três execuções. Logo, não houve um acréscimo de utilização de memória do SGBD sobre a MV.

Embora que a utilização de memória do BDVM é inferior se comparado ao BDV e BDSO, isso não pode ser considerado uma vantagem dessa abordagem, haja visto que o BDVM teve o desempenho inferior e o consumo de energia superior.

Também é mostrado na Figura 6 a utilização de recursos de rede e disco. É observado que o volume de dados lidos e recebidos é superior em relação os dados escritos e enviados. Isso é devido ao tipo de *Benchmark* utilizado. O *TPC-C* mede o comportamento de aplicações OLTP, que conforme visto anteriormente, são aplicações que exploram os recursos do SGBD simulando transações essencialmente de compra e venda com múltiplos usuários. O BDVM tem uma utilização de rede e disco inferior em 8% comparado com as outras abordagens, assim como também é observado com a utilização de memória.

Da mesma forma, esse comportamento pode ser explicado pelo desempenho inferior do BDVM devido a inclusão da camada adicional de *hypervisor*. A proporção de leitura/escrita e envio/recebimento de dados é a mesma entre a BDVM, BDV e BDSO sendo que o primeiro é 8% inferior.

Analisando as figuras é possível determinar que o BDSO e BDV tem um desempenho equivalente sendo que este último ainda proporciona os benefícios que a virtualização oferece. O BDVM tem um desempenho inferior entre 8 a 10% devido a inclusão da camada adicional de virtualização. A Camada adicional de virtualização proporciona uma administração flexível do SGBD porém, o aumento do consumo dos recursos é nítido, conforme observado nas Figuras 4 e 5. O consumo por recursos de *hardware* refletem diretamente no consumo de energia. Além disso, a perda de desempenho é representada na Figura 3.

Tanto o BDVM quanto o BDV oferecem os mesmos benefícios proporcionados pela virtualização, entre eles podemos citar facilidade de administração das instâncias de banco de dados sobre a MV ou instância de banco de dados virtual, consolidação de recursos e escalabilidade. No entanto, para as aplicações de HPC que são naturalmente consumidores de recursos computacionais de alto desempenho, a utilização de sistema de banco de dados virtuais como um recurso para HPC pode ser tornar uma opção atrativa, já que o BDV tem o desempenho semelhante ao BDSO, porém com as mesmas facilidade do BDVM.

#### IV. TRABALHOS RELACIONADOS

A tecnologia de virtualização adiciona uma camada intermediária de *software* entre as aplicações e o *hardware*. Esta camada, chamada *hypervisor* ou monitor de máquinas virtuais (MMV), mapeia recursos virtuais visíveis às aplicações (no contexto deste trabalho, o SGBD) medindo os recursos físicos disponíveis no hospedeiro.

A virtualização pode resolver vários problemas importantes na área de banco de dados, como usabilidade, gerenciamento, escalabilidade, disponibilidade e implantação. Aplicações de banco de dados podem se favorecer desse tipo de infraestrutura através de algumas características como migração e consolidação, como pode ser visto em [9] [17] [18]. Assim, dependendo da demanda da aplicação de banco de dados, o ambiente pode se ajustar dinamicamente provendo mais ou menos recursos à aplicação.

Vários trabalhos analisam o *overhead* da camada de virtualização como em [3] [2] [19], que mostram uma perda de desempenho entre 7 e 11% se comparado com aplicações em um sistema operacional nativo. Alguns trabalhos como em [5] [8] avaliam o *overhead* da camada de virtualização com aplicações de banco de dados. Pode ser visto também que em algumas aplicações de alto desempenho o ganho de desempenho é superior ao *overhead* [20] [21], o que justifica a sua utilização neste tipo de ambiente.

O trabalho apresentado neste artigo se torna complementar aos anteriores, pois não avalia apenas o *overhead* da camada de virtualização, mas também compara um MMV tradicional com uma nova proposta que permite isolamento e segurança para aplicações, que é baseada em *containers* em nível de banco de dados.

#### V. CONCLUSÃO E TRABALHOS FUTUROS

Este artigo apresentou uma avaliação inicial de ambientes virtualizados de alto desempenho que utilizam sistemas de banco de dados através de instâncias de banco de dados virtualizadas.

Instâncias em ambientes virtualizados são tratadas como *containers* que provém um ambiente protegido para a execução da aplicação. Portanto, o isolamento pode apresentar fatores determinantes sobre a escolha deste tipo de ambientes de produção para aplicações reais, pois podem determinar o impacto que aplicações de uma instância podem causar no desempenho e segurança de outras aplicações que executam em outras instâncias.

Além disso, outro fator determinante para a utilização destes ambientes é a escalabilidade. Esse tipo de avaliação pode mostrar que em um ambiente onde os recursos são disponibilizados dependendo da demanda da aplicação como, por exemplo, ambientes de HPC, estas aplicações terão o suporte necessário de suas camadas de virtualização para aproveitar os recursos disponíveis, aumentando o seu desempenho.

Neste contexto, o presente trabalho tem como objetivo identificar a melhor forma de executar um sistema de banco de dados em um ambiente virtualizado. Para isso,

foram realizadas avaliações de desempenho em ambientes que utilizam banco de dados sobre máquinas virtuais (BDVM) e banco de dados virtuais (BDV). Como parâmetro, também foram executadas avaliações de desempenho em ambientes de banco de dados sobre um sistema operacional nativo (BDSO). Outra análise realizada é a utilização de recursos, como CPU, memória, rede e consumo de energia, de cada abordagem estudada.

Atualmente, a abordagem mais comum para ambientes virtualizados é o suporte à banco de dados executados sobre MV. Porém, o presente trabalho mostra que a execução do sistema de banco de dados virtual (BDV) é tão eficiente quanto à execução do SGBD diretamente no sistema operacional com as vantagens da virtualização.

Analisando as figuras da Seção III-C1, é possível determinar que o BDSO e BDV tem um desempenho equivalente sendo que este último ainda proporciona os benefícios que a virtualização oferece, tais como consolidação, escalabilidade e alta disponibilidade. O BDVM tem um desempenho inferior entre 8 a 10%, observado na Figura 3 devido a inclusão da camada adicional de virtualização. A camada adicional de virtualização proporciona uma administração do SGBD flexível, porém o aumento do consumo dos recursos é nítido, conforme observado anteriormente nas Figuras 4 e 5.

Portanto, este trabalho mostra que os serviços de banco de dados usando a virtualização do banco de dados (BDV) podem aproveitar todas as características inerentes aos ambientes virtualizados sem perder o desempenho e ainda manter os benefícios oferecidos pela virtualização.

Como continuação deste trabalho é necessário a análise de situações variadas em relação à quantidade de transações por segundo e o número de usuários simultâneos e o estudo de ferramentas de monitoração de consumo de recursos para o autogerenciamento das instâncias virtuais. Neste trabalho utilizamos apenas uma instância de virtualização, pois o foco principal era avaliar o *overhead* dos diferentes tipos de camadas de virtualização estudados.

Também podemos estender o ambiente de teste com a utilização de várias instâncias. Ao utilizarmos várias instâncias de modo concomitante e concorrente, iremos testar outras características desses ambientes virtualizados, tais como isolamento e escalabilidade.

#### REFERÊNCIAS

- [1] B. Bacci, M. Danelutto, S. Pelagatti, and M. Vanneschi, "Skie: A heterogeneous environment for hpc applications," *Parallel Computing*, vol. 25, no. 13 – 14, pp. 1827 – 1852, 1999.
- [2] H. Raj and K. Schwan, "High performance and scalable i/o virtualization via self-virtualized devices," in *Proceedings of the 16th international symposium on High performance distributed computing*, ser. HPDC '07. New York, NY, USA: ACM, 2007, pp. 179–188.
- [3] A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel, "Diagnosing performance overheads in the xen virtual machine environment," in *Proceedings of the 1st ACM/USENIX international conference on Virtual*

- execution environments*, ser. VEE '05. New York, NY, USA: ACM, 2005, pp. 13–23.
- [4] V. Nae, R. Prodan, T. Fahringer, and A. Iosup, “The impact of virtualization on the performance of massively multi-player online games,” in *Proceedings of the 8th Annual Workshop on Network and Systems Support for Games*, ser. NetGames '09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 9:1–9:6.
- [5] A. Abounaga, C. Amza, and K. Salem, “Virtualization and databases: state of the art and research challenges,” in *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, ser. EDBT '08. New York, NY, USA: ACM, 2008, pp. 746–747.
- [6] R. Elmasri, S. Navathe, M. Pinheiro, C. Canhette, G. Melo, C. Amadeu, and R. de Oliveira Morais, *Sistemas de banco de dados*. Pearson Addison Wesley, 2005.
- [7] U. Minhas, J. Yadav, A. Abounaga, and K. Salem, “Database systems on virtual machines: How much do you lose?” in *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, april 2008, pp. 35–41.
- [8] R. N. Calheiros, G. Rodrigues, T. Ferreto, and C. A. F. De Rose, “Avaliando o ambiente de virtualização xen utilizando aplicações de bancos de dados,” in *VIII Workshop em Sistemas Computacionais de Alto Desempenho (WSCAD'07)*. Gramado, Brazil: SBC, October 2007, pp. 171–178.
- [9] A. A. Soror, U. F. Minhas, A. Abounaga, K. Salem, P. Kokosielis, and S. Kamath, “Automatic virtual machine configuration for database workloads,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 953–966.
- [10] “LXC Linux Containers,” 2012, [Online; accessed 19-July-2012]. [Online]. Available: <http://lxc.sourceforge.net>
- [11] “FreeBSD Jails,” 2012, [Online; accessed 19-July-2012]. [Online]. Available: <http://www.freebsd.org>
- [12] “Solaris Containers,” 2012, [Online; accessed 19-July-2012]. [Online]. Available: <http://www.oracle.com/technetwork/server-storage/solaris/containers-169727.html>
- [13] N. Regola and J.-C. Ducom, “Recommendations for virtualization technologies in high performance computing,” in *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, ser. CLOUDCOM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 409–416.
- [14] Y. Wada, Y. Watanabe, K. Syoubu, J. Sawamoto, and T. Katoh, “Virtual database technology for distributed database,” *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 0, pp. 214–219, 2010.
- [15] “Tpc consortium,” 2012, [Online; accessed 19-July-2012]. [Online]. Available: <http://www.tpc.org/tpcc/detail.asp>
- [16] “Hammerora,” 2012, [Online; accessed 19-July-2012]. [Online]. Available: <http://hammerora.sourceforge.net>
- [17] P. Shivam, A. Demberel, P. Gunda, D. Irwin, L. Grit, A. Yumerefendi, S. Babu, and J. Chase, “Automated and on-demand provisioning of virtual machines for database applications,” in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '07. New York, NY, USA: ACM, 2007, pp. 1079–1081.
- [18] C. Curino, E. P. Jones, S. Madden, and H. Balakrishnan, “Workload-aware database monitoring and consolidation,” in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, ser. SIGMOD '11. New York, NY, USA: ACM, 2011, pp. 313–324.
- [19] P. Apparao, S. Makineni, and D. Newell, “Characterization of network processing overheads in xen,” in *Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing*, ser. VTDC '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2–.
- [20] J. R. Lange, K. Pedretti, P. Dinda, P. G. Bridges, C. Bae, P. Soltero, and A. Merritt, “Minimal-overhead virtualization of a large scale supercomputer,” in *Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, ser. VEE '11. New York, NY, USA: ACM, 2011, pp. 169–180.
- [21] W. Huang, J. Liu, B. Abali, and D. K. Panda, “A case for high performance computing with virtual machines,” in *Proceedings of the 20th annual international conference on Supercomputing*, ser. ICS '06. New York, NY, USA: ACM, 2006, pp. 125–134.