

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**AVALIAÇÃO DOS IMPACTOS
DE UM NOVO PARADIGMA DE
VIRTUALIZAÇÃO DE BANCO
DE DADOS**

TIMOTEO ALBERTO PETERS LANGE

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Cesar Augusto FonticIELha De Rose

**Porto Alegre
2013**

Dados Internacionais de Catalogação na Publicação (CIP)

L274a Lange, Timoteo Alberto Peters
Avaliação dos impactos de um novo paradigma de virtualização de banco de dados / Timoteo Alberto Peters Lange. – Porto Alegre, 2013. 85 p.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Cesar Augusto FonticIELha De Rose.

1. Informática. 2. Banco de Dados. I. Marcon, Cesar Augusto FonticIELha De Rose. III. Título.

CDD 005.74

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "Avaliação dos Impactos de um Novo Paradigma de Aplicação de Banco de Dados" apresentada por Timoteo Alberto Peters Lange como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Inteligência Computacional, aprovada em 30/08/2013 pela Comissão Examinadora:

Prof. Dr. César Augusto FonticIELha De Rose –
Orientador

PPGCC/PUCRS

Prof. Dr. Duncan Dubugras Alcoba Ruiz –

PPGCC/PUCRS

Profa. Dra. Andrea Schwertner Charão –

UFSM

Homologada em 19/12/2013, conforme Ata No. 024 pela Comissão Coordenadora.

Prof. Dr. Luiz Gustavo Leão Fernandes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 – P32- sala 507 – CEP: 90619-900

Fone: (51) 3320-3611 – Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos

DEDICATÓRIA

Dedico este trabalho à minha amada esposa.

“O passado é história, o futuro é mistério, o presente é uma dádiva e por isso se chama presente.”
(Deepak Chopra)

AGRADECIMENTOS

Agradeço a DELL computadores, patrocinadora deste trabalho. Ao LAD - Laboratório de Alto Desempenho, por ter oportunizado este trabalho e ao professor Cesar De Rose, orientador e professor. A minha amada esposa, agradeço especialmente por estar ao meu lado, apoiando nos momentos tensos e participando das conquistas.

AVALIAÇÃO DOS IMPACTOS DE UM NOVO PARADIGMA DE VIRTUALIZAÇÃO DE BANCO DE DADOS

RESUMO

Estudos recentes demonstraram vantagens na utilização de Sistemas Gerenciadores de Banco de Dados (SGBD) em ambientes virtuais, como a consolidação de vários SGBDs isolados por máquinas virtuais em uma única máquina física para reduzir os custos de manutenção e consumo de energia. Além disso, a migração em tempo real pode melhorar a disponibilidade do banco de dados, permitindo que as operações de manutenção sejam transparentes para os usuários. No entanto, há questões que ainda precisam ser resolvidas, como a degradação do desempenho do SGBD quando executado em ambientes virtuais e a instabilidade das conexões durante a etapa do migração do serviço. Neste contexto, novas técnicas de virtualização que estão surgindo, como o banco de dados virtual, podem ser consideradas uma alternativa menos intrusiva para a virtualização tradicional de SGBD sobre máquinas virtuais. Esta pesquisa analisa os aspectos desta nova abordagem de virtualização, como o desempenho, estabilidade de conexão durante um processo de migração de banco de dados e a capacidade de isolamento. Este trabalho mostra resultados muito promissores em comparação com a abordagem tradicional sobre máquinas virtuais, incluindo uma migração em tempo real mais eficiente e estável, mantendo as características de isolamento necessários para um SGBD virtualizado.

Palavras Chave: Virtualização de banco de dados, Banco de dados virtual, Avaliação de desempenho.

EVALUATION OF THE IMPACTS OF A NEW PARADIGM OF DATABASE VIRTUALIZATION

ABSTRACT

Recent studies have demonstrated advantages in using Data Base Management System (DBMS) in virtual environments, like the consolidation of several DBMS isolated by virtual machines on a single physical machine to reduce maintenance costs and energy consumption. Furthermore, live migration can improve database availability, allowing transparent maintenance operations on host machines. However, there are issues that still need to be addressed, like overall performance degradation of the DBMS when running in virtual environments and connections instabilities during a live migration. In this context, new virtualization techniques are emerging, like the virtual database, which is considered a less intrusive alternative for the traditional database virtualization over virtual machines. This research analyzes aspects of this new virtualization approach, like performance and connection stability during a database migration process and its isolation capabilities. This evaluation shows very promising results compared to the traditional approach over virtual machines, including a more efficient and stable live migration, maintaining the required isolation characteristics for a virtualized DBMS.

Keywords: Database Virtualization, Virtual Database, Performance Evaluation.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1.1 – Processo de pesquisa. | 29 |
| Figura 2.1 – Estrutura de emulação de <i>hardware</i> | 33 |
| Figura 2.2 – Estrutura de virtualização total. | 34 |
| Figura 2.3 – Estrutura de paravirtualização. | 35 |
| Figura 2.4 – Estrutura de virtualização do sistema operacional. | 35 |
| Figura 3.1 – Estrutura de um BD - adaptado de [Dat04] | 41 |
| Figura 3.2 – BD em ambiente virtualizado usando Cluster. | 45 |
| Figura 3.3 – BD em ambiente virtualizado usando MV. | 46 |
| Figura 4.1 – Comparação entre as camadas de BD. | 52 |
| Figura 4.2 – Processo de migração de um BD sobre MV. | 54 |
| Figura 4.3 – Processo de migração de um BD virtual. | 55 |
| Figura 5.1 – Arquitetura do ambiente de avaliação (BDV e BDMV). | 60 |
| Figura 5.2 – Comparação de desempenho: transações executadas por minuto BDSO, BDMV e BDV | 64 |
| Figura 5.3 – Comparação de utilização de CPU: BDSO, BDMV e BDV | 64 |
| Figura 5.4 – Comparação do consumo de energia: BDSO, BDMV e BDV | 65 |
| Figura 5.5 – Comparação de operações de entrada e saída (rede e disco): BDSO, BDMV e BDV | 66 |
| Figura 5.6 – Comparação de operações de memória (Page-In, Page-Out e Page-Fault): BDSO, BDMV e BDV | 67 |
| Figura 5.7 – Utilização de CPU: Alteração de 16 CPUs para 8 CPUs. | 68 |
| Figura 5.8 – Desempenho TPM: alteração de 4 CPUs para 16 CPUs. | 69 |
| Figura 5.9 – Comparação das transações executadas por minuto durante a migração: BDV e BDMV. | 70 |
| Figura 5.10 – Comparação da utilização de CPU durante a migração: BDV e BDMV. | 71 |
| Figura 5.11 – Comparação da utilização de CPU durante a migração: BDV e BDMV. | 72 |
| Figura 5.12 – Comparação da utilização de memória durante a migração: BDV e BDMV. | 72 |
| Figura 5.13 – ER do BD utilizado no estudo de caso. | 75 |
| Figura 5.14 – Estudo de caso: comparação do desempenho do ambiente atual (BDMV) e BDV | 76 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 3.1 – Comparação entre os principais métodos de Alta Disponibilidade. | 47 |
| Tabela 4.1 – Características das implementação de BD sobre MV e BD virtual. | 56 |
| Tabela 5.1 – Transações TPC-C | 61 |
| Tabela 5.2 – Utilização de E/S: Valores de Leitura e Escrita com diminuição de CPU | 68 |
| Tabela 5.3 – Resultado de TPM com Aumento de Recursos à Instância 04. | 69 |
| Tabela 5.4 – Comparação de Desempenho Durante a Migração: BDVM e BDV. | 70 |
| Tabela 5.5 – Comparação do processo de migração: BDMV e BDV. | 73 |
| Tabela 5.6 – Matriz de Comparação | 74 |

LISTA DE SIGLAS

ER – Entidade Relacionamento
BD – Banco de Dados
BDMV – Banco de Dados sobre Máquina Virtual
BDSO – Banco de dados sobre Sistema Operacional
BDV – Banco de Dados Virtual
CPU – Central Processing Unit
LANS – Local Area Network
MV – Máquina Virtual
MC – Matriz de Comparação
OLTP – On-Line Transaction Processing
SGBD – Sistema Gerenciador de Banco de dados
SO – Sistema Operacional
TPC – Transaction Processing Performance Council
TPM – Transaction per Minute
TI – Tecnologia de Informação
VLAN – Virtual Area Network

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 25 |
| 1.1 | CONTEXTUALIZAÇÃO | 26 |
| 1.2 | OBJETIVO DO TRABALHO | 27 |
| 1.2.1 | OBJETIVOS ESPECÍFICOS | 27 |
| 1.2.2 | QUESTÕES DE PESQUISA | 27 |
| 1.2.3 | LIMITAÇÕES DA PESQUISA | 28 |
| 1.2.4 | ETAPAS DA PESQUISA | 28 |
| 1.3 | ORGANIZAÇÃO DO TEXTO | 29 |
| 2 | VIRTUALIZAÇÃO | 31 |
| 2.1 | DEFINIÇÃO DE VIRTUALIZAÇÃO | 31 |
| 2.2 | MOTIVAÇÕES PARA VIRTUALIZAR | 32 |
| 2.3 | TIPOS DE VIRTUALIZAÇÃO | 33 |
| 2.3.1 | EMULAÇÃO DE HARDWARE | 33 |
| 2.3.2 | VIRTUALIZAÇÃO TOTAL | 34 |
| 2.3.3 | PARA-VIRTUALIZAÇÃO | 34 |
| 2.3.4 | VIRTUALIZAÇÃO DO SISTEMA OPERACIONAL | 35 |
| 2.4 | RECURSOS QUE PODEM SER VIRTUALIZADOS | 36 |
| 2.5 | CONCLUSÕES DESTE CAPÍTULO | 37 |
| 3 | BANCO DE DADOS | 39 |
| 3.1 | DEFINIÇÃO DE BANCO DE DADOS | 39 |
| 3.2 | ESTRUTURA DE BANCO DE DADOS | 40 |
| 3.3 | IMPLEMENTAÇÃO DE BANCO DE DADOS EM AMBIENTES VIRTUALIZADOS | 42 |
| 3.3.1 | DESEMPENHO EM AMBIENTES VIRTUALIZADOS | 43 |
| 3.3.2 | GERÊNCIA DE RECURSOS EM AMBIENTES VIRTUALIZADOS | 44 |
| 3.3.3 | ALTA DISPONIBILIDADE EM AMBIENTES VIRTUALIZADOS | 45 |
| 3.4 | CONCLUSÕES DESTE CAPÍTULO | 47 |
| 4 | NOVO PARADIGMA PARA VIRTUALIZAÇÃO DE BANCO DE DADOS | 49 |
| 4.1 | DESAFIOS DA VIRTUALIZAÇÃO DE BANCO DE DADOS | 49 |
| 4.2 | MODELO DO BANCO DE DADOS VIRTUAL | 50 |
| 4.2.1 | GERÊNCIA DE RECURSOS DE BD VIRTUAL | 52 |

| | | |
|----------|---|-----------|
| 4.2.2 | ALTA DISPONIBILIDADE USANDO BD VIRTUAL | 55 |
| 4.3 | COMPARAÇÃO ENTRE AS ABORDAGENS DE UTILIZAÇÃO DE BD EM AMBI- ENTES VIRTUAIS | 55 |
| 4.4 | CONCLUSÕES DESTE CAPÍTULO | 57 |
| 5 | AVALIAÇÃO DE BANCO DE DADOS VIRTUAL | 59 |
| 5.1 | METODOLOGIA DE AVALIAÇÃO | 59 |
| 5.1.1 | AMBIENTE DE EXECUÇÃO | 60 |
| 5.1.2 | BENCHMARK E DESCRIÇÃO DOS TESTES | 61 |
| 5.2 | RESULTADOS OBTIDOS | 63 |
| 5.2.1 | AVALIAÇÃO DE DESEMPENHO | 63 |
| 5.2.2 | AVALIAÇÃO DE GERÊNCIA DE RECURSOS | 67 |
| 5.2.3 | AVALIAÇÃO DE ALTA DISPONIBILIDADE | 69 |
| 5.3 | MATRIZ DE COMPARAÇÃO | 73 |
| 5.4 | ESTUDO DE CASO | 74 |
| 5.5 | CONCLUSÕES DESTE CAPÍTULO | 77 |
| 6 | CONCLUSÃO | 79 |
| 6.1 | RESUMO DAS CONTRIBUIÇÕES | 80 |
| 6.2 | TRABALHOS FUTUROS | 81 |
| | REFERÊNCIAS | 83 |

1. INTRODUÇÃO

Atualmente, a virtualização de recursos computacionais vem sendo utilizada na maioria dos recursos de Tecnologia de Informação (TI), tais como servidores, redes, sistemas operacionais e aplicações. Entre eles, a virtualização de servidores é a mais popular e implementa uma camada lógica sobre um sistema físico, de forma que todos os dispositivos possam ser acessados ou compartilhados por aplicações clientes suportadas neste ambiente [AM10].

Dentre as principais vantagens da utilização de tecnologias de virtualização, podemos citar o fornecimento de forte isolamento, segurança e consolidação de servidores físicos [BDF⁺03]. Além disso, também podem ser exploradas novas possibilidades, tais como a alta disponibilidade e balanceamento de carga de trabalho [AM10], [MST⁺05].

Neste contexto, Sistemas Gerenciadores de Banco de Dados (SGBDs) são suportados normalmente em um ambiente físico, implementado diretamente no *hardware* [AZR⁺12]. A principal característica deste tipo de implementação é o desempenho nativo, por não apresentar a camada intermediária de virtualização, porém, benefícios como de consolidação e gerência de recursos não se aplicam neste tipo de implementação.

Nesse sentido, para suprir necessidades como gerenciamento de recursos e alta disponibilidade, existe uma tendência em tentar explorar os benefícios da virtualização para vários tipos de aplicações, inclusive para SGBDs [MYAS08]. De fato, pesquisas recentemente avaliam o impacto da utilização de SGBDs em ambientes virtualizados e propõem soluções específicas para esse tipo de ambiente [SMA⁺08], [Tya09], [AAS08].

Geralmente, ambientes virtualizados suportam SGBDs executados sobre máquinas virtuais (VM) [CRFD07]. A principal vantagem de executar um sistema de banco de dados em um ambiente virtualizado pode ser considerada a consolidação de vários servidores virtuais em um único servidor físico, reduzindo assim o custo de manutenção e consumo de energia [CJMB11]. A utilização desta abordagem adiciona uma camada intermediária responsável pela virtualização, provocando uma queda de desempenho [MYAS08], porém adicionando os recursos inerentes à virtualização, tais como consolidação e gerenciamento de recursos [HLAP06].

Normalmente, o servidor de banco de dados possui recursos limitados de *hardware* devido à demanda deste tipo de aplicação. Além desta característica, existem outras propriedades interessantes que devem ser levadas em consideração e que justificam a discussão sobre virtualização de servidores para banco de dados [SDG⁺07]. Tais características incentivam como uma boa prática motivada pelo desempenho a utilização de sistemas de banco de dados em ambientes físicos, aproveitando ao máximo o desempenho oferecido diretamente pelo hardware.

Como alternativa às abordagens acima citadas, pode ser utilizada a virtualização da instância de banco de dados [AZR⁺12]. Essa abordagem utiliza a virtualização baseada em *containers*. Uma das principais vantagens na utilização desta abordagem é a possibilidade de utilização de recur-

sos inerentes à virtualização, sem a perda de desempenho ocasionada pela inclusão de uma camada intermediária [LCR⁺12].

Neste sentido, o desafio é compreender as características de cada abordagem e avaliar a melhor forma para empregar uma infraestrutura de banco de dados em ambiente virtualizado. Considerando os benefícios trazidos pela virtualização e procurando alternativas que minimizam a degradação de desempenho provocado pela inserção de uma camada adicional de *software*, este trabalho tem como objetivo avaliar e identificar a melhor opção de implementação de uma infraestrutura de banco de dados em um ambiente virtualizado, avaliando as abordagens conhecidas e analisando a melhor alternativa à virtualização de banco de dados em ambientes virtualizados.

1.1 Contextualização

Conforme abordado na Seção anterior, 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) mediando os recursos físicos disponíveis no hospedeiro [Car08].

A principal vantagem de executar um SGBD em um ambiente virtualizado é a capacidade de gerenciamento e a consolidação de vários servidores virtuais em um único servidor físico, reduzindo assim o custo de manutenção e consumo de energia [AAS08] [BDF⁺03]. Dentre outras vantagens como, redução de custos com licenças, facilidade de gerenciamento e alta disponibilidade, tornam a virtualização uma alternativa atraente para sistemas de banco de dados, implantado geralmente por um sistema de banco de dados sobre uma VM [Tya09].

A virtualização pode resolver vários problemas importantes na área de banco de dados, tais como a usabilidade, o gerenciamento, a escalabilidade, a disponibilidade e a implantação [Jon06]. Aplicações de banco de dados podem se favorecer desse tipo de infraestrutura através de algumas características como migração e consolidação [SDG⁺07]. Assim, dependendo da demanda da aplicação de banco de dados, o ambiente pode se ajustar dinamicamente provendo mais ou menos recursos à aplicação [SDG⁺07] [CJMB11].

Vários trabalhos analisam o *overhead* da camada de virtualização como [MST⁺05], [RS07] [AMN06] [LPD⁺11], que apresentam uma perda de desempenho entre 7 e 11%, se comparado com aplicações em um sistema operacional nativo. Alguns trabalhos como [AAS08], [CRFD07] [LCR⁺12] avaliam o *overhead* da camada de virtualização com aplicações de banco de dados. Pode ser visto também que em algumas situações o ganho com a gerência de recursos minimiza a degradação provocada pelo *overhead* da camada de virtualização, o que justifica a sua utilização em ambientes virtualizados [LPD⁺11], [HLAP06].

Neste sentido, este trabalho 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 a gerência de recursos e isolamento para aplicações de SGBD baseada em *containers*, aplicada para banco de dados [AZR⁺12].

1.2 Objetivo do Trabalho

O objetivo geral deste trabalho é estudar e avaliar o desempenho de um sistema de banco de dados instalado diretamente sobre o *hardware*, sobre MV e utilizando o conceito de banco de dados virtual, realizando uma comparação entre as abordagens e identificando as características de implementação. Além disso, tem como objetivo de prover um estudo de caso de um ambiente real, comparando os benefícios da nova abordagem.

1.2.1 Objetivos Específicos

Motivado pelo objetivo geral deste trabalho, direciona-se a proposta para um contexto mais centrado, através dos seguintes objetivos específicos:

- Como principal contribuição, tem por objetivo específico explorar o conceito de virtualização de banco de dados utilizando a tecnologia de *containers*.
- Testar, avaliar e comparar particularidades de funcionamento tais como valores de desempenho, consumo de recursos e características de gerência das abordagens mais comuns, comparando com abordagem explorada neste trabalho.
- Desenvolver uma matriz de comparação para identificar o modelo adequado a ser aplicado, variando as necessidades para cada implementação.
- Realizar um estudo de caso de um ambiente real, aplicando a matriz de comparação e observando o desempenho de ambiente reais.

1.2.2 Questões de Pesquisa

O presente trabalho busca resolver as seguintes questões de pesquisa:

- Existe uma alternativa à área de banco de dados para diminuir o *overhead* causado pela inclusão da camada adicional de virtualização, porém mantendo os benefícios trazidos pela mesma?
- Qual é a eficiência da abordagem banco de dados virtual?
- Qual é o custo do processo de migração utilizada na consolidação, das abordagens tratadas neste trabalho?

- É possível executar múltiplas instâncias de banco de dados virtuais em um mesmo servidor físico sem que uma prejudique o desempenho da outra, garantindo o isolamento durante a execução?

1.2.3 Limitações da Pesquisa

A presente pesquisa, apesar do rigor nos procedimentos empregados apresenta algumas limitações. Na fundamentação teórico-empírica, por ser um tema ainda pouco abordado, a escassez de literatura foi um fator limitador.

Os resultados obtidos só podem ser avaliados em ambientes proprietários com o uso da licença de *software*.

Os valores obtidos referentes ao desempenho de transações executadas nos ambientes de banco de dados referencia um tipo de carga de trabalho específica.

1.2.4 Etapas da Pesquisa

O processo de pesquisa deste trabalho foi desenvolvido em quatro etapas: Estudo, Planejamento, Construção e Validação. Estas etapas podem ser observadas na Figura 1.1 e visam nortear os objetivos a serem alcançados durante o processo de desenvolvimento da pesquisa, objetivando os resultados de cada etapa. As etapas executadas durante o processo de pesquisa deste trabalho são descritas a seguir:

- **Estudo:** a primeira fase deste trabalho foi identificar e selecionar os sistemas de bancos de dados que apresentassem as características necessárias para o desenvolvimento da pesquisa. Esta atividade foi conduzida através de pesquisa do estado da arte, identificando as abordagens existentes.
- **Planejamento:** além de um levantamento detalhado sobre os SGBDs, também foi necessário um entendimento sobre virtualização e consolidação de sistemas, identificando as melhores práticas descritas no estado da arte. Foi também necessário um entendimento sobre *benchmarks*, onde foi definido o sistema de medição de desempenho mais adequado para a realização deste trabalho.
- **Construção:** Após o estudo da arte envolvendo banco de dados e virtualização, o próximo passo foi implementar os sistemas de bancos de dados nos modelos abordados anteriormente: utilização em ambiente nativo de *hardware*, utilização em uma máquina virtual e banco de dados virtual. É nesta etapa que os dados foram gerados e coletados.
- **Validação:** Os dados de desempenho gerados pelo *benchmark* e os dados de utilização de recursos foram analisados pela matriz de comparação, agrupada pelo modelo de implementa-

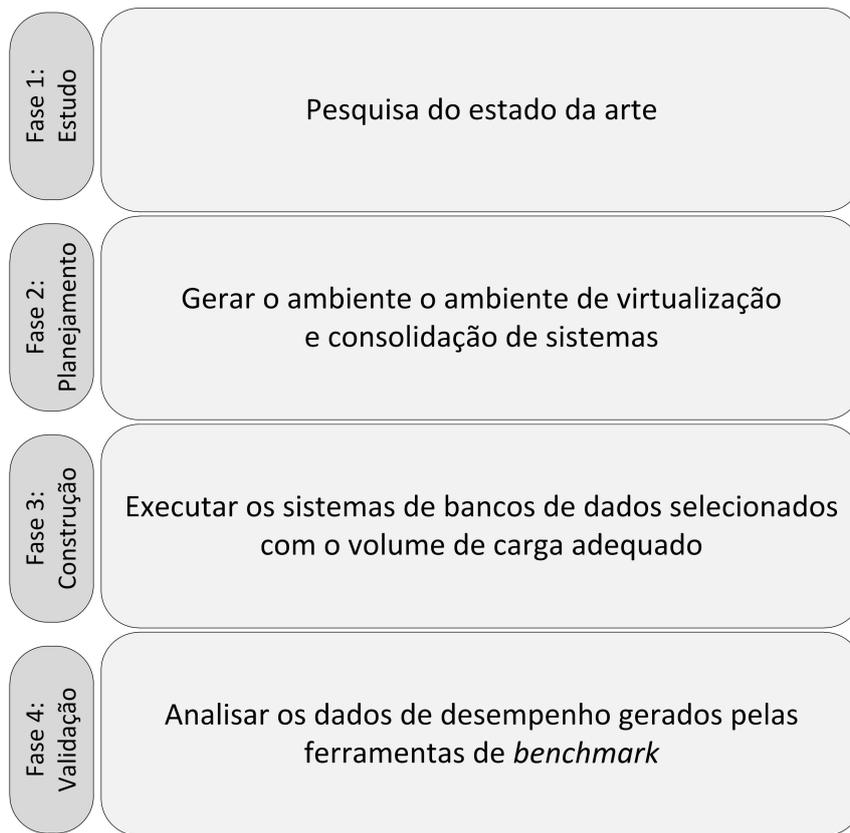


Figura 1.1 – Processo de pesquisa.

ção. Nesta etapa final, foi identificado o melhor modelo de implementação de um sistema de banco de dados de acordo com as características necessárias à execução e aplicado no estudo de caso.

1.3 Organização do Texto

O texto está organizado da seguinte maneira: primeiramente, no Capítulo 2 é apresentada uma contextualização sobre os conceitos de virtualização; No Capítulo 3 são expostos os princípios de banco de dados. Esse estudo é necessário para a correta compreensão deste trabalho e sua relação com os ambientes virtualizados; O Capítulo 4 trata do novo paradigma de virtualização de sistemas de banco de dados, mostrando uma alternativa à utilização convencional de virtualização, focando em uma das funcionalidades de gerenciamento e consolidação, qual seja, o processo de migração em um ambiente virtualizado; No Capítulo 5 são detalhados: o ambiente de avaliação, o *benchmark* e a descrição dos testes. Também neste Capítulo são mostrados os resultados dos testes e o resultado do estudo de caso em um ambiente real; Por fim, no Capítulo 6 é apresentado o resumo das contribuições, conclusões alcançadas e uma possível continuação desta pesquisa.

2. VIRTUALIZAÇÃO

A virtualização de recursos adiciona uma camada intermediária de software entre as aplicações e o *hardware*. Esta camada mapeia os recursos virtuais visíveis às aplicações mediando com os recursos físicos disponíveis no *hardware* hospedeiro [SMA⁺08]. A camada de comunicação intermediária entre os recursos físicos e os recursos virtuais possibilita o compartilhamento dos recursos e a realocação destes recursos quando necessário. Assim, em um servidor físico podem ser executadas múltiplas aplicações (isoladas uma da outra) compartilhando os mesmos recursos de *hardware* [SMA⁺08].

Este capítulo está organizado da seguinte forma: primeiramente na Seção 2.1 são apresentados alguns conceitos de virtualização; Na Seção 2.2 são descritas as principais motivações que justificam o processo de virtualização; a seguir, na Seção 2.3 são abordados os principais tipos de virtualização; e por último, na Seção 2.4 são analisados alguns exemplos de recursos que podem ser virtualizados.

2.1 Definição de virtualização

A virtualização é amplamente definida como o ato de executar um *software* em um ambiente virtual [Car08]. Um ambiente virtual existe quando um Sistema Operacional (SO) é de alguma maneira emulado ou não é executado diretamente no *hardware* físico. No contexto deste trabalho, vamos adotar essa definição, já que apresenta a virtualização como a possibilidade de executar um *software* ou serviço em um ambiente virtual, sem que os mesmo tenham o conhecimento de que estão sendo executados neste ambiente.

Me outra definição, virtualização é o processo de desassociação do *hardware* conhecido pelo SO e a máquina física propriamente dita, comumente considerada uma camada *hardware* criada em *software*. Simplificando, a virtualização é semelhante a um computador dentro de um outro computador, implementado em *software*, gerando a ilusão da existência de dispositivos físicos, como placas de som, CPUs, memória e armazenamento físico [CJ06].

Um SO sendo executado em um ambiente virtualizado é conhecido como uma Máquina Virtual (MV). A tecnologia de virtualização permite que várias MVs, com sistemas operacionais diferentes, executem lado a lado, isolados e na mesma máquina física, emulando um sistema completo de *hardware*, desde o microprocessador até placa de rede.

Cada MV pode compartilhar um conjunto comum de recursos de *hardware*, isoladas de tal forma que uma MV não influencia nem depende de outra. Este tipo de isolamento oferece a utilização do compartilhamento de *hardware* de maneira que durante a execução da MV não é percebido que o *hardware* também pode estar sendo usado por outra MV ao mesmo tempo.[BDF⁺03]

2.2 Motivações para virtualizar

De acordo com o apresentado na Seção anterior, a virtualização é implementada através de uma camada intermediária de *software*, que é responsável pela administração e compartilhamento dos recursos físicos do *hardware* hospedeiro. Esta camada pode implicar diretamente no desempenho. Já há algum tempo os fabricantes de *hardware* têm incluído instruções específicas que auxiliam no processo de virtualização, assessorando esta camada de *software* a resolver certas dificuldades conhecidas, como por exemplo, a tradução de endereços de memória e o desempenho. Neste contexto, também podem ser citadas as dificuldades de segurança, gerenciamento e dimensionamento da MV [Car08].

Conhecendo algumas das principais dificuldades do processo de virtualização, torna-se necessário estudar e identificar a melhor forma de implementar a virtualização, minimizando ou eliminando alguns incômodos ocasionados, motivados pelos benefícios que essa tecnologia pode ocasionar para os ambientes de TI. Dentre as principais vantagens da utilização de virtualização, podemos citar:

- **Consolidação da infra-estrutura:** são necessários menos equipamentos físicos para disponibilizar a mesma quantidade de sistemas. Esta consolidação representa redução do espaço necessário para a acomodação dos equipamentos físicos gerando redução de custos com energia (tanto para funcionamento como para resfriamento e redução de custos para aquisição e manutenção dos equipamentos). A consolidação também possibilita economizar com licenças e reduzir a ociosidade dos recursos computacionais [BDF⁺03].
- **Maior eficiência na utilização dos recursos:** uma prática utilizada é virtualizar um ambiente que tem um perfil de alta utilização, de modo que os recursos estejam sempre sendo utilizados e permaneçam a menor parte do tempo ociosos [BDF⁺03].
- **Aumento da disponibilidade:** dada a facilidade de migrar sistemas virtuais entre servidores físicos é possível manter um ambiente replicado para o caso de falhas ou paradas programadas de manutenção [BDF⁺03], [SKM08].
- **Melhoria no gerenciamento:** considerando que os recursos são virtuais, estes podem ser migrados, replicados ou redimensionados com maior facilidade. Entre as práticas comuns, uma delas é conceder mais memória, processamento, ou espaço em disco para um sistema quando for necessário sem que seja necessário alterar o *hardware* do servidor [SKM08].
- **Flexibilidade da infra-estrutura:** diversos sistemas operacionais heterogêneos podem ser executados no mesmo ambiente, além de executar ambientes legados em recursos novos de *hardware* [BDF⁺03].
- **Facilidade de execução de *backup*:** é possível fazer uma cópia completa do sistema (recurso conhecido como *snapshot*) em seu estado atual, incluindo toda sua configuração. Algumas

ferramentas ainda disponibilizam a execução do *backup* durante a execução, eliminando a necessidade de parada [BDF⁺03], [SKM08].

- **Redução do tempo de entrega de novos servidores:** novos servidores podem ser criados rapidamente sem a necessidade de espera de aquisição de um novo *hardware*. Outra melhora é pelo fato destes servidores serem criados a partir de *templates* já disponíveis no ambiente [BDF⁺03].

Motivado pelas vantagens apresentadas, a virtualização pode ser considerada uma estratégia apropriada para a área de gerência de infra-estrutura, pois conforme visto, tais benefícios provocam a migração dos recursos físicos para um ambiente virtualizado, agregando funcionalidades exclusivas à virtualização.

2.3 Tipos de virtualização

A virtualização pode ser usada em diferentes modelos e níveis de acesso ao *hardware* hospedeiro [Jon06]. Neste trabalho são abordados os principais modelos de virtualização, apresentados a seguir.

2.3.1 Emulação de hardware

A emulação de *hardware* é a utilização de uma arquitetura completamente diferente da utilizada no *hardware* hospedeiro. Esse tipo de virtualização pode ser utilizada por diversos motivos, como por exemplo, utilizar uma arquitetura atual com maior desempenho para emular uma arquitetura de *hardware* antiga que não seja mais comercializada, sem que seja necessário migrar os sistemas legados. Na Figura 2.1 temos a estrutura de camadas deste modelo de virtualização. É possível observar que existe uma camada extra (*Hardware MV A* e *Hardware MV B*) entre as camadas de SO Convidado e *Hardware*. Essa camada extra é a responsável por emular o *hardware* virtualizado independente do *hardware* hospedeiro [Car08].

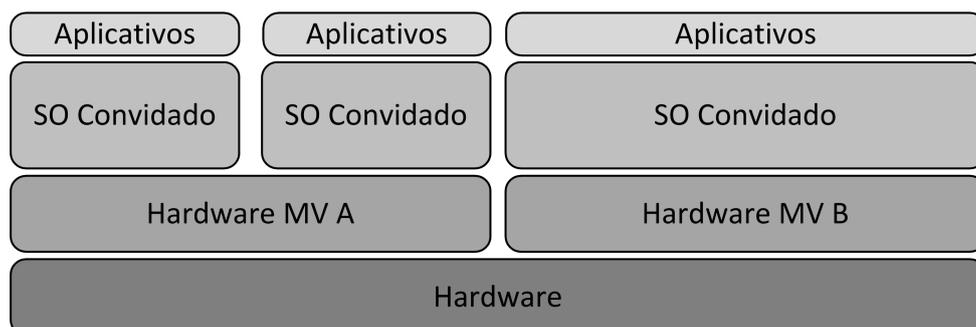


Figura 2.1 – Estrutura de emulação de *hardware*.

Analisando algumas das desvantagens, pode ser verificado que neste modelo cada instrução executada no SO convidado deve ser mapeada de uma arquitetura virtualizada para a arquitetura física, prejudicando assim o desempenho [Jon06]. Isto ocorre porque uma máquina emulada, recria uma arquitetura de pilha, registradores, filas, etc, onde ao executar uma instrução, essa deverá repassar a instrução de uma arquitetura emulada para ser executada na arquitetura nativa.

2.3.2 Virtualização total

Virtualização total utiliza uma camada de *software* para abstrair os recursos físicos, porém diferentemente da anterior, o objetivo é fornecer ao SO visitante uma réplica do *hardware* subjacente. Dessa forma o SO visitante é executado sem modificações sobre o MMV. As estrutura desse tipo de virtualização pode ser observada na Figura 2.2. Nesse tipo de virtualização, o *hardware* do sistema hospedeiro é emulado de forma que os SOs instalados (SO convidado) são executados sobre a camada de controle (Hypervisor MMV) e devem ser compatíveis com a arquitetura nativa [CJ06].

Esse tipo de virtualização é mais rápida que a emulação de *hardware*, porém o desempenho ainda é inferior do que a utilização nativa devido à mediação da camada intermediária de *software* (*Hypervisor*). A camada *hypervisor* é responsável pela segurança dos recursos de *hardware* e pelo controle do fluxo de informações entre MVs e o *hardware* hospedeiro [PPS⁺07].

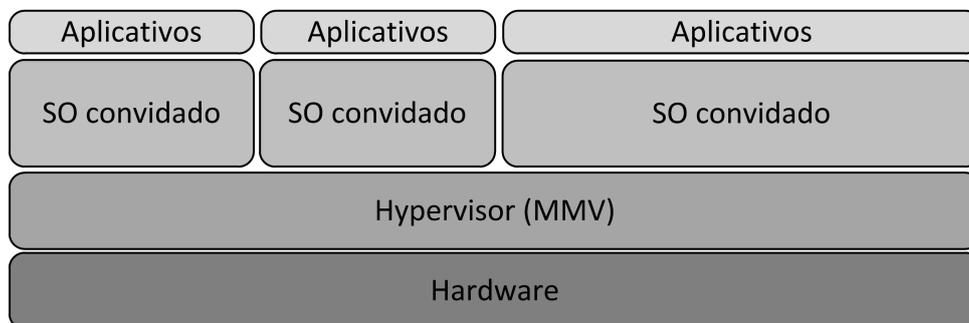


Figura 2.2 – Estrutura de virtualização total.

A melhoria deste tipo de virtualização comparado com a anterior, pode ser explicado considerando que o *hypervisor* simplesmente repassa as instruções a serem executadas para o *hardware* nativo, reduzindo assim o custo de processamento em relação a emulação de *hardware*.

2.3.3 Para-virtualização

A para-virtualização também utiliza uma camada de *software* de controle (*Hypervisor MMV*), porém nesta abordagem há alterações nos SOs convidados para que estes cooperem com o processo de virtualização [CJ06]. Neste modelo são permitidos somente SO que estejam aptos a estas alterações, o que limita a utilização, porém, há um ganho significativo de desempenho se

comparada aos modelos anteriores, considerando que os SOs convidados podem se beneficiar do fato de conhecerem que estão em um ambiente virtual [CJ06]. A estrutura desse tipo de virtualização pode ser observada na Figura 2.3. É possível notar que entre as camadas de SO convidado e *hypervisor* há uma camada de alteração do SO, responsável por possibilitar que as instruções sejam executadas diretamente no *hardware* hospedeiro.

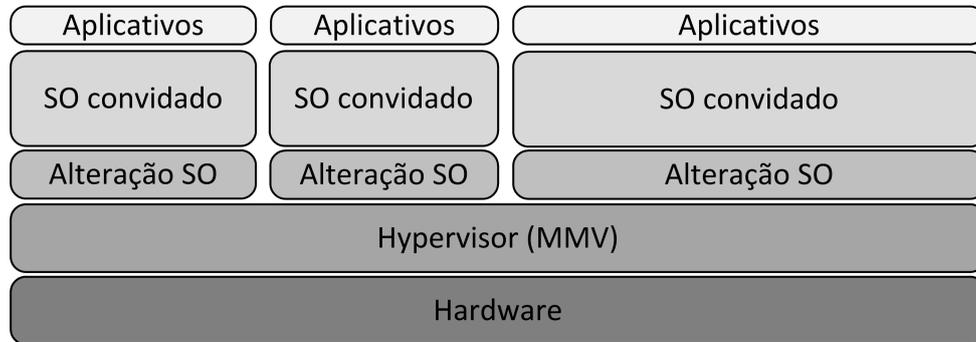


Figura 2.3 – Estrutura de paravirtualização.

Uma das características dessa abordagem é que o SO convidado é modificado para acessar o MMV. Isso pode ser considerado uma desvantagem, pois o SO deve ser adaptado ao *hypervisor* [Car08]. Porém, tal característica permite que as instruções não obriguem mais a serem traduzidas ou repassadas para o *hardware* hospedeiro, possibilitando a execução de algumas instruções diretamente, justificando assim o aumentando de desempenho.

2.3.4 Virtualização do sistema operacional

A virtualização do SO permite que um tipo de SO conviva com diversas outras instâncias do mesmo sistema, compartilhando os recursos físicos. A MV funciona como um processo do sistema anfitrião. A estrutura desse tipo de virtualização é apresentada na Figura 2.4. Neste método, o SO deve ter um suporte a este tipo de operação, sendo a principal vantagem o desempenho, que é comparado à utilização nativa [Jon06].

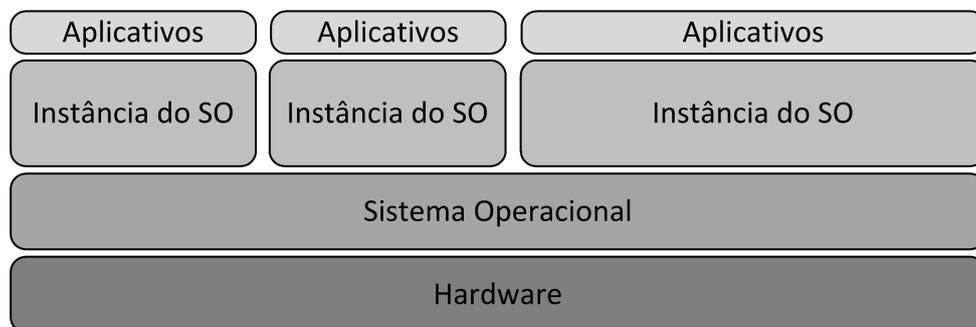


Figura 2.4 – Estrutura de virtualização do sistema operacional.

Essa utilização é normalmente implementada por *containers*. De forma simplificada, são criados servidores virtuais em uma subcamada do SO (*kernel*). Também pode ser descrito como

um particionamento de um servidor físico em múltiplas partições menores, e cada partição é isolada das demais e do SO anfitrião [XNR⁺13].

Devido à implementação, normalmente em nível de *kernel* do SO, pode ser considerada como uma desvantagem a necessidade de que o ambiente virtual não pode fazer uso de outros SOs, ficando restrito sempre a utilização do mesmo SO do ambiente hospedeiro.

Dentre os tipos de virtualização apresentados, podemos observar as principais características de cada uma, possibilitando assim uma avaliação imparcial. Um fator importante na decisão do tipo de virtualização a ser adotado é a interferência que a camada de virtualização gera na execução das instruções, sendo que pode ser observado que a última é a que menos interfere, possibilitando uma execução com desempenho comparado à execução nativa.

2.4 Recursos que podem ser virtualizados

A virtualização pode ser considerada atualmente como uma prática comum em ambientes de TI [BDF⁺03]. Atualmente vários recursos podem ser virtualizados. Dentre os principais recursos que podem ser virtualizados em um ambiente de TI, este trabalho pode citar:

- **Virtualização de servidores:** possibilita a criação de múltiplas MV isoladas uma das outras, com seus respectivos SO e em uma única máquina física, ainda que sejam logicamente distintas, porém com perfis de *hardware* consistente [AM10].

Esta modalidade muitas vezes pode tomar o lugar da prática onerosa de consolidação de servidores manual, através da combinação de vários servidores virtuais em um servidor físico. O foco da virtualização de servidores está em maximizar a eficiência do *hardware* do servidor, a fim de aumentar o retorno sobre o investimento do *hardware* [AM10], [CJ06].

- **Virtualização de aplicações:** cria uma instância do aplicativo de acordo com a necessidade e distribuída através de uma rede conforme a demanda. Esta estratégia de virtualização vai de mãos dadas com a iniciativa de padronizar os serviços de internet [Jon06].

Aplicativos virtualizados podem usar uma camada de abstração em comum que define um protocolo permitindo a comunicação de um aplicativo com outro em um formato de mensagem padrão. Assim, as aplicações podem chamar outro aplicativo, a fim de executar as funções solicitadas. A aplicação virtualizada não só é capaz de invocar remotamente os pedidos e retornar resultados, mas também garantir que o estado do aplicativo e outros dados estejam disponíveis e com recursos consistentes em todos os servidores que executam a aplicação.

- **Virtualização de storage:** possibilita que dispositivos de armazenamento separados possam ser combinados e sejam acessados como uma única unidade. Essa modalidade tenta maximizar a eficiência do *hardware* de armazenamento em uma arquitetura virtual [SKM08].

- **Virtualização de rede:** possibilita a criação de múltiplas redes virtuais que podem ser combinadas em uma única rede física (pode ser suportado em um único equipamento de rede) ou ainda, podem criar redes separadas logicamente em várias partes no mesmo ambiente. Uma prática comum é criar LANs (*Local Area Network*) virtuais, ou VLANs (*Virtual Area Network*), a fim de gerenciar de forma mais eficaz uma rede [SKM08].
- **Virtualização de banco de dados:** propicia que servidores de banco de dados possam ser tratados como serviços, possibilitando o acesso aos dados localizados em qualquer lugar geograficamente sem a necessidade de saber ou se importar onde os dados realmente residem. A virtualização de banco de dados permite o uso de múltiplas instâncias virtuais de banco de dados simultaneamente, de forma transparente e independentemente da sua localização física. Estas práticas podem ser empregadas em mineração de dados e sistemas de armazenamento de dados [AAS08], [CRFD07], [MYAS08], [Tya09].

No contexto deste trabalho, banco de dados é uma classe de sistemas que tem características específicas e que devem ser consideradas antes de virtualizar. No Capítulo 3 essas características serão abordadas de forma mais detalhada.

2.5 Conclusões deste capítulo

No decorrer deste Capítulo, foram introduzidos conceitos sobre virtualização. Inicialmente foram apresentadas algumas definições, tendo como referência comum de que a virtualização dissocia a execução de um sistema do *hardware*, criando uma camada de abstração responsável pela segurança e compartilhamento de recursos [Car08] [CJ06].

Porém, a desassociação do sistema com a camada de *hardware* por si só não justifica a virtualização do ambiente de TI. Ao considerar as vantagens como: melhora na gerência de recursos, economia, alta disponibilidade e consolidação, o processo de virtualização se torna interessante, mesmo com a ônus da perda de desempenho.

A perda de desempenho pode ser minimizada considerando as características de cada sistema de virtualização. Alguns tipos de virtualização apresentam vantagens sobre outras, como por exemplo, facilidade de gerenciamento, transparência do *hypervisor* ou desempenho semelhante ao ambiente nativo. Dos tipos de virtualização apresentados, é possível considerar a virtualização de SO como um tipo em potencial para explorar as facilidades oferecidas e minimizando a perda de desempenho [XNR⁺13].

Desta forma, tendo como possibilidade e execução de ambientes virtualizados com desempenho quase nativo, a virtualização torna-se um atrativo para alguns recursos, dentre eles, podemos destacar a virtualização de banco de dados.

Os conceitos aqui apresentados tornan-se fundamentais para o prosseguimento deste trabalho, devido a relação da proposta com as tecnologias de virtualização.

3. BANCO DE DADOS

Banco de Dados (BD) 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. [EN11]. 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 BD [Dat04].

Logo, o BD é uma classe de recursos que tem características bem específicas e estas devem ser consideradas antes de virtualizar um servidor de BD.[MYAS08]. Este Capítulo aborda uma visão geral sobre BD e a sua utilização em ambientes virtualizados.

O texto está organizado da seguinte forma, primeiramente, Na Seção 3.1 é apresentada a definição e as características de um BD; Na seqüência, é explorado na Seção 3.2 a estrutura de BD; Na Seção 3.3 é abordado a implementação de BD em ambientes virtualizados, fazendo um estudo das principais dificuldades da utilização nesses ambientes e e por último, A Seção 3.4 traz as conclusões alcançadas com este Capítulo.

3.1 Definição de Banco de Dados

Banco de dados é um recurso que pode ser interpretado como um sistema computadorizado de armazenamento de registros, porém, não se resume apenas ao armazenamento dos registros, mas oferece também a possibilidade de manipulação e gerenciamento destes registros. A manipulação dos registros poderá acontecer sob a forma de alterações, inserções e exclusões de dados [Dat04].

A definição de BD, adotada neste trabalho, descreve como uma coleção de dados relacionados ou fatos que podem ser armazenados e que possuem um significado implícito [EN11]. Algumas das principais características de um BD são apresentadas a seguir:

- **Representa fatos do mundo real:** BD representa alguns aspectos do mundo real na estrutura dos dados. Alguns autores chamam essa estrutura de dados de mini mundo. Qualquer mudança no mini mundo pode se refletida em todo o BD [EN11].
- **Dados organizados:** BD é uma coleção lógica e coerente de dados com algum significado inerente. Uma organização de dados ao acaso (randômica) não pode ser corretamente interpretada como um BD [EN11].
- **Dados coerentes:** O BD é projetado, construído e populado por dados, atendendo a um assunto em comum. Possui um grupo de usuários definidos que acessam esses dados por meio de algumas aplicações previamente geradas, administrando os dados de acordo com o interesse desse grupo de usuários [EN11].

- **Tamanho flexível:** O BD pode ser de qualquer tamanho e de complexidade ao longo da sua existência [EN11]. Dessa forma, um conjunto de dados pode conter de centenas até milhões de registros, de variados formatos e complexidades. Normalmente os dados estão relacionados a um domínio (assunto comum), proporcionando que esses se relacionem entre si de diversas formas. Banco de dados corporativos consomem uma porção considerável dos recursos físicos do servidor, considerando o grande volume de operações executadas [CJMB11].

Outra maneira simplificada de identificar um BD é associar os dados propriamente com um conjunto de programas que acessam estes dados. Este conjunto de programas possibilita a correta manipulação, segurança e coerência dos dados e normalmente é identificado como Sistema Gerenciador de Banco de Dados (SGBD) [CJMB11]. Com isto, é possível concluir que um BD é um conjunto de arquivos físicos (*data files*) e os seus programas (SGBD) que possibilitam a correta manipulação e manutenção dos dados.

Os SGBDs devem ter a premissa de possibilitar o acesso aos dados garantindo a unicidade, integridade, independência e segurança. [Dat04] Outra função que atualmente é atribuída ao SGBD é a garantia de acesso a múltiplas conexões ao BD de forma simultânea garantido total controle e segurança [AZR⁺12].

O SGBD deve incluir um software de controle para garantir que em caso de múltiplos acessos simultâneos tentarem atualizar o mesmo dado, o faça de um modo controlado e ordenado, para assegurar que os resultados das atualizações sejam corretos. Uma regra fundamental do software do SGBD multiusuário é garantir que as transações concorrentes operem corretamente [AZR⁺12].

3.2 Estrutura de Banco de dados

Conforme apresentado na Seção anterior, um BD é composto basicamente por dois componentes: um conjunto de programas responsáveis pelo gerenciamento do acesso aos arquivos de dados e os dados, armazenados nesses arquivos. O primeiro, possui interfaces para a criação e manipulação dos dados, além de outras funcionalidades, como o controle de segurança e integridade. A segunda parte de um BD são os dados propriamente ditos, que normalmente estão dispostos em um ou mais arquivos seguindo uma estrutura física proprietária do SGBD em questão [AZR⁺12].

Os SGBDs tem como uma de suas características a independência entre estruturas lógicas de dados como tabelas, *views* e índices das estruturas de armazenamento físicas (estrutura de arquivos). Isto pode ser observado na Figura 3.1. Como as estruturas lógicas e físicas são separadas, a camada física de dados pode ser armazenada sem influenciar o acesso às estruturas lógicas. Uma analogia é renomear uma estrutura lógica, por exemplo uma tabela, e não renomear o arquivo físico que armazena essa tabela. Normalmente, estes arquivos físicos são criados no momento em que uma estrutura de dados lógica é definida.

Um arquivo de dados existe fisicamente no disco e é criado pelo SGBD. Nele são armazenadas as estruturas de dados, tais como as tabelas, índices e os registros. Estes arquivos que são

em um formato proprietário, normalmente não são acessados por outros programas. Os arquivos de dados podem estar localizados em um sistema de arquivos local do sistema operacional ou em um grupo de discos, como por exemplo, em uma *storage*.

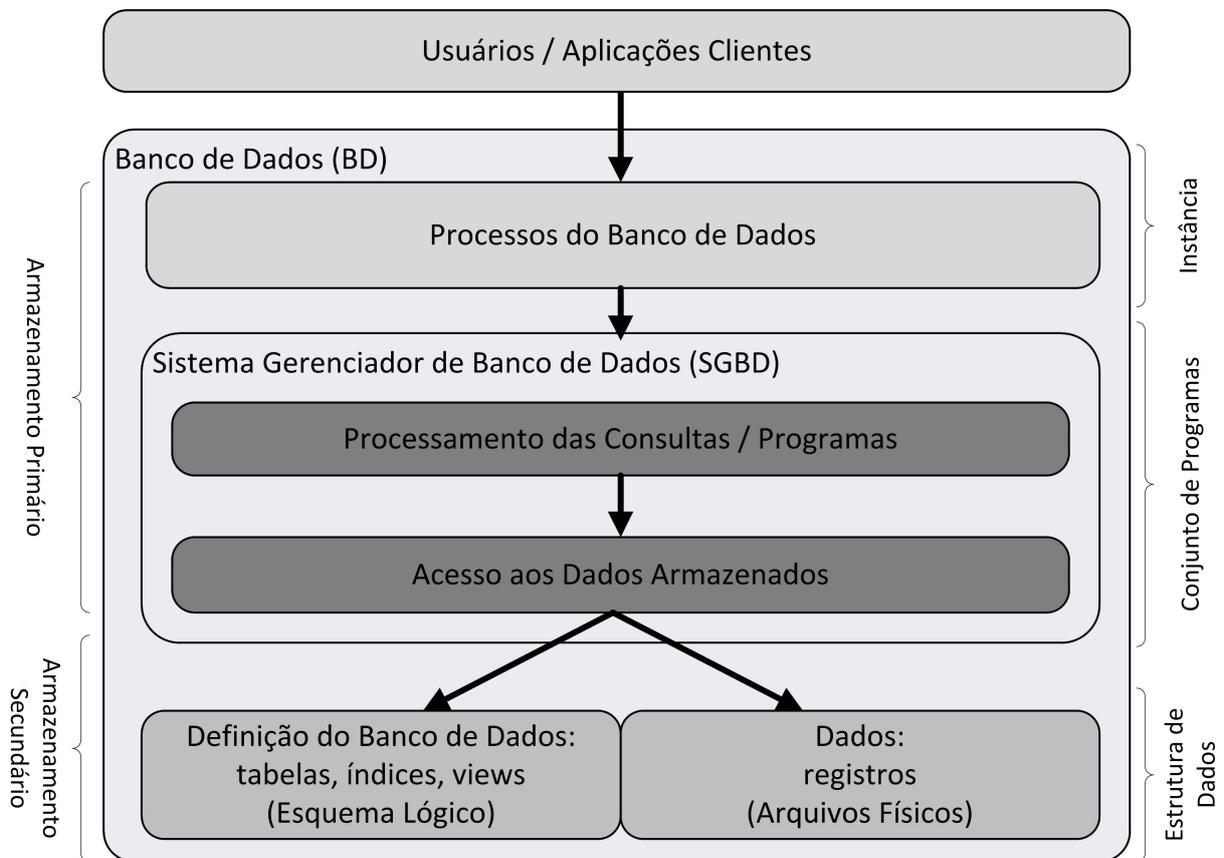


Figura 3.1 – Estrutura de um BD - adaptado de [Dat04]

Usando o armazenamento dos arquivos físicos em uma *storage*, pode-se obter uma independência superior das estruturas lógicas em relação aos arquivos físicos durante a execução do SGBD, comparado com a mesma execução a partir de um servidor onde o armazenamento é local. Nesta segunda abordagem normalmente os arquivos de dados são acessados por um único servidor de BD.

Entendendo a independência entre estruturas lógicas e físicas, o SGBD torna-se o responsável por permitir o acesso aos dados físicos através das estruturas lógicas, controlando o acesso e garantindo assim a segurança aos mesmos. Para isso, o SGBD cria uma estrutura de acesso identificada normalmente como instância de banco de dados (Figura 3.1) [Dat04].

Uma instância de BD ou simplesmente instância, opera como um serviço que trata todas as solicitações dos aplicativos clientes, intermediando os pedidos e retornando dados nos BD gerenciados por essa instância. Instância pode ser definido como o destino das solicitações dos aplicativos. A instância atende as instruções convertendo em operações nos objetos do BD e, se as permissões exigidas tiverem sido concedidas, executará a operação. Todos os dados recuperados são retornados ao aplicativo que fez a solicitação. É possível executar várias instâncias a um mesmo BD porém, uma instância pode acessar um único BD.

Neste contexto, este trabalho usa a definição de instância como a combinação dos processos sendo executados na memória e que são utilizados para a gerência e acesso aos arquivos de dados [Dat04]. Esta definição é muito importante para a correta compreensão e prosseguimento deste trabalho.

Os valores dos parâmetros de uma instância, como utilização de memória, tamanho e tipo de acesso, podem ser definidos no momento de sua criação. Esses parâmetros irão alocar recursos do sistema operacional que foi configurado e iniciar os processos responsáveis por manipular os arquivos de dados [SMA⁺08].

As instâncias são armazenadas em um nível diferente dos dados. As mídias de armazenamento de computador formam uma hierarquia de armazenamento que inclui duas categorias principais:

Armazenamento primário: inclui os recursos de armazenamento que podem ser acessados diretamente pela unidade central de processamento (CPU), como a memória principal e a memórias cache, que são em geral pequenas, se comparadas ao volume de dados armazenado. A vantagem da utilização deste tipo de armazenamento é a velocidade de acesso as informações, porém com uma capacidade reduzida [SKM08].

Armazenamento secundário: são identificados como os recursos de discos (ou *storages*). Esses dispositivos geralmente possuem maior capacidade combinado com menor custo, porém proporcionam o acesso mais lento, se comparado aos dispositivos de armazenamento primário. Os dados de um armazenamento secundário não podem ser processados diretamente pela CPU [SKM08].

Na Figura 3.1 pode ser observada a estrutura de um BD e a hierarquia de armazenamento. SGBDs podem armazenar e manipular grandes volumes de dados que normalmente devem ser mantidos por longos períodos de tempo. Os dados podem ser acessados e processados repetidamente ao longo da sua existência e, em alguns casos, devem permanecer e estar disponíveis permanentemente. Neste caso, os dados devem ser armazenados na memória secundária onde as alterações dos dados devem ser persistidas fisicamente.

Em contraste, a instância é armazenada na memória primária e permite o acesso aos arquivos de dados. Uma instância só existe durante o período de execução e no momento em que essa deixar de executar, a instância libera a memória primária permanecendo somente os dados persistidos. Quando uma instância é encerrada, os dados acessados por ela não estarão mais disponíveis aos aplicativos clientes.

3.3 Implementação de banco de dados em ambientes virtualizados

Na infra-estrutura, a virtualização pode ser a principal estratégia para auxiliar a atividade de administração dos recursos existentes. Atualmente os BD são disponibilizados em um ambiente virtualizado sobre uma máquina virtual [AAS08]. Assim, entre os vários desafios da implementação do BD em ambientes virtualizados, este trabalho destaca dois: o problema de desempenho dos

servidores virtuais em comparação a utilização de servidores físicos, que é uma preocupação comum, e a dificuldade de consolidação dos servidores virtuais sobre uma máquina física.

3.3.1 Desempenho em ambientes virtualizados

Não bastasse o alto consumo de recursos, BD também são normalmente bastante complexos quanto à sua correta instalação, configuração e otimização. Também deve ser levado em consideração o desempenho, observada como desvantagem da utilização de virtualização em servidores de BD. Porém, de acordo com estudos realizados [CRFD07] [LCR⁺12] [MYAS08], a perda de desempenho em média chega a dois dígitos percentuais, e é devido em grande parte às falhas de paginação (*Page Fault*). Uma falha de paginação é ocasionada quando os dados necessários para execução de determinada instrução não se encontram na memória principal, ou se encontrada, não está presente no espaço de endereçamento do processo corrente [MYAS08] [SMA⁺08].

Além do problema de paginação, outro problema que pode afetar o desempenho de um BD sobre MV são as camadas de *software* inseridas entre os recursos, tais como o *hardware* e o SO, haja visto que esse camada adicional pode comprometer os ciclos do processador, gerando um ônus de desempenho durante a execução [AMN06].

Em um ambiente de MVs, onde cada uma está executando um BD independente, elas estão competindo pelos recursos físicos de um servidor. Uma das dificuldades é alocar os recursos conforme a necessidade de cada BD. Considerando os diferentes tipos de recursos físicos, tais como memória, capacidade de CPU ou velocidade de I/O, atribuídas às MVs, a dificuldade é alocar uma parte ou fração de cada recurso da máquina física para cada MV. Desta forma, devem ser definidos os parâmetros de configuração dos recursos das MV na hora de inicialização e cada MV deve ser monitorada para identificar possíveis falhas ou gargalos [HLAP06].

Ao criar uma MV para uma ou mais aplicações é importante configurar corretamente os recursos. Uma decisão importante que deve ser feita ao configurar uma MV é decidir o quanto dos recursos físicos disponíveis serão destinados a cada MV. Atualmente, há pesquisas que propõem um sistema automático para tomar esta decisão nas MV que hospedam os BD [PPS⁺07] [SDG⁺07].

Ao considerar os pontos positivos sobre virtualização mencionados no Capítulo anterior, é possível destacar além daqueles, a consolidação de BD em ambiente virtualizados. O aumento da demanda por recursos ao BD pode ser facilmente resolvida através da gerência de BD em ambientes virtualizados com ferramentas específicas de virtualização [AAS08].

Difícilmente servidores de BD são totalmente monitorados durante a execução, independentemente do SO que está sendo utilizado. Para os administradores de BD, que são os responsáveis por manter cada servidor, esses servidores geram uma demanda particular no gerenciamento de desempenho, podendo ser considerado um desafio.

3.3.2 Gerência de recursos em ambientes virtualizados

Em contrapartida, uma das vantagens positivas que a virtualização fornece aos administradores desses ambientes de TI é a fácil migração das MVs entre os servidores físicos, aumentando assim o tempo de disponibilidade do serviço e consolidando os recursos [CJMB11].

Porém, essa facilidade ao longo da administração tem gerado um incômodo, pois a simples atividade de consolidar vários serviços de BD sobre um mesmo equipamento físico vem ocasionando além da perda de desempenho esperada da virtualizada, um tempo de inatividade, provocando uma queda de desempenho acentuada durante o processo de consolidação.

Esse tempo gasto de indisponibilidade durante o processo de migração do servidor de um BD, vem criando uma atenção especial nessa atividade. Como a MV pode ser movida de um servidor sem recursos para outro servidor com mais recursos, a migração pode ser considerada a principal atividade de sobrecarga que os administradores estão enfrentando atualmente no processo de consolidação [CJMB11].

Neste processo, a fase de migração de MV pode ser implementada usando várias técnicas diferentes para reduzir o impacto. Dentre as principais, este trabalho aborda:

Migração Offline/Online: a forma mais simples de migrar um servidor de banco de dados é parar o serviço, mover a MV para o novo servidor e iniciar o serviço no novo ambiente. Embora sejam consideravelmente simples e segura, todas as conexões e transações são perdidas durante a execução da migração [HLAP06].

Migração sob Demanda: para diminuir o tempo de inatividade do serviço, pode-se usar uma técnica de migração onde é iniciada uma nova cópia da instância de banco em outro servidor que recebe todas as novas conexões mantendo as antigas. Essa abordagem pode ser considerada mais eficiente que a anterior, porém produz uma sobrecarga alta após a migração para manter a sincronia entre origem e destino [SDG⁺07].

Migração em Tempo Real: semelhante à técnica anterior, porém para resolver o problema de sobrecarga de sincronia, essa técnica grava o estado do banco de dados (das transações) recuperando o seu estado atual após a migração [SDG⁺07].

Desta forma, para consolidar múltiplos BD sobre um servidor físico, devem ser utilizadas múltiplas MVs, uma para cada BD, dependendo do grau de abstração desejado. Nesta abordagem, durante o processo de consolidação, além dos dados do BD (SO e SGBD) serem migrados, há uma sobrecarga do tráfego, considerando que a MV também deve ser transferida.

Em suma, o desempenho um BD em um ambiente virtualizado comparado com um ambiente não virtualizado é menor [LCR⁺12]. Além da perda de desempenho, há também a dificuldade no processo de consolidação, onerando ainda mais a execução de um DB em um ambiente virtualizado, que é normalmente disponibilizado em uma MV. Vários fabricantes de SGBDs investigam esse nicho onde são oferecidas alternativas para contornar essa dificuldade. Na próxima Seção, são

apresentados algumas abordagens para implementação de alta disponibilidade que implementam uma das técnicas de migração do BD.

3.3.3 Alta disponibilidade em ambientes virtualizados

Alta disponibilidade trata da disponibilidade dos sistemas reduzindo paradas planejadas e não planejadas. Existem várias técnicas de alta disponibilidade para banco de dados, porém neste trabalho serão abordadas duas técnicas entre outras, a clusterização e a virtualização.

Clusterização: O ambiente de *cluster* consiste no uso coordenado de vários servidores (físicos ou virtuais), atuando como se fossem um único grande computador (conceito de *Grid Computing*). Nos servidores interligados, há apenas o sistema operacional e o software de banco de dados instalados (SGBD). Os arquivos de dados permanecem armazenados em uma área de armazenamento externa, normalmente (*storage*), compartilhando os dados (arquivos físicos) para múltiplas instâncias de BD dos servidores interligados [LPD⁺11].

Em caso de parada não planejada de um dos servidores do *cluster*, o próprio serviço de BD deverá detectar e transferir as sessões conectadas do serviço parado para o outro servidor automaticamente, sem interromper a disponibilidade do ambiente. Se a necessidade for a de substituir, retirar ou acrescentar servidores ao *cluster*, também não será necessário interromper a disponibilidade do ambiente.

Neste modelo são necessários que todos os nós do cluster estejam com o banco de dados ativo (Figura 3.2), gerando assim um ônus de monitoração entre as instâncias e sincronia de dados [LPD⁺11] [Jon06].

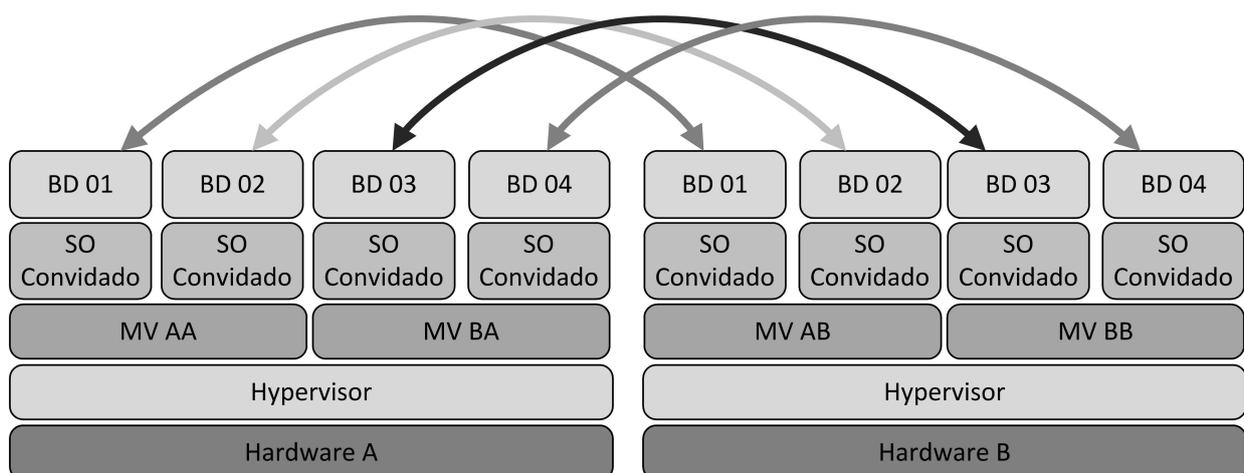


Figura 3.2 – BD em ambiente virtualizado usando Cluster.

Virtualização: a virtualização do BD é mais vantajosa em relação à técnica anterior porque somente uma instância BD estará sendo executada e, em caso de parada, somente esta precisará ser migrada juntamente com a MV quando for detectada a falha [Jon06].

Além de oferecer a capacidade de disponibilidade ininterrupta, o BD virtualizado pode oferecer a capacidade de aumentar os recursos verticalmente. Esta capacidade pode ser definida como uma técnica de potencializar os recursos de *hardware* para um determinado servidor virtual. Isso geralmente é feito através do aumento do número dos processadores ou adição de mais memória primária para a MV, porém sempre é limitada a capacidade dos recursos físicos disponíveis [BDF⁺03].

Um *hypervisor* normalmente possibilita aumentar os recursos lógicos da MV, com o impacto no desempenho do SGBD. Caso o servidor físico não suporte a liberação de mais recursos, a MV poderá ser migrada para outro servidor físico que comporte os parâmetros da MV (Figura 3.3). Conforme visto na seção anterior, em caso de necessidade de migração de uma MV para aumentar os recursos físicos no servidor físico, este processo não chega a gerar uma interrupção na execução do BD.

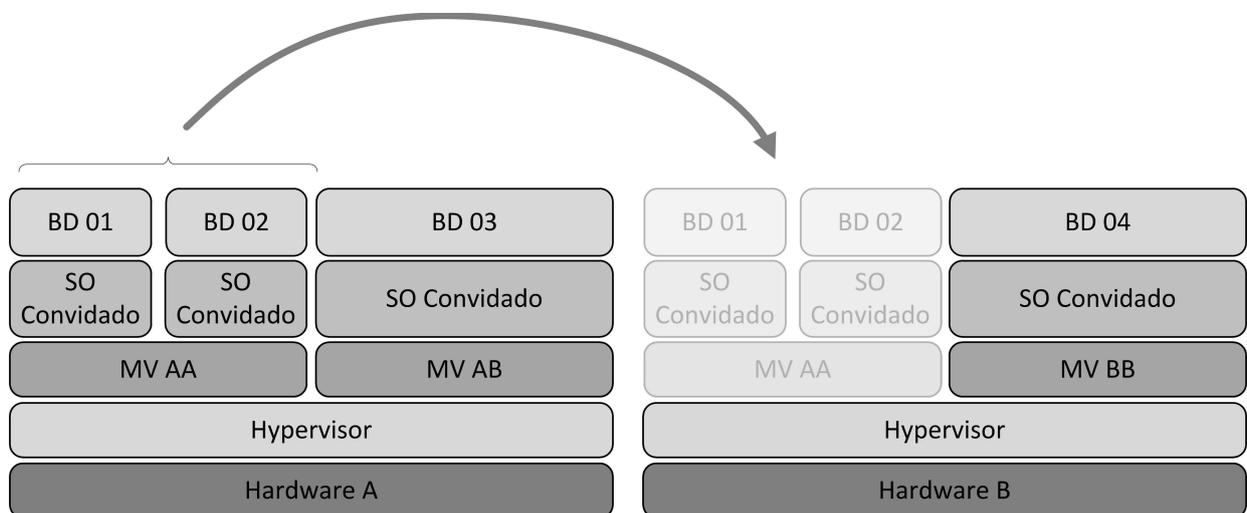


Figura 3.3 – BD em ambiente virtualizado usando MV.

Estudando as soluções apresentadas, a primeira apresenta o BD de forma distribuída, com vários SGBDs sendo executados em diferentes servidores, cada um deles com uma ou mais instâncias ativas, criando uma sobrecarga durante a sincronização dos processos executados no BD. Porém, em caso de migração devido a uma parada, a sobrecarga durante o processo de migração é menor. A segunda solução, que é mais comum, é a utilização de um SGBD sobre uma MV, que pode ser considerada a forma tradicional de virtualizar um BD. Neste modelo, somente um SGBD é instalado sobre uma MV e esta normalmente executa uma instância, eliminando a necessidade de sincronização de processos entre os servidores. Na Tabela 3.1 podem ser verificadas as principais diferenças entre as duas soluções para alta disponibilidade em ambientes de banco de dados virtualizados.

Porém, dentre as abordagens apresentadas para alta disponibilidade utilizando o processo de migração de BD, a segunda pode ser considerada mais vantajosa, devido ao fato desta necessitar apenas de uma MV para a execução do BD e não de múltiplas MV para manter o BD em execução ininterrupta.

Tabela 3.1 – Comparação entre os principais métodos de Alta Disponibilidade.

| Critério | Máquina Virtual | Clusterização |
|---|-----------------|---------------|
| Migração sessões online | Sim | Sim |
| Executa em ambiente Distribuído | Não | Sim |
| Migração é transparente para o usuário | Sim | Sim |
| Possibilita reconfiguração | Sim | Sim |
| Executa Múltiplas instâncias (requer sincronização) | Não | Sim |
| Sobrecarga durante o processo de migração | Sim | Não |
| Possui instâncias ociosas | Não | Sim |

3.4 Conclusões deste capítulo

Este Capítulo explorou os conceitos sobre BD. Estes conceitos são importantes para o entendimento do restante do trabalho. Entre os conceitos mais importantes, é possível destacar a definição de BD como um conjunto de programas e dados. Os programas (SGBD) são responsáveis por gerenciar e manter os arquivos de dados. Também é exposto o conceito de instância de BD como uma estrutura lógica que possibilita o acesso ao dados, mantido pelo SGBD e armazenado na memória principal.

Tendo como principais vantagens da utilização de tecnologias de virtualização, o fornecimento de forte isolamento, segurança e consolidação de servidores físicos [Jon06], pode ser inevitável a introdução da virtualização na arquitetura de administração de servidores de BD.

A virtualização de BD normalmente é implementada através da execução sobre uma MV. Este Capítulo explorou essa prática, abordando alguns aspectos, quais sejam, o problema de desempenho devido à inclusão de uma camada adicional de virtualização, o problema de consolidação de vários BD em um único servidor virtual que se dá principalmente através do processo de migração de uma MV, movendo a MV de servidor físico para outro e o benefício de alta disponibilidade inerente a esse tipo de ambiente, focando em clusterização e na utilização de MVs.

No próximo Capítulo é demonstrada uma alternativa às soluções convencionais abordadas, explorando os benefícios de implementação de uma abordagem alternativa.

4. NOVO PARADIGMA PARA VIRTUALIZAÇÃO DE BANCO DE DADOS

Embora a prática de execução de BD sobre MVs possa ser explorada, existem vários desafios a serem superados, dentre eles, o desafio de melhorar o desempenho e a dificuldade de consolidação. Conforme apresentado anteriormente, o primeiro é motivado pela inserção da camada intermediária entre o SGBD e o *hardware*, obrigando uma execução extra: a tradução de endereços virtuais para endereços físicos [AAS08]. Neste sentido, surge então a abordagem alternativa para utilização de BD em ambientes virtualizados: o Banco de Dados Virtual.

Este Capítulo explora essa abordagem alternativa, apresentando esse paradigma e comparando com as implementações estudadas. Primeiramente na Seção 4.1 são abordados os desafios da virtualização de BD; em seguida, na Seção 4.2 é apresentado o modelo de BD virtual com suas respectivas características, destacando as vantagens desta utilização; Nas Seções 4.2.1 e 4.2.1 são apresentados os processos de migração utilizando a abordagem convencional e a abordagem utilizando o BD virtual; e por último, na Seção 4.4 é apresentado a conclusão deste Capítulo.

4.1 Desafios da virtualização de banco de dados

Fabricantes de *hardware* e *software* estão incluindo cada vez mais funcionalidades de virtualização em seus produtos para dar suporte à crescente demanda por ambientes virtualizados [Jon06].

Assim, tornam-se cada vez mais escassos os argumentos para não utilizar a virtualização para BD. Cada vez mais, novos produtos estão sendo lançados visando à redução do impacto da virtualização de BD. Em um ambiente virtual de BD, normalmente os serviços são executados em ambientes virtuais isolados. Uma característica desses ambientes virtualizados é que múltiplos SGBDs podem ser executados simultaneamente em um servidor físico com isolamento total, tornando a virtualização ainda mais atraente.

Porém, sabendo que o desempenho de um servidor de BD em um ambiente virtualizado é menor, o desafio agora é o minimizar o *overhead*, embora conforme mencionado anteriormente, a perda de desempenho pode ser limitado em particular por tradução de endereços virtuais para a arquitetura física [Jon06]. Uma possível solução é desenvolver uma camada de aplicação utilizando a abstração entre o SO e o *hardware* físico, comunicando de uma forma mais eficiente. Essa camada não só é capaz de atender aos pedidos do ambiente virtual, mas também é capaz de confirmar o estado da aplicação SGBD e disponibilizar os dados restantes sobre todos os recursos físicos.

Outro desafio, dado o crescimento do ambiente de TI, é a consolidação de servidores virtuais sobre os servidores físicos [MYAS08]. Essa prática vem se tornando uma alternativa para dar suporte a ambientes complexos. Uma das características de um ambiente consolidado é a alta taxa de virtualização dos recursos. Uma estrutura bem consolidada pode gerar uma economia nos custos,

como por exemplo, a economia de consumo de energia, de refrigeração, melhor utilização dos recursos computacionais e com os custos de licenciamento sem perda de desempenho. Outros ganhos também podem ser uma melhor administração dos sistemas com maior flexibilidade para o ambiente, menor espaço ocupado e menos emissão de CO² para o meio ambiente. Porém, a consolidação não é um processo fácil, haja vista que este processo consiste em adequar os recursos virtuais da melhor forma possível, assegurando a correta alocação dos recursos físicos em um ambiente que normalmente não pode sofrer paradas para realizar tais atividades. Nesse aspecto, permanece o desafio de consolidar um ambiente com BD de forma eficiente, executando uma migração de um SGBD e alocando os recursos da melhor forma possível.

Para manter os ambientes de alta demanda sempre disponíveis, devem-se ter como premissa a alta disponibilidade, e neste contexto, um BD também pode ser considerado como um fator fundamental em um ambiente de TI [VBVB09]. Dentre as soluções normalmente utilizadas e apresentadas anteriormente, temos a clusterização (replicação) e a utilização de MV para suportar o BD em um ambiente de TI, que fornecem um padrão de serviços ininterruptos baseadas em migração do ambiente, porém também ocasionam sobrecarga aos administradores desses ambientes. O desafio neste caso é oferecer um padrão de serviço ininterrupto com o menor tempo de indisponibilidade, gerando o menor ônus possível na administração, facilitando assim a atividade do administrador.

Embora a utilização de virtualização tenha atrativos, vistos na Seção 2.2, é nítido que ainda há desafios de melhorias que podem ser alcançadas. Pesquisas atuais buscam alternativas para superar tais desafios, comparando as abordagens de virtualização de BD [AAS08] [AZR⁺12]. Alguns trabalhos indicam o emprego de BD em ambientes virtuais utilizando técnicas alternativas de virtualização, como por exemplo, o uso de *containers* que apresentam um melhor desempenho se comparado com a execução de MV [XNR⁺13]. Embora cada solução possua seus benefícios e limitações, um problema que é compartilhado por todos é o desempenho.

A seguir, é apresentada a abordagem alternativa à virtualização de BD baseadas em MV, demonstrando suas características e benefícios, realizando uma comparação entre as implementações da abordagem convencional que utiliza MV e o modelo alternativo.

4.2 Modelo do banco de dados virtual

Nesta abordagem é proposta a virtualização da instância de BD, realizado pelo SGBD. Essa implementação pode aumentar a flexibilidade através de uma arquitetura muito mais eficiente, neste caso, a virtualização da instância de BD [AZR⁺12]. Desta forma, algumas desvantagens da utilização de um sistema de BD sobre uma MV podem ser minimizadas, como por exemplo, a perda de desempenho sendo possível manter os benefícios que a virtualização pode trazer, entre eles, a consolidação de BD.

Banco de dados virtual é implementado através da virtualização da instância de BD sobre um sistema operacional com recursos exclusivos para executar esse tipo de função. Os aplicativos

que fornecem acesso ao banco de dados (SGBD) são instalados em cada servidor que irá prover o serviço de virtualização da instância. Em um único servidor podem ser virtualizadas várias instâncias de BD, possibilitando o gerenciamento individual de cada instância que é tratada como uma extensão do SO.

Esta técnica é similar à virtualização baseada em *containers*, disponíveis em alguns sistemas operacionais modernos (por exemplo, *Linux Containers* [Lin12], *FreeBSD Jails* [Fre12] e *Solaris Zones* [Sol12]). 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 SO [XNR⁺13]. De maneira análoga, a virtualização em nível de BD permite a criação de múltiplas instâncias de BD, que são isoladas entre si, sem a necessidade de ter uma MV que replica toda a pilha de SO e o próprio SGBD.

Na Figura 4.1 são apresentadas as camadas de *software* necessárias à implementação de um BD. A Figura 4.1(a), mostra a estrutura de um BD sobre o ambiente nativo. Esta é a forma tradicional de executar um BD sem os recursos de consolidação. A alta disponibilidade pode ser alcançada mediante a replicação do ambiente em máquinas físicas.

A Figura 4.1(b) mostra a implementação de BD sobre uma MV. É possível notar o acréscimo nas camadas de *software*, causado pela adição da camada de *hypervisor*. Este tipo de implementação possibilita a consolidação de BD e a implementação da alta disponibilidade sem a obrigatoriedade de replicar o ambiente.

Finalmente, na Figura 4.1(c) é apresentada a estrutura de *softwares* da abordagem de BD virtual. Nesta abordagem não há uma camada adicional de *hypervisor*, permanecendo a função de virtualização no SGBD, que através de modificação específicas no SO possibilita o gerenciamento individual de cada instância de BD.

Este tipo de virtualização permite que um tipo de SO conviva com diversas outras instâncias do mesmo sistema, compartilhando os recursos físicos, definição esta já apresentada neste trabalho na Seção 2.3.4. O ambiente virtual funciona como um processo do sistema anfitrião, mediante alterações no SO. Porém, a necessidade de alterações no SO anfitrião e a obrigatoriedade de que todos os ambientes virtualizados sejam do mesmo tipo, podem ser consideradas desvantagens, haja visto que limita a virtualização à ambientes homogêneos.

A semelhança do desempenho ao ambiente nativo se justifica devido a camada intermediária entre o SGBD e o *hardware* ser menor (Figura 4.1) diminuindo o *overhead* causado pela inclusão de uma camada intermediária de *hypervisor*. Considerando que o SO não é virtualizado, como em uma MV, e que somente as instâncias do BD é virtualizada, não é necessário traduzir cada instrução do SGBD para o SO hospedeiro. Observado que o recurso de virtualização está no *kernel* do SO, as instruções podem ser executadas diretamente no *hardware*. Neste modelo, o *hypervisor* tem como uma de suas funções controlar os recursos disponíveis para que todas as instâncias acessem os recursos conforme a configuração individual [AZR⁺12]. Tal característica permite que as instruções não necessitem ser traduzidas ou repassadas para o *hardware* hospedeiro pela camada intermediária.

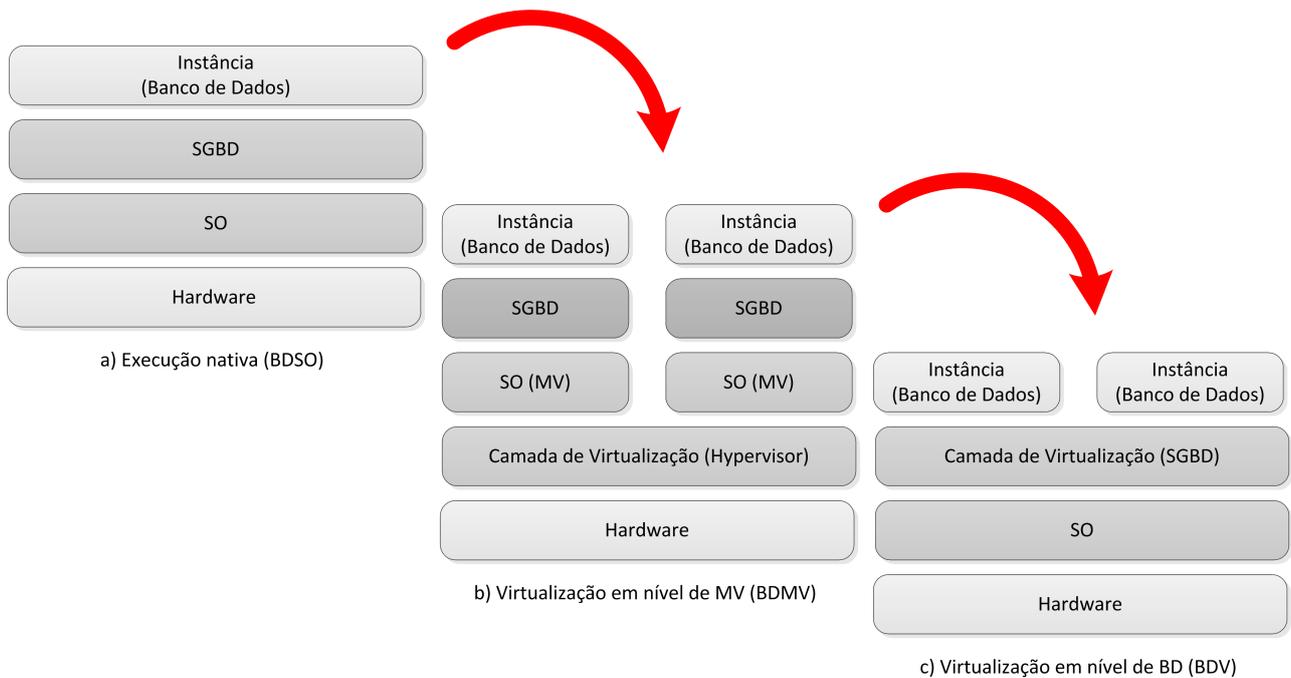


Figura 4.1 – Comparação entre as camadas de BD.

ria, possibilitando a execução de algumas instruções diretamente, justificando assim o desempenho semelhante ao ambiente nativo.

Considerando as motivações para virtualizar o BD e os seus desafios, esta abordagem (Figura 4.1(c)) é uma alternativa à administração eficaz de um ambiente de TI, mantendo os benefícios conquistados nesses ambientes virtualizados e trazendo algumas vantagens, tais como uma consolidação mais eficiente, diminuição do *overhead* e alta disponibilidade.

4.2.1 Gerência de recursos de BD virtual

A administração das instâncias virtuais é uma atividade muito importante no gerenciamento do BD e pode ser considerada uma vantagem, dada a facilidade de se manusear as instâncias virtuais, porém pode também ser considerada um problema, caso não se tenha ferramentas adequadas para o controle das instâncias ou até mesmo caso se crie um número excessivo destas, observada a facilidade desta operação.

Em um ambiente de BD virtual, é possível consolidar a instância de BD de forma muito mais eficaz a abordagem apresentada anteriormente. Neste tipo de consolidação, é possível monitorar os recursos disponíveis de *hardware*, adicionando ou removendo recursos ao BD virtualizado. Além de alocar os recursos conforme necessário, é possível suportar vários serviços de BD virtuais em um único servidor físico, realocando os serviços em outros servidores conforme necessário [AZR⁺12]

A atividade de realocação de instâncias de BD é realizada normalmente através do processo de migração. Neste aspecto, essa atividade é uma das que mais teve ganho de desempenho. O

processo de migração de um BD sobre uma MV e a migração da instância virtual são descritos a seguir.

Consolidação mediante migração de BD sobre MV

A migração de um BD sobre MV utiliza uma técnica para transferir o sistema convidado de uma máquina física para outra, com a mínima interrupção de serviço, o que é fundamental para permitir a consolidação. O processo de migração de um BD sobre uma MV, pode ser executado da mesma forma que migrar uma VM com qualquer outra aplicação. Os passos necessários para realizar a migração são:

1. **Iniciar MV no host de destino:** Nesta fase é iniciada a MV no *host* de destino. Este processo inicia com a cópia de todas as páginas de memória do *host* de origem ao *host* de destino. Se durante a cópia uma página de memória sofre alteração, é necessário realizar uma nova cópia da página inteira, até um limite onde o *hypervisor* considerar que a maioria das páginas foram copiadas (Figura 4.2(a) e 4.2(b)).
2. **Verificar o estado da MV copiada:** Esta fase é executada no momento em que a cópia das páginas de memória da MV atingirem o limiar definido pelo *hypervisor*. Para finalizar a cópia, a VM de origem é suspensa para que seja finalizada a cópia das páginas de memória para a VM de destino. Esta suspensão e cópia das páginas deixadas no hospedeiro de origem é considerado um tempo de inatividade e pode durar de milissegundos até vários segundos, dependendo do tipo de aplicação que está sendo executada (Figura 4.2(c)).
3. **Finalizar origem e iniciar destino:** Após finalizar o processo de cópia das páginas de memória da MV de origem, esta é desligada e a VM de destino passa a responder a todas as requisições (Figura 4.2(d)).

Este processo de consolidação possibilita mover um servidor de BD de um *host* hospedeiro com poucos recursos disponíveis para outro *host* hospedeiro com mais recursos, propiciando a inclusão de desempenho à VM mediante a adição de mais recursos, como por exemplo, mais memória ou núcleos de processador.

É importante destacar que neste trabalho não é considerada a consolidação de MVs que estão em diferentes áreas de armazenamento e que todas as MVs devem estar nas mesmas áreas de rede (mesma LAN).

Consolidação mediante migração de BD virtual

A consolidação de BD virtuais pode ser realizada de forma mais flexível, visto que neste tipo de migração não é necessário copiar o estado da MV, considerando que somente serão transportadas as sessões ativas ao final da execução da transação no BD, sem a mínima necessidade de interrupção

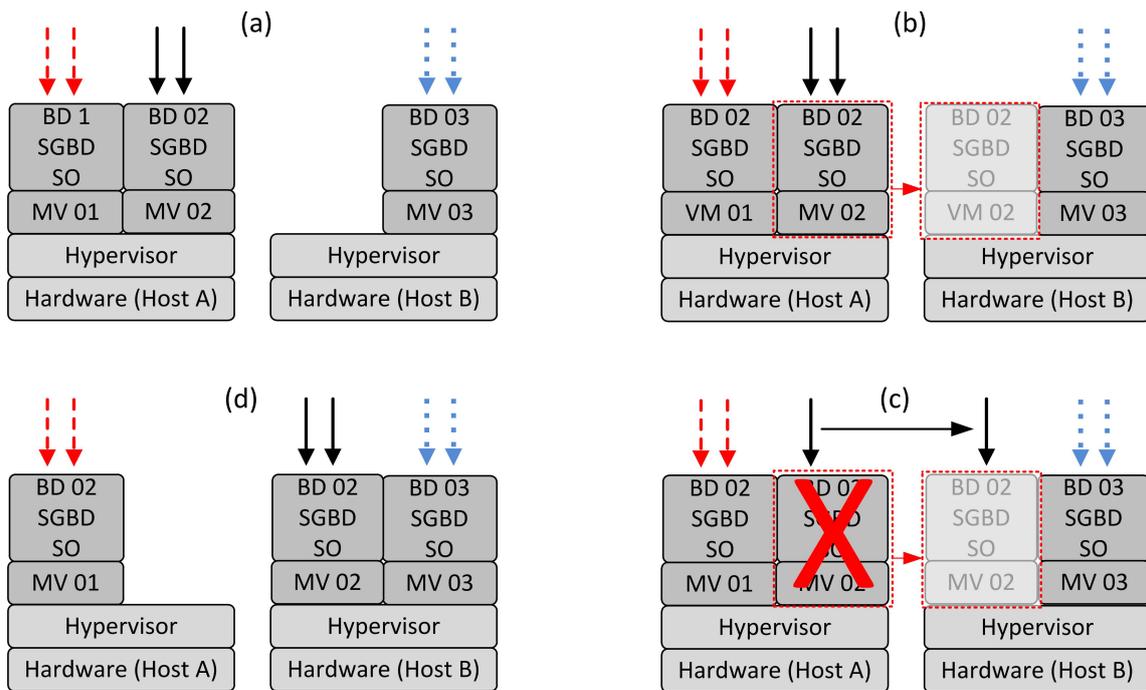


Figura 4.2 – Processo de migração de um BD sobre MV.

do serviço, minimizando ainda mais a perda de desempenho durante esse processo. Este processo pode ser feito através das seguintes etapas:

1. **Iniciar a instância no *host* de destino:** nesta fase é iniciada uma nova instância de BD no *host* de destino. (Figura 4.3(b));
2. **Transportar as sessões ativas:** em seguida, todas as sessões ativas são transportadas do *host* de origem para o *host* de destino. O transporte da sessão é realizado somente após o término da execução da transação, dispensando assim a necessidade de sincronização do estado do sistema. No caso em que a sessão não esteja executando uma transação ou que já tenha concluído, a transferência é executada imediatamente (Figura 4.3(b) e 4.3(c));
3. **Finalizar instância no *host* de origem:** a última fase é executada quando todas as sessões são transferidas para o *host* de destino e este passar a responder às requisições do BD, sendo finalizada a instância de origem. (Figura 4.3(d)).

É possível observar que neste tipo de migração é necessária apenas para migrar a instância de BD (gerenciado pelo SGBD) e não a pilha completa da *software* representado na Figura 4.1(b). Assim como na migração anterior, é importante destacar que a migração de BD virtual ocorre em um ambiente da mesma área de armazenamento e em uma mesma área de rede.

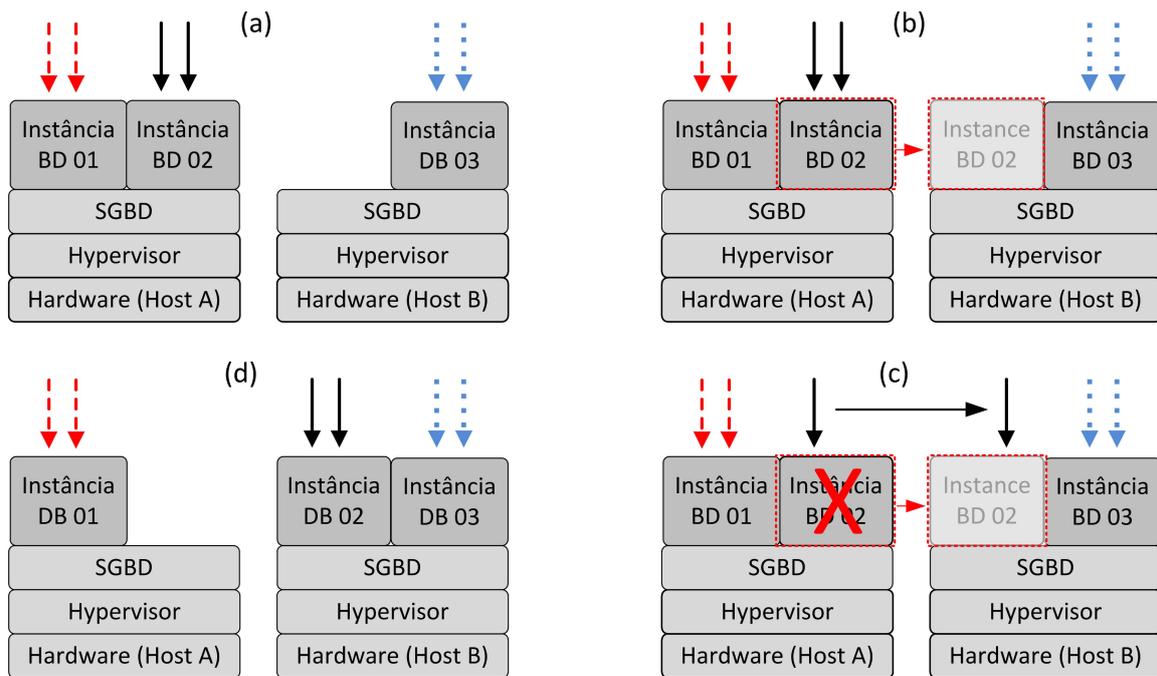


Figura 4.3 – Processo de migração de um BD virtual.

4.2.2 Alta disponibilidade usando BD virtual

Alta disponibilidade usando BD virtual pode ser alcançada de forma mais simples observado que **não é necessário replicar o ambiente**. Um ambiente de alto desempenho não virtualizado normalmente replica os serviços em ambientes semelhantes, para que em caso de parada o outro assuma. Como alternativa, nesta abordagem é realizada a virtualização da instância de BD, eliminando a necessidade de replicar o ambiente. Em caso de parada não programada, é necessário iniciar o serviço em outro ambiente disponível.

Outro fator que pode motivar a utilização desta abordagem é que assim como em uma MV, a **instância virtualizada suporta todos os métodos que uma MV** oferece, possibilitando a consolidação de vários serviços sobre um único servidor físico, porém com mais eficiência, tal como a perda mínima de desempenho durante o processo de migração.

Além da possibilidade de mover um BD quando uma falha for detectada, ainda é possível realizar o balanceamento de carga de trabalho realocando o serviço de BD em um servidor desocupado.

4.3 Comparação entre as abordagens de utilização de BD em ambientes virtuais

Com a grande variedade de opções disponíveis de sistemas de SGBDs, algumas considerações podem ser levadas em conta durante o processo de escolha, como por exemplo: SO, licenciamento e capacidade de execução. Para realizar a comparação entre os SGBDs atuais foram

analisados na literatura as principais alternativas utilizadas pela comunidade acadêmica e industrial. Dado o foco deste trabalho em virtualização do BD, a análise é restrita ao modelo de execução em um ambiente virtual. Os sistemas de BD abordados nesta comparação podem ser executados também em um ambiente não virtualizado, diretamente no *hardware* hospedeiro, porém sem os benefícios da virtualização. Dentre as abordagens estudadas, este trabalho destaca:

Banco de dados sobre máquina virtual: os BDs estão completamente isolados uns dos outros mediante a execução de MV sobre o *hardware* hospedeiro. O desempenho é parcialmente dependente da capacidade do *hypervisor* em gerenciar as requisições da MV ao *hardware*.

Uma MV possibilita executar múltiplas BD sobre um mesmo *hardware* motivado que uma MV desconhece que está em um ambiente virtualizado e que existem outras MV competindo por recursos de *hardware*. Cada BD normalmente é executado em um MV, assim podem ser necessárias múltiplas MV para múltiplos BD.

Os arquivos de dados de cada MV podem ser armazenados em discos locais (internos a MV) ou discos compartilhados. Os discos locais de cada MV existem como arquivos no SO hospedeiro o que os torna muito fáceis de fazer o *backup*, mover ou implantar, além de fornecer o ambiente ideal para desenvolvimento e teste. Esta prática exige mais recursos para migrar uma MV, dado que os arquivos de dados também serão movidos. Utilizando disco compartilhando, os arquivos de dados permanecem em um local compartilhando, permitindo acesso aos dados por qualquer MV.

Banco de dados virtual: o sistema anfitrião executa cada BD (instância virtual) como um processo do SO, razão pela qual os BDs desconhecem que está sendo executado vários BDs sobre o mesmo *hardware*. Nesta abordagem, não é necessário reinstalar toda a pilha a *software*, como SO e SGBD, sendo que estes são compartilhados do sistema anfitrião.

Nesta abordagem, os arquivos de dados são normalmente armazenados em disco compartilhado devido à flexibilidade de executar a migração de um BD durante a sua execução.

Embora cada abordagem possua benefícios e limitações, um problema de desempenho ainda é compartilhado por todos. Na Tabela 4.1 são resumidas as principais características de cada abordagem.

Tabela 4.1 – Características das implementação de BD sobre MV e BD virtual.

| Critério | Banco de dados sobre MV | Banco de dados virtual |
|--------------------------|------------------------------------|--------------------------|
| Virtualização | Máquina virtual | Processo do SO anfitrião |
| Camada Adicional | Hypervisor | Sem camada adicional |
| Migração | Toda a MV | Sessões ativas |
| Localização transparente | Sim | Sim |
| Consolidação | Múltiplas MVs | Múltiplos processos |
| Arquivo de Dados | Discos compartilhados, disco local | Discos compartilhados |
| Tipo de Ambiente | Heterogêneo | Homogêneo |
| Desempenho | Perda de Desempenho | Semelhante ao Nativo |

Embora essa alternativa de BD virtual pareça ser mais eficiente em relação à abordagem convencional que utiliza MV, é necessário identificar o quanto é mais eficiente essa nova abordagem. Para isso, o próximo Capítulo aborda entre outras, uma avaliação e comparação de desempenho das abordagens.

4.4 Conclusões deste capítulo

Neste Capítulo foi apresentado uma alternativa à abordagem de virtualização de BD sobre uma VM, o BD virtual. Nesta abordagem, apenas a instância de BD é virtualizada através do uso de *containers*. Este paradigma surge como opção para superar os desafios que normalmente são impostos à virtualização tradicional.

Ambientes de TI normalmente possuem dezenas, chegando muitas vezes a centenas de recursos, tais como servidores, *storages* e ativos de rede. Na atual evolução das tecnologias de virtualização, muitos desses recursos acabam sendo virtualizados e conseqüentemente os servidores de BD também são virtualizados. Porém, garantir a correta alocação dos servidores de BD nos ambientes virtuais e diminuir o *overhad* causado pela implementação da camada adicional de *hypervisor* são os principais desafios que se pretende superar com esta abordagem.

A abordagem de BD virtual pode ser implementada através das tecnologias semelhantes a *containers*, que implementam a virtualização a nível de *kernel*, onde cada ambiente virtual é executado como um serviço do SO hospedeiro. Uma das principais característica desse tipo de virtualização é que o desempenho do ambiente virtual é semelhante ao ambiente nativo, e cada ambiente é executado de forma independente e isolada entre si.

Durante a discussão do novo modelo, foram confrontadas as funcionalidade de consolidação e alta disponibilidade com o modelo convencional. Um processo de consolidação eficiente deve ocorrer sem a necessidade de parar o serviço ou, com uma queda mínima de desempenho. Nesse aspecto, a nova abordagem traz um ganho significativo durante o processo de migração, um dos recursos essenciais à consolidação. Porém, no quesito de alta disponibilidade, as duas abordagens tem um funcionamento muito semelhante, pois executam o serviço de BD virtualizado, porém estudos sugerem que a virtualização baseada em *containers* é mais eficiente pois somente a instância de BD é virtualizada e não toda a pilha de *software*, representado na figura 4.1.

Neste sentido, essa nova abordagem de virtualização do BD traz alguns benefícios em comparação a abordagem tradicional, os quais serão avaliados no próximo Capítulo.

5. AVALIAÇÃO DE BANCO DE DADOS VIRTUAL

Avaliar um ambiente virtualizado não é uma atividade naturalmente fácil, considerando a diversidade dos sistemas existentes e as características de implementação [WWS⁺10]. Devido a essa diversidade, foi necessária a padronização dos métodos para avaliar o desempenho, contemplando os mais diversos sistemas computacionais, tais como BD sobre MV e BD virtuais.

Neste Capítulo serão descritas as estruturas do ambiente de avaliação e as avaliações que foram realizadas com suas respectivas observações que puderam ser obtidas a partir da análise desses resultados. As Sessões estão organizadas como segue: Sessão 5.1 é apresentada a metodologia de avaliação, a seguir, na Sessão 5.2 são mostrados os resultados dos experimentos e, na Sessão 5.3 é abordada a matriz de comparação. Na Sessão 5.4 mostra os resultados alcançados com a avaliação do ambiente real utilizando a matriz de comparação e, por último, na Sessão 5.5 são colocadas algumas conclusões deste Capítulo.

5.1 Metodologia de Avaliação

Foram definidos os seguintes cenários para avaliar o desempenho de execução, consolidação e capacidade de isolamento de BD em ambientes virtualizados, quais sejam:

- BDs suportados em um ambiente nativo, ou seja, implementado diretamente no *hardware* serão aqui identificados como **Banco de Dados sobre Sistema Operacional (BDSO)**, cuja estrutura de *software* pode ser observada na Figura 4.1(a). A principal característica deste tipo de implementação é o desempenho nativo, por não apresentar a camada intermediária de virtualização, porém os recursos de consolidação e gerência de recursos não se aplicam neste tipo de implementação.
- Ambientes virtualizados que dão suporte a BD executados sobre MV, serão aqui identificados como **Banco de Dados sobre Máquina Virtual (BDMV)** e podem ser observados na Figura 4.1(b). A utilização desta abordagem adiciona uma camada intermediária responsável pela virtualização, provocando uma queda de desempenho, porém adicionando os recursos inerentes à virtualização, tais como consolidação e gerenciamento de recursos.
- A virtualização da instância de BD, neste trabalho será identificado como **Banco de Dados Virtual (BDV)**. Essa abordagem utiliza a virtualização baseada em *containers* e as camadas de *softwares* podem ser observadas na Figura 4.1(c). Uma das principais vantagens na utilização desta abordagem é a possibilidade de utilização de recursos inerentes à virtualização sem a perda de desempenho ocasionada pela inclusão de uma camada intermediária.

Com isto, a avaliação é realizada utilizando medições e comparações de desempenho dos ambientes BDMV e BDV. Como parâmetro de comparação, também foram executadas avaliações

de desempenho no BDSO. Outra análise realizada é a utilização de recursos, como CPU, memória, rede e consumo de energia dos cenários definidos.

5.1.1 Ambiente de execução

O ambiente de execução é formado por dois servidores Dell PowerEdge 810 com 2 Intel®Xeon®6500 com 64GB de memória RAM cada um, interligados com *storage* DELL EqualLogic PS400 dedicado utilizando o protocolo *ISCI*. A rede está dividida em duas subredes distintas, uma pública, para gerenciamento dos servidores e outra privada, dedicada à comunicação entre os servidores e o *storage*, ambas utilizando uma conexão GigaBit Ethernet.

O SO utilizado é o OEL 5.6 (Oracle Enterprise Linux 5 update 6) com kernel 2.6.18-238.el5. O SGBD utilizado é o Oracle®11g r2 e o *hypervisor* é o VMware ESXi 5. Os detalhes da arquitetura do ambiente de teste podem ser observados na Figura 5.1.

A escolha do SGBD é justificada pelo fato de este possuir as funcionalidade e características descritas na Sessão 4.2 e que são necessárias para executar os experimentos desejados. O estudo do estado da arte demonstra que atualmente este pode ser, o representante da tecnologia de virtualização por *containers* para BD.

O *hypervisor* utilizado para a MV é legitimado devido ao fato do fabricante do SGBD apresentar que este *hypervisor* possui o menor *overhead* de desempenho comparado com outros, sendo considerado como a melhor alternativa para o ambiente virtualizado em questão [AZR⁺12].

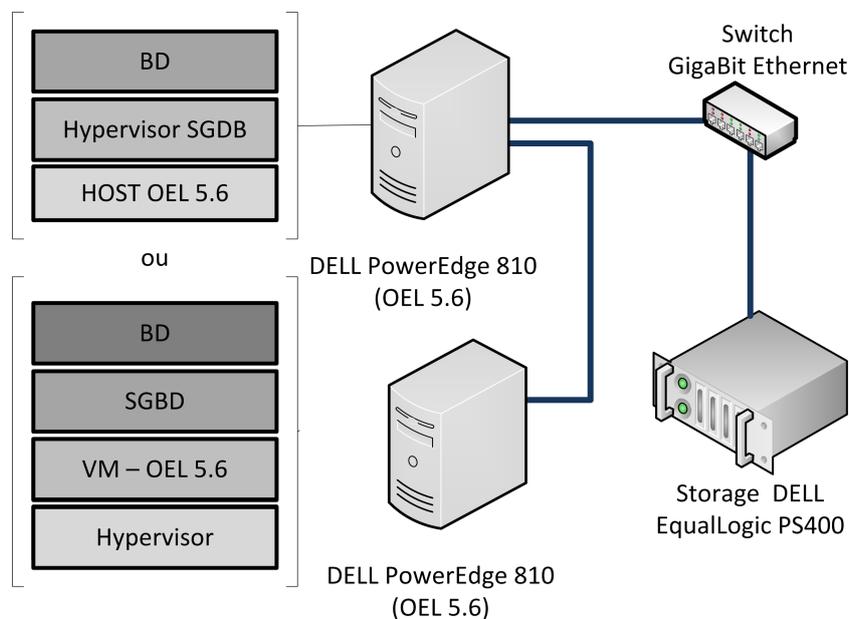


Figura 5.1 – Arquitetura do ambiente de avaliação (BDV e BDMV).

O ambiente de avaliação BDSO foi configurado utilizando o OEL 5.6 sem suporte a virtualização. O SGBD foi configurado diretamente sobre o SO e todos os recursos de *hardware* são disponibilizados diretamente pelo SO.

No ambiente DBMV, o SO foi configurado sobre uma MV com todos os recursos de *hardware* disponíveis. O SGBD foi instalado diretamente no SO da MV. A instância de BD foi criada aplicando 100% dos recursos disponíveis da VM. Neste ambiente, os recursos disponíveis ao BD são gerenciados pelo SO da VM.

No ambiente VDB, o BD foi configurado sobre o SO e este instalado diretamente sobre o *hardware* hospedeiro. A instância de BD foi criada utilizando 100% dos recursos disponíveis, porém neste ambiente o suporte à virtualização foi habilitado no SGBD.

5.1.2 Benchmark e descrição dos testes

Contemplando os mais diversos sistemas computacionais, dentre eles podemos citar OLTP (*On-Line Transaction Processing*) [tpc13] e DSS (*Decision Support System*), é necessário definir o tipo de operação que deseja testar. Neste caso, para avaliar o *overhead* da camada de virtualização no desempenho de BD para operações típicas de aplicações que demandam recursos computacionais, foram utilizadas operações do tipo On-Line Transaction Processing (OLTP) [tpc13].

Sistemas OLTP são caracterizados por suportar vários usuários simultâneos com execução de múltiplas transações (operações em banco de dados, como por exemplo, seleção, atualização e exclusão de dados) simultâneas. Estes tipos de transações permitem avaliar as operações típicas de ambientes que exigem uma elevada disponibilidade de acesso simultâneo aos dados com alto desempenho.

Para analisar o desempenho de sistemas OLTP, foi utilizado o *benchmark* do tipo TPC-C. Este *benchmark* é definido pelo *Transaction Processing Performance Council* (TPC) [tpc13]. TPC é um órgão que define os métodos para avaliar o processamento de transações para *benchmark* de BD, apresentando os resultados de desempenho categorizados para a indústria. O *benchmark* TPC-C continua a ser um ponto de referência popular para comparar o desempenho de OLTP em *hardware* e várias configurações de *software*. Como resultado TPC-C fornece informações sobre a quantidade de transações executadas por minuto (TPM). Existem cinco operações básicas que representam o comportamento de um sistema OLTP e podem ser observadas na Tabela 5.1 [tpc13].

Tabela 5.1 – 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% |

O TCP-C pode ser implementado por qualquer *software*, desde que obedeça aos requisitos de implementação definidos pela *TPC*. Neste trabalho foi utilizado o *software* de *Benchmark Hammerora* [ham13]. A ferramenta Hammerora pode ser definida como um gerador de transações de BD, de código fonte livre. Foi projetada para executar testes de estresse em BD, monitorar e gerar relatórios de desempenho. O número de clientes virtuais a serem utilizados e a quantidade de transações são parâmetros configuráveis na ferramenta.

O ambiente de teste foi configurado de maneira que apenas o SGBD estivesse em execução em cada caso de teste e tanto o SO quanto o SGBD sempre estivessem no mesmo estado inicial para cada teste.

Foram realizadas 10 execuções do *benchmark* para cada caso de teste, variando de 10, 50, 100 e 200 conexões simultâneas, com intervalo de confiança chegando a 95%. Em todos os casos de avaliação de desempenho, foi utilizada apenas uma instância do BD. Para executar a avaliação de escalabilidade e isolamento, foram executadas várias instâncias concomitantes e concorrentes, com o mesmo número de execuções e variações de conexões do teste de desempenho.

Com o objetivo de avaliar a utilização de recursos, como CPU, memória, rede e energia, foram utilizadas as ferramentas *dstat* em conjunto com *sysstat* disponíveis à versão do SO utilizado. Os resultados dessa avaliação são mostrados na Seção 5.2.1.

Na avaliação do processo de migração, utilizado para consolidar BD, foi utilizada apenas uma instância de BD ou VM por vez. Durante a migração, o volume das transações foi determinada pela execução de 10 *warehouses* de dados, com 10, 50, 100 e 200 conexões simultâneas. Em cada caso de teste, os recursos de arquitetura foram alocados em sua totalidade, ou seja, 100% de todos os recursos disponíveis, como memória e CPU. A avaliação consiste em migrar a instância de BD ou MV, com o *benchmark* em execução, monitorando os recursos, tais como memória, rede, CPU e TPM dos servidores durante o processo de migração. Os resultados são mostrados na Seção 5.2.3.

Para avaliar o isolamento das instâncias de BD, foram realizados dois experimentos, quais sejam: o primeiro experimento avaliou os efeitos sobre o desempenho das instâncias ao diminuir o número de CPUs por instância durante a execução do *benchmark*. Inicialmente, a execução de uma instância de banco de dados iniciou com 16 núcleos de CPU e durante a execução, o número de CPUs foi alterado de 16 para 8, diminuindo os recursos disponíveis à instância de banco de dados. O objetivo deste experimento foi avaliar o tempo necessário para a consolidação da alteração dos recursos e, o impacto sobre o banco de dados causados por esta mudança.

O segundo experimento de isolamento é para verificar se o isolamento realmente limita a carga da CPU e memória. Para isso, foram realizadas execuções de referência utilizando quatro instâncias com quatro núcleos de CPU e durante a execução, o limite de uma instância de BD foi alterado para 16 CPUs. O objetivo deste experimento foi comparar o desempenho entre as execuções: antes e depois da mudança de CPU e se uma instância afeta o desempenho de outra instância de BD. Os resultados destes experimentos são mostrados na Seção 5.2.2.

5.2 Resultados Obtidos

Esta Seção apresenta os resultados da avaliação de desempenho dos cenários BDSO, BDMV e BDV. Também são apresentados os resultados das observações dos processos de migração e isolamento necessários à avaliação de cada cenário.

5.2.1 Avaliação de desempenho

Para avaliar o desempenho foi executado o TPC em ambientes de 10, 50, 100 e 200 usuários com 100.000 operações cada. Estes valores representam os diferentes níveis de processamento para OLTP. Outros trabalhos, como [VBVB09], utilizaram valores dentro dessas faixas para avaliarem SGBDs.

Na Figura 5.2 é mostrada a execução do *benchmark* TPC-C para 50 usuários com 100.000 transações cada. Neste contexto, o desempenho é determinado pelo TPM dos cenários BDSO, BDMV e BDV. No monitoramento de desempenho foram utilizados os relatórios gerados pela própria ferramenta Hammerora.

Comparação de Execuções de Transações (TPM)

O desempenho do BD executado diretamente sobre o *hardware*, neste contexto BDSO, com média de 175.487 TPM, em comparação com o DBVM, com média de 161.227 TPM, é superior em 9%. Esse comportamento é esperado e pode ser explicado, como visto em [CRFD07], devido à inclusão de uma camada *hypervisor* necessária para a execução da MV. Além da inclusão da camada de virtualização, que interfere no desempenho das instruções do processador, a quantidade de operações de memória, como por exemplo *Page-Faults*, também afeta o desempenho da MV em até dois dígitos percentuais [MYAS08].

É possível observar também o resultado da execução do BDV com 175.957 TPM. Uma das conclusões desta avaliação, mostra que o resultado de execução do BDV é equivalente à execução do BDSO, permanecendo ambos dentro do intervalo de confiança.

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

A seguinte análise tem como objetivo avaliar a utilização dos recursos, como CPU, memória, rede e energia. Comparando a utilização desses recursos, é possível argumentar os fatores que motivaram o comportamento sobre o desempenho dos cenários citados anteriormente.

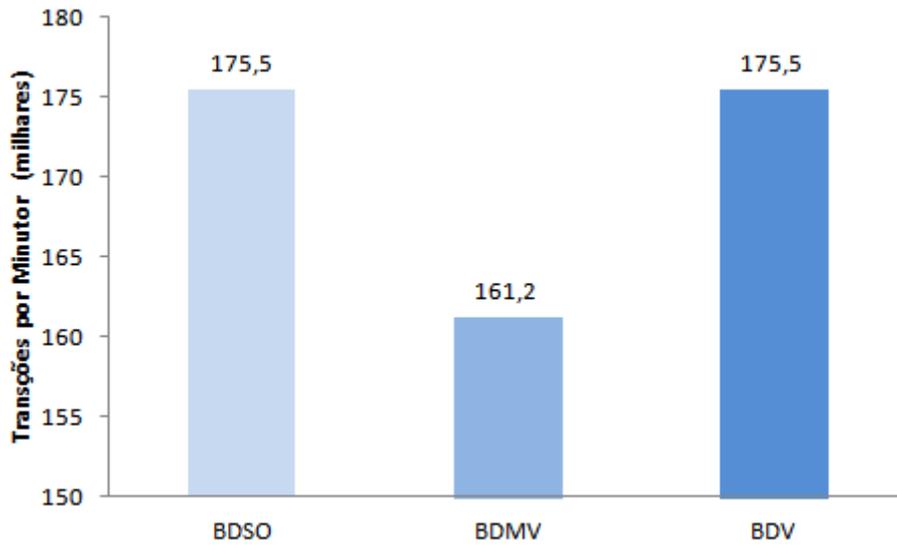


Figura 5.2 – Comparação de desempenho: transações executadas por minuto BDSO, BDMV e BDV

Comparação de Utilização de CPU

O resultado na Figura 5.3 representa um acréscimo de 39% de utilização de CPU do BDMV, comparado com BDV e o BDSO. Foi observado 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.

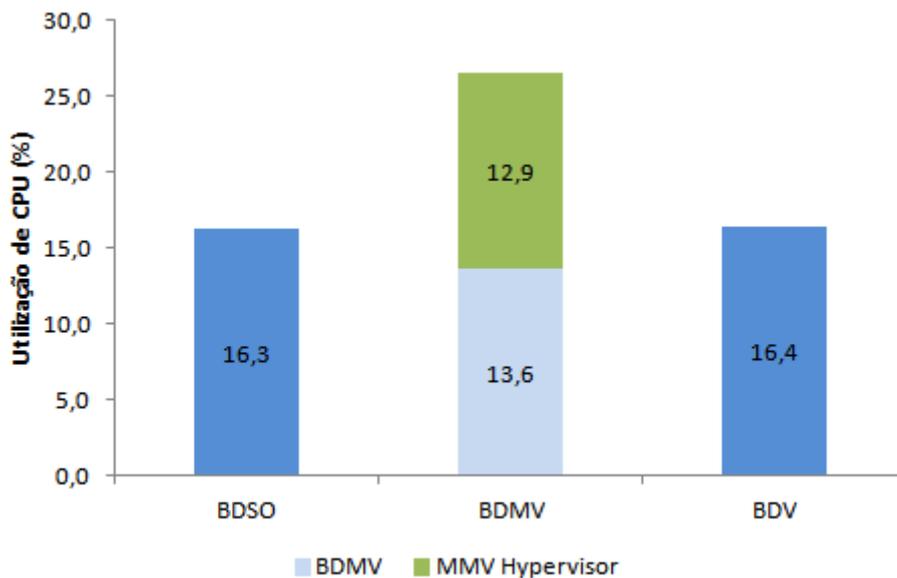


Figura 5.3 – Comparação de utilização de CPU: BDSO, BDMV e BDV

Comparando a execução dos três cenários, 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 BDMV é maior em relação à 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* [MYAS08].

Comparação de Consumo de Energia

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 todos os valores de consumo, registrados a cada segundo.

O esforço adicional do BDVM causado pela camada intermediária de virtualização pode ser observado na Figura 5.4 que apresenta o consumo de energia. O consumo de energia do BDSO e do BDV é inferior ao consumo do BDVM. A abordagem que utilizava MV teve uma média de consumo de 9% maior do que as demais abordagens. Este comportamento pode ser explicado e conforme observado na seção anterior, é devido a alta utilização massiva dos recursos de *hardware*, como por exemplo CPU.

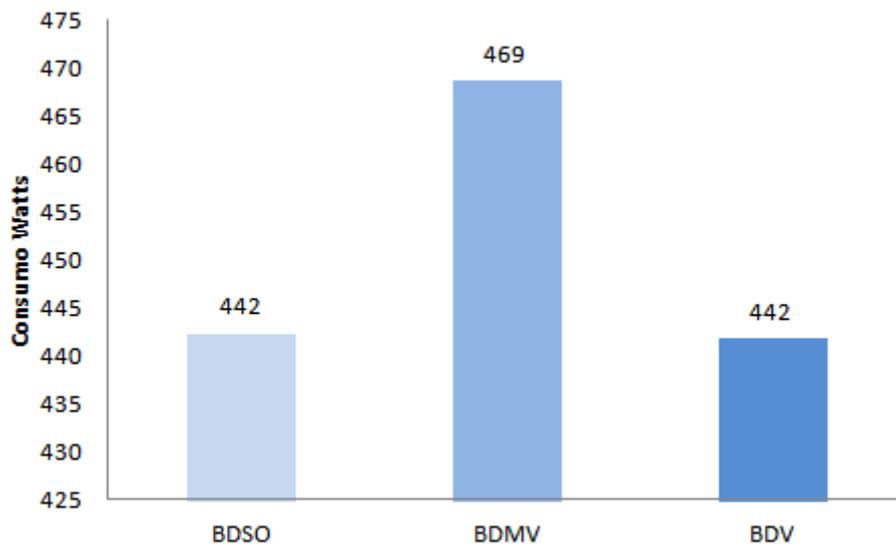


Figura 5.4 – Comparação do consumo de energia: BDSO, BDVM e BDV

Comparação de operações de entrada e saída

Outra análise realizada é mostrada na Figura 5.5 que compara a utilização de recursos como 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 BD simulando transações essencialmente de compra e venda com múltiplos usuários, ocasionando esse comportamento. O BDVM tem uma utilização de rede e de 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 BDMV devido à 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. É possível afirmar que o BDSO e o BDV são equivalentes, considerando o intervalo de confiança.

Esse resultado pode ser explicado considerando que o BDVM tem um maior esforço em traduzir instruções da MV para o *hardware*, impactando em uma sobrecarga na utilização de CPU, o que ocasiona a menor taxa de transferência de dados via rede ou de gravação no disco.

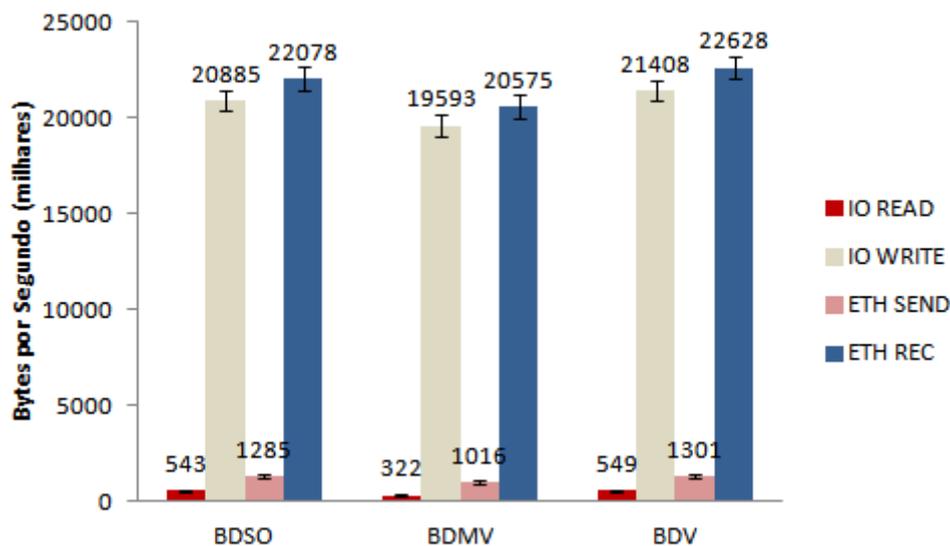


Figura 5.5 – Comparação de operações de entrada e saída (rede e disco): BDSO, BDMV e BDV

Comparação de Operações de Memória

Na Figura 5.6 são mostradas as operações de páginas executadas na memória. Esse monitoramento foi realizado na camada do SGBD, ou seja, sobre a máquina nativa e sobre a MV. São mostradas, a quantidade de Páginas Carregadas (*Page-In*), Páginas Descarregas (*Page-Out*) e Páginas não Encontradas (*Page-Fault*). Esta última também pode ser considerada um dos motivos responsáveis pelo baixo desempenho de uma aplicação sobre VM [MYAS08]. Pode-se notar que as operações de memória do BDMV são inferiores 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 seja inferior, se comparada ao BDV e BDSO, isso não pode ser considerado uma vantagem dessa abordagem, haja vista que o BDVM teve o desempenho inferior e o consumo de energia superior.

As análises apresentadas nesta seção possibilitam determinar que o BDSO e BDV tem um desempenho equivalente, sendo que este último ainda proporciona os benefícios que a virtualização

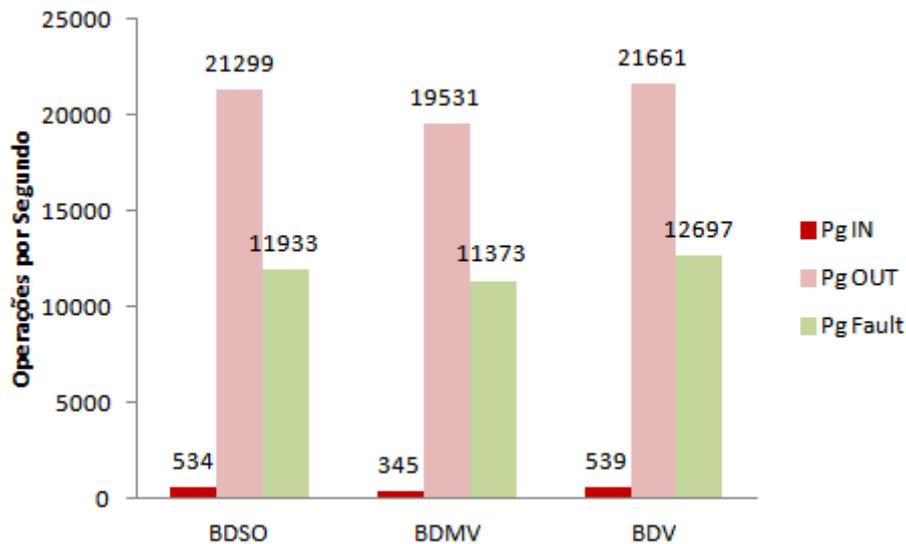


Figura 5.6 – Comparação de operações de memória (Page-In, Page-Out e Page-Fault): BDSO, BDMV e BDV

oferece. O BDVM tem um desempenho inferior entre 8 a 10% devido à inclusão da camada adicional de virtualização.

A Camada adicional de virtualização proporciona uma administração flexível do BD, porém, o aumento do consumo dos recursos é nítido e pode ser observado nas Figuras 5.3 e 5.4. O consumo por recursos de *hardware* refletem diretamente no consumo de energia. Além disso, a perda de desempenho é representada na Figura 5.2.

É possível determinar que a camada de *hypervisor* ocasiona a perda de desempenho devido principalmente a dois fatores: inclusão de uma camada intermediária na arquitetura [LCR⁺12] e devido ao alto índice de operações de páginas de memória, em especial pelas operações de páginas não encontradas (*Page-Faults*) [MYAS08].

5.2.2 Avaliação de gerência de recursos

Como mencionado na Seção 5.1.2, para avaliar a gerência de recursos foi analisada a capacidade de isolamento das instâncias do BDV. Foram executadas duas experiências para identificar os efeitos da alteração de recursos durante a execução do *benchmark* quais sejam, a primeira utilizado para identificar o tempo necessário para o ajuste à nova configuração de recursos, observando o impacto no desempenho causado pela alteração, e a segunda é baseada na observação da interferência de desempenho de uma instância de BD sobre outras.

A Figura 5.7 apresenta o resultado do primeiro teste: exhibe o comportamento durante o processo de reconfiguração de recursos do banco de dados para uma determinada instância, alterando de 16 CPUs para 8 CPU. Pode ser observado que a utilização da CPU é estável em 47% o que caracteriza 50% dos recursos disponíveis de CPU (32 CPU ao total), porém ao limitar os recursos de

CPU para 25% (8 CPUs), imediatamente o gráfico demonstra a queda no percentual de utilização, passando para 23% de utilização, identificado no gráfico.

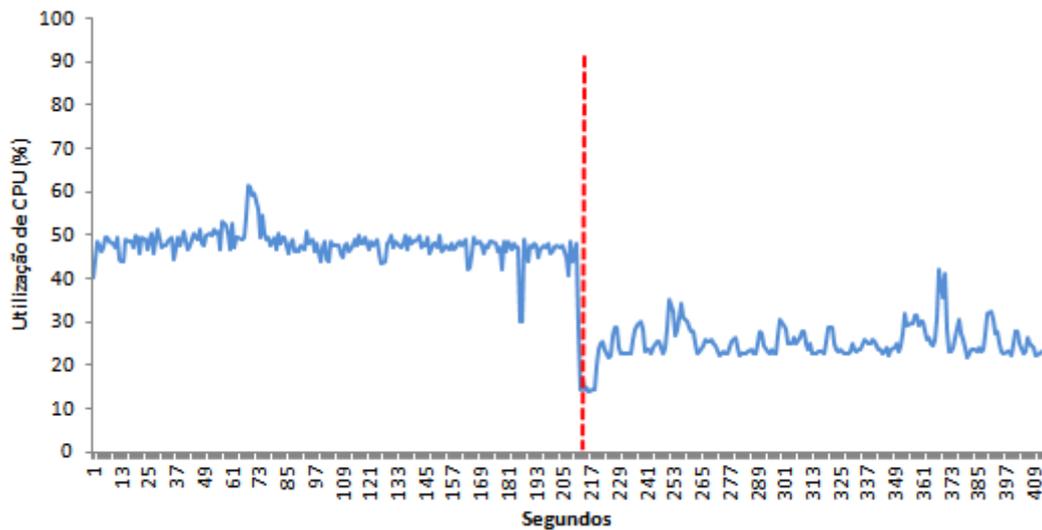


Figura 5.7 – Utilização de CPU: Alteração de 16 CPUs para 8 CPUs.

A Tabela 5.2 mostra a variação entre os períodos pré e pós alterações de recursos. É perceptível a variação após a alteração da quantidade de CPUs, bem como as operações de Leitura/Escrita diminuíram significativamente.

Esse comportamento sugere que é possível isolar os recursos utilizados por uma instância de banco de dados.

Tabela 5.2 – Utilização de E/S: Valores de Leitura e Escrita com diminuição de CPU

| Métrica | Leitura | Escrita |
|---------------------------|--------------|----------------|
| Execução com 16 CPUs | 765542 bytes | 17513242 bytes |
| Execução com 08 CPUs | 648628 bytes | 10954980 bytes |
| Média de bytes | 707085 bytes | 14234111 bytes |
| Variação (Antes - Depois) | 116914 bytes | 6558262 bytes |

O segundo experimento teve como objetivo validar se o isolamento do BDV evita a interferência da utilização de recursos durante a execução de uma instância de BD sobre os recursos de outras instâncias de BD. Para isso, comparou-se a execução de 4 instâncias de BD, cada uma com 4 núcleos de CPU, e durante a execução de referência foi alterada a quantidade de CPU de uma instância de BD, de 4 para 16 núcleos.

A Figura 5.8 mostra o resultado de desempenho de execução simultânea das 4 instâncias de BD. Inicialmente, todas as instâncias foram configuradas com 5% dos recursos de CPU disponíveis. A Figura 5.8(a) demonstra que todas as instâncias executam a mesma quantidade de transações. Durante a execução, a Instância 04 teve a sua quantidade CPUs alterada de 4 a 16 (de 5% para 50%).

Pode ser verificado que, após a alteração, o volume de transações da Instância 04 aumentou de 43532 para 88382 TPM (Figura 5.8(b)).

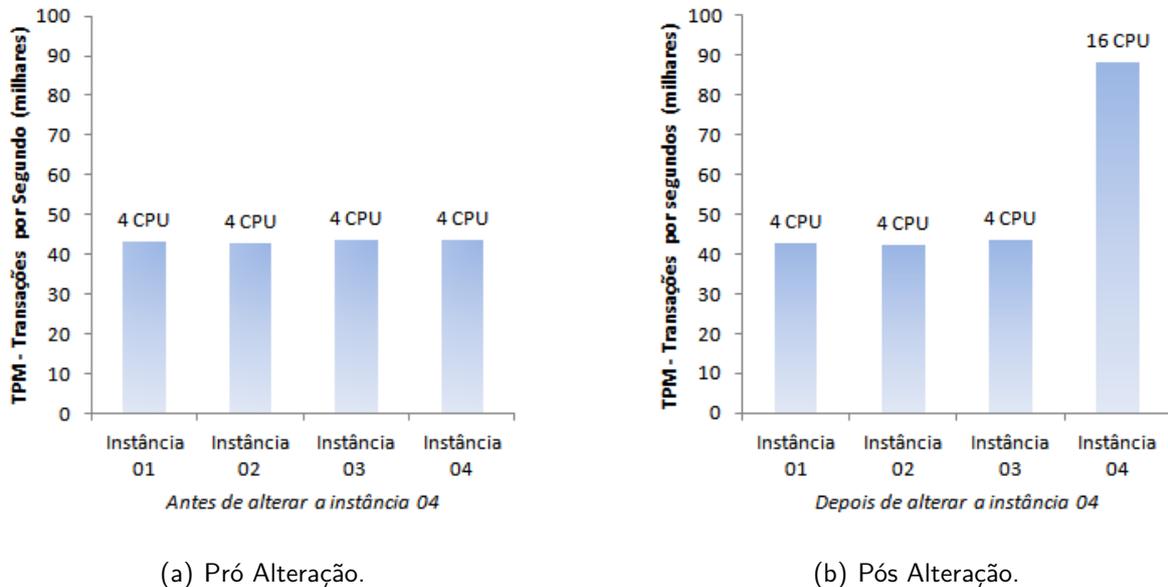


Figura 5.8 – Desempenho TPM: alteração de 4 CPUs para 16 CPUs.

Com o aumento dos recursos, neste caso, a quantidade de CPUs da instância 04, houve uma maior eficiência, sem interferir nas outras instâncias de BD, que continuaram com o mesmo comportamento (Tabela 5.3). Podemos, então, afirmar que através do isolamento há reserva de recursos para a execução do BDV.

Tabela 5.3 – Resultado de TPM com Aumento de Recursos à Instância 04.

| Métrica | Instância 01 | Instância 02 | Instância 03 | Instância 04 |
|---------------------------------|--------------|--------------|--------------|--------------|
| Antes do Aumento | 43203 | 42735 | 44440 | 43532 |
| Após o Aumento | 42651 | 42184 | 43440 | 88382 |
| Média de Transações por segundo | 676 | 668 | 682 | 1038 |
| Valor Máximo de Transações | 66156 | 61536 | 66588 | 105084 |
| Consumo de Memória | 8 Gb | 8 Gb | 8 Gb | 8 Gb |

5.2.3 Avaliação de alta disponibilidade

Para avaliar a alta disponibilidade foi analisado o processo de migração de BD. A primeira avaliação baseia-se na execução de benchmark TPC-C durante a execução desse processo. A Figura 5.9 representa a 10 *warehouses* com 200 conexões simultâneas. Pode ser observado que o desempenho do BDV é de 15% maior em relação ao desempenho do BDMV.

Outra análise possível de se realizar (Figura 5.9) é o tempo gasto no processo de migração do banco de dados. O BDV levou em média 126 segundos (representado por linhas vermelhas),

enquanto que o BDVM levou em média de 540 segundos (representado pelas linhas Verdes). Esse comportamento pode ser explicado, conforme visto na Seção 4.2.1 pois os aplicativos de banco de dados realizam modificações constantes nas páginas de memória do servidor e isto implica na necessidade de realizar a cópia das páginas de memória modificadas várias vezes durante o processo de migração da VM. Por outro lado, a Seção 4.2.1 apresenta que o BDV migra apenas os processos de sessão e somente após a finalização da execução da transação.

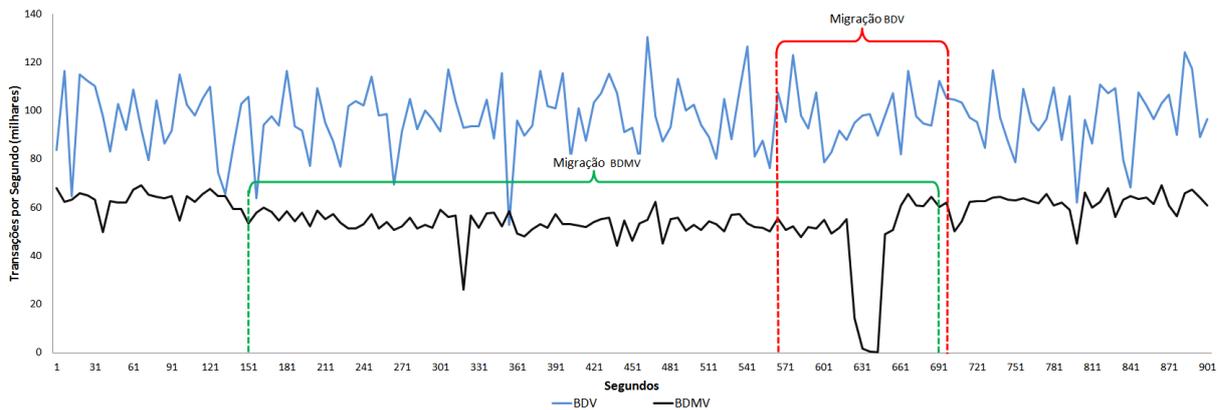


Figura 5.9 – Comparação das transações executadas por minuto durante a migração: BDV e BDMV.

Também é perceptível que durante o processo de migração BDV o desempenho pode ser considerado estável, sem perda significativa de transações. Por outro lado, foi observado que as operações BDMV pararam de responder por curto período de tempo, em média 24 segundos. Esta parada do BDMV é necessária para finalizar a cópia das páginas de memória, conforme mencionado na Seção 4.2.1, o que também foi observado em outros trabalhos [VBVB09]. No entanto, essa parada do BDMV não foi perceptível ao ponto de ser detectado qualquer tipo de inatividade por demora de resposta.

Na Tabela 5.4 é apresentado os valores médios de transações executadas sem o processo de migração e com o processo de migração do BDMV e do BDV. Analisando esses dados, é possível sugerir que o desempenho das execuções do BDV são mantidas durante todo o processo de migração, o que não ocorre durante a migração do BDVM.

Tabela 5.4 – Comparação de Desempenho Durante a Migração: BDVM e BDV.

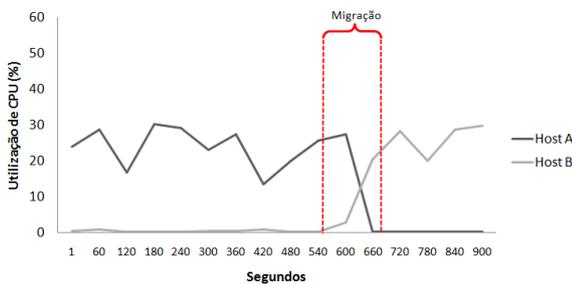
| Métrica | BDMV | BDV |
|--|--------------------|--------------------|
| Media | 55890.00 | 97007.02 |
| Média durante Migração | 51234.56 | 96764.43 |
| Média sem a Migração | 62049.05 | 96843.28 |
| Perda de Desempenho Durante a Migração | 17.42% | 0.081% |
| Intervalo de Confiança | 54170.15, 57609.45 | 94844.04, 99170.00 |

As análises de recursos que se seguem é mais um passo em direção ao nosso objetivo de avaliar o processo de migração.

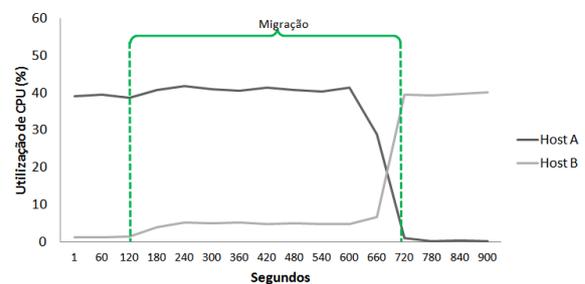
Utilização de CPU Durante Migração

Para monitorar a utilização de CPU durante do processo de migração foi utilizada a ferramenta *mpstat*. Este utilitário é um software de linha de comando utilizado em sistemas operacionais do tipo *Unix* para extrair estatísticas relacionadas ao processamento. É comumente utilizada no monitoramento de computadores, a fim de diagnosticar sobrecargas ou para gerar estatísticas sobre o uso da CPU do computador.

Pode ser observado na Figura 5.10 que o BDV tem a utilização da CPU inferior de em 12,9% em comparação com ao ambiente DBVM com o mesmo *workload*. O ambiente DBVM, identificado na Figura 5.10(b) tem um acréscimo da utilização de CPU no *host* de destino logo no início da migração, porém a utilização de CPU passa ao mesmo nível do *host* de origem somente ao concluir a migração. Conforme observado na Seção anterior, o tempo médio de migração do BDVM foi de 540 segundos, período este em que os dois *hosts* terão uma sobrecarga de utilização de CPU. Já o ambiente do BDV, identificado na Figura 5.10(a), leva 126 segundos sendo que o *host* de destino inicia imediatamente o processamento, reduzindo a utilização de CPU no *host* de origem proporcionalmente.



(a) Utilização de CPU BDV.



(b) Utilização de CPU BDVM.

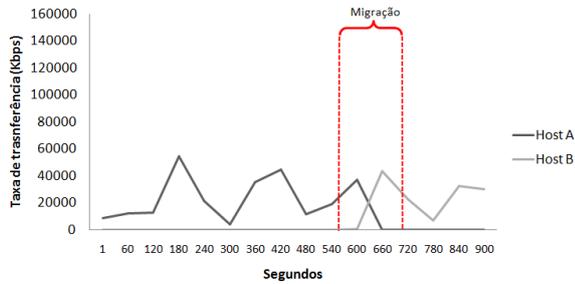
Figura 5.10 – Comparação da utilização de CPU durante a migração: BDV e BDMV.

Utilização de Rede Durante a Migração

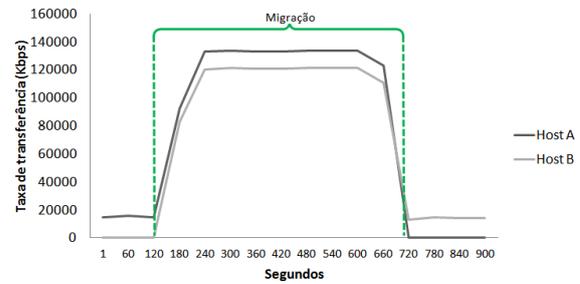
A taxa de utilização da rede é o recurso que exhibe a maior diferença no uso durante o processo de migração, o qual pode ser observado na Figura 5.11. É possível verificar que a rede do BDV, Figura 5.11(a), não sofre um aumento perceptível no uso durante o período de migração, o que permite sugerir que não há nenhuma mudança em termos de consumo. Uma possível explicação para este comportamento pode ser que, considerando que o BDV transfere as sessões somente após a execução da transação, torna-se desnecessário copiar os dados durante a execução.

O ambiente DBVM, Figura 5.11(b), sofre um aumento considerável na utilização ao longo de todo o processo de migração. A transferência dos dados antes e após o processo de migração

é semelhante à taxa de utilização da rede de BDV, no entanto, observa-se que o BDMV tem uma sobrecarga durante este período em virtude da cópia VM, que tem o seu próprio SO e SGBD instalado e que são copiados a partir de um hospedeiro para outro. O volume de dados transmitidos a partir de um hospedeiro para outro está relacionado com o tamanho da MV, isto é, quanto mais recursos são atribuídos à MV, maior será o período e maior será a taxa de dados transferidos durante o processo de migração.



(a) Utilização de Rede BDV.



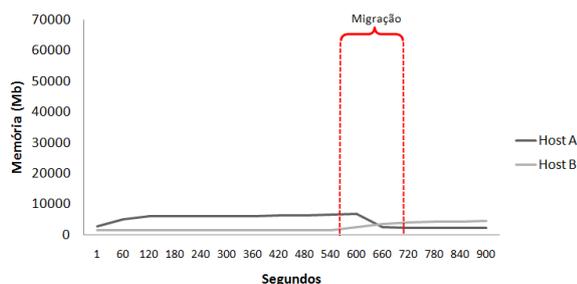
(b) Utilização de Rede BDMV.

Figura 5.11 – Comparação da utilização de CPU durante a migração: BDV e BDMV.

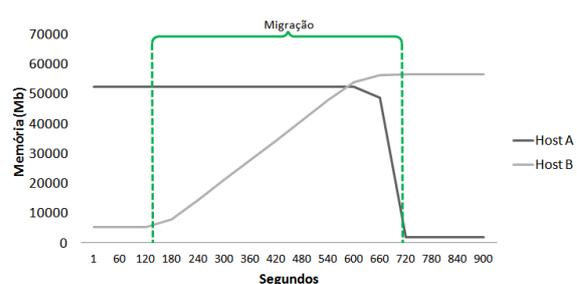
Utilização de Memória Durante a Migração

Pode ser observado na Figura 5.12 que o uso da memória do BDMV (média 59Gb) em comparação com o uso de memória do BDV (média de 6.2Gb) chega a nove vezes a mais. Isto se dá devido ao fato de que a MV (BDMV) utiliza toda a memória reservada pelo *hypervisor*, enquanto que o BDV utiliza somente a memória consumida pela SO e pela instância de BD.

Considerando que a VM (BDMV) utiliza todos os recursos reservados e que o *hardware* tem 64GB de RAM, esta VM pode ocupar os 64Gb de memória disponível, observado 5.12(b), enquanto que o BDV, observado na Figura 5.12(a) tem sua memória utilizada principalmente pela instância de banco de dados, utilizando de forma dinâmica, conforme necessidade de mais recursos. Este comportamento pode ajudar a entender as razões da elevada taxa de transferência de dados, considerando que MVs menores (com menos recursos alocados) demandam menos tempo e taxa de transferência, o que não foi observado no BDV.



(a) Utilização de Memória BDV.



(b) Utilização de Memória BDMV.

Figura 5.12 – Comparação da utilização de memória durante a migração: BDV e BDMV.

A Tabela 5.5 permite analisar todos os dados obtidos durante os testes realizados no processo de migração.

É possível destacar que a migração do BDV é realizada num período de tempo mais curto, com um valor menor operações de entrada e saída e de consumo de memória. É perceptível também que um DBV não perde desempenho durante este período e que o tamanho do banco de dados não afeta os parâmetros observados para a BDV. Esta avaliação complementa os trabalhos anteriores, onde foram realizadas avaliações de migração apenas de banco de dados sobre MV.

Tabela 5.5 – Comparação do processo de migração: BDMV e BDV.

| Métricas (média) | DBVM | | BDV | |
|------------------------|--------------|--------------|--------------|--------------|
| | Sem Migração | Com Migração | Sem Migração | Com Migração |
| Transações por minutos | 65.789 | 48.690 | 97007 | 96.891 |
| Utilização rede | 5% | 98% | 4% | 6% |
| Utilização memória | 96% | 97% | 12% | 12% |
| Utilização CPU | 39% | 45% | 23% | 24% |

5.3 Matriz de Comparação

Uma Matriz de Comparação (MC) pode ser considerada uma forma simples de selecionar a melhor alternativa para determinar se o requisito é apresentado no modelo desejado. Assim, é colocado em uma tabela, de um lado, os requisitos funcionais, por exemplo, ao longo da primeira linha listamos os modelos de implementação de banco de dados estudadas e de outro lado, na primeira coluna, as características analisadas.

Os demais espaços da tabela são utilizados para indicar se cada uma das alternativas apresenta ou não a característica desejada. Desta forma, cabe ao administrador de TI identificar o que é mais importante para ele, selecionando as características que ele deseja.

Os requisitos que fazem parte desta MC foram identificados e avaliados neste trabalho, sendo classificados conforme segue:

1. **Desempenho:** identifica a abordagem que tem o desempenho nativo ao ambiente, ou seja, mostra a implementação cuja execução do SGBD pode ser diretamente sobre o *hardware* ou com poucas camadas intermediárias que prejudiquem o desempenho, sendo identificado como alto para o desempenho nativo (sem a tradução das instruções) e baixo para o desempenho que sofre degradação.
2. **Gerência de recursos:** mostra quais abordagens oferecem a possibilidade de gerenciar os recursos de *hardware* para os BD em execução. Estes recursos podem ser quantidade de CPU ou memória alocada para cada BD, conservando tais recursos dedicados e evitando a interferência de utilização e preservando a segurança entre as instâncias virtualizadas.

3. **Alta disponibilidade:** representa as abordagens que implementam recursos para alta disponibilidade do serviço de BD. Neste contexto, este requisito é dado pela funcionalidade de migração em tempo real do BD de dados, com a mínima interrupção do serviço e com pouca ou nenhuma perda de performance.
4. **Fácil Administração do ambiente:** indica as abordagens que possibilitam a configuração do ambiente com o mínimo esforço de instalação.
5. **Multiplataforma:** identificam as abordagens que tem a possibilidade de ser instalado em um ambiente multiplataforma, como por exemplo: sobre SO Windows ou sobre Linux.

Dados os requisitos que compõem a MC, esta é representada abaixo:

Tabela 5.6 – Matriz de Comparação

| Requisito | BDSO | BDVM | BDV |
|---------------------------------|------|-------|------|
| Desempenho | Alto | Baixo | Alto |
| Gerência de Recursos | Não | Sim | Sim |
| Alta disponibilidade | Não | Sim | Sim |
| Fácil Administração do ambiente | Sim | Não | Não |
| Multiplataforma (SO) | Sim | Sim | Não |

De posse desta MC é possível agora realizar um estudo de caso, identificando as características necessárias à implementação da solução e, posteriormente comparar com o ambiente atual. O resultado da utilização desta MC pode ser observada na próxima seção.

5.4 Estudo de caso

O estudo de caso foi realizado em um ambiente real de execução de BD, onde foram identificadas as características desejáveis para obter o melhor desempenho deste ambiente através da análise da MC.

Este ambiente é composto por mais de 10 milhões de tuplas, organizadas em tabelas relacionais de dados. Sua arquitetura é composta de uma MV devido ao fato de serem necessárias funcionalidades como a alta disponibilidade e a gerência de recursos. Um dos desafios que o administrador deste banco de dados enfrenta é o problema da perda de desempenho, devido ao fato de que o BD é executado sobre uma MV. Porém, a MV é fundamental para garantir as funcionalidades acima relacionadas.

A estrutura do BD pode ser visualizada na Figura 5.13. É possível notar que não existem chaves entre a tabela de dados Posicionamento e as tabelas Propriedades e PO. Na tabela Propriedades a relação ocorre pela utilização do Varray na tabela posicionamento, conforme mostrado na Entidade de Relacionamento (ER) (Figura 5.13).

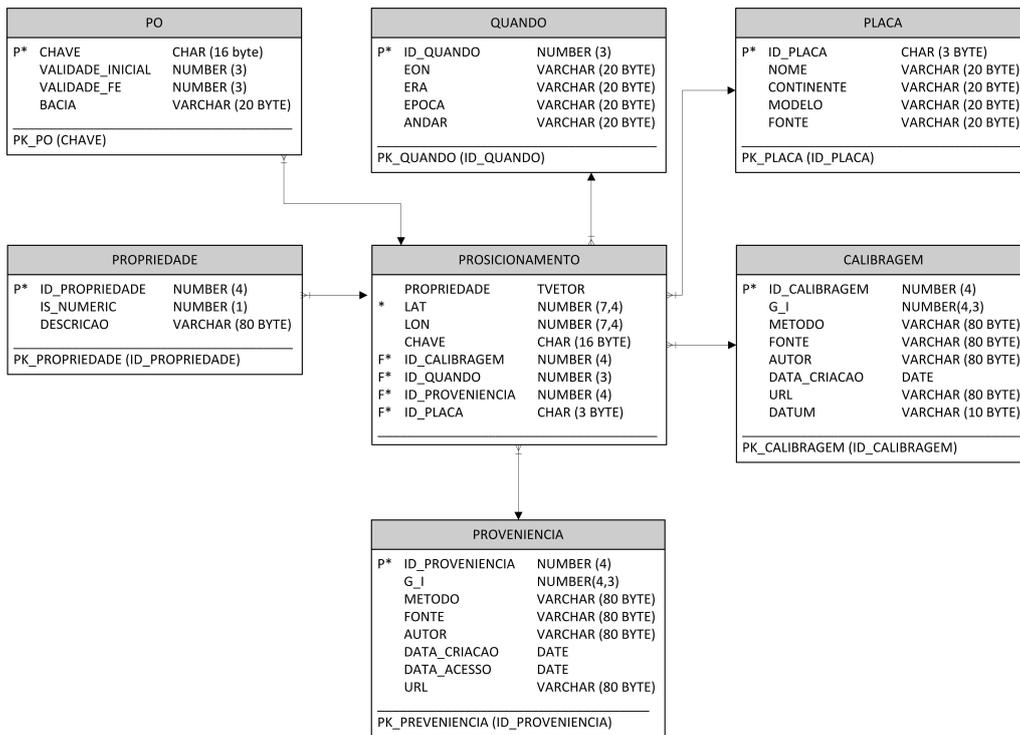


Figura 5.13 – ER do BD utilizado no estudo de caso.

Um dos principais problemas do ambiente atual é o *overhead* causado pela execução de transações com alto índice de processamento. Dentre as principais transações que tem uma utilização massiva de *hardware* é possível destacar a *query* apresentada a seguir:

```
select
  p.chave, p.lat, p.lon, p.id_quando, p2.*
from
  posicionamento p, TABLE(p.propriedades) p2
where
  p2.batimetria < -5000;
```

Para realizar o estudo de caso, o primeiro passo para obter um melhor desempenho na execução das transações é a identificação dos requisitos necessários de BD. Faz-se necessária uma análise das premissas que são essenciais para manter o ambiente de BD para garantir o melhor desempenho, mantendo as características já existentes. Foram identificadas as seguintes características conforme segue:

- Desempenho:** neste ambiente de BD o desempenho é fundamental. O processamento realizado é intenso e exige uma grande quantidade de leitura de dados sendo pesquisando em várias tabelas e acessando vários índices de dados.
- Gerência de recursos:** considerando que este tipo de banco de dados executa múltiplas pesquisas simultaneamente e com vários BDs, a gerência de recursos pode se tornar um ponto crítico, exigindo o fornecimento de um mecanismo que forneça um forte isolamento entre as

execuções das transações, possibilitando a alteração dos parâmetros de recursos durante a execução.

3. **Alta disponibilidade:** Neste caso, o ambiente deve garantir a alta disponibilidade para a execução do BD de forma ininterrupta.
4. **Implementação:** Todos os recursos de *hardware* e *software* necessários para a implementação das abordagens BDSO, BDVM e BDV estão disponíveis e podem ser utilizados.
5. **Multiplataforma:** Atualmente o ambiente está sendo executado em uma plataforma homogênea, utilizando SO e SGBD suportados nas abordagens BDSO, BDMV e BDV.

Considerando as necessidades acima mencionadas e analisando a MC dos requisitos listados acima, a melhor abordagem para implementar o serviço de BD e garantir um desempenho eficiente com o gerenciamento de recursos e alta disponibilidade, sem dúvida, é a abordagem de Banco de Dados Virtual (BDV).

Desta forma foi configurado o BDV garantindo a fiel replicação da estrutura atual em execução. Para garantir a replicação dos dados foi realizada a cópia física dos arquivos de dados e inicializada o novo BD, mantendo os atuais parâmetros de execução. Também foi utilizado o mesmo *hardware*, estrutura de *storage* e rede para manter um resultado confiável.

Para comparar o desempenho foi executada a mesma *query* nos dois ambientes, observando o tempo de retorno de cada uma. Para garantir um resultado imparcial para cada execução a sessão era fechada e para novas execuções, era inicializada uma nova sessão, evitando assim que controles de acesso permanecessem na memória, priorizando certas execuções.

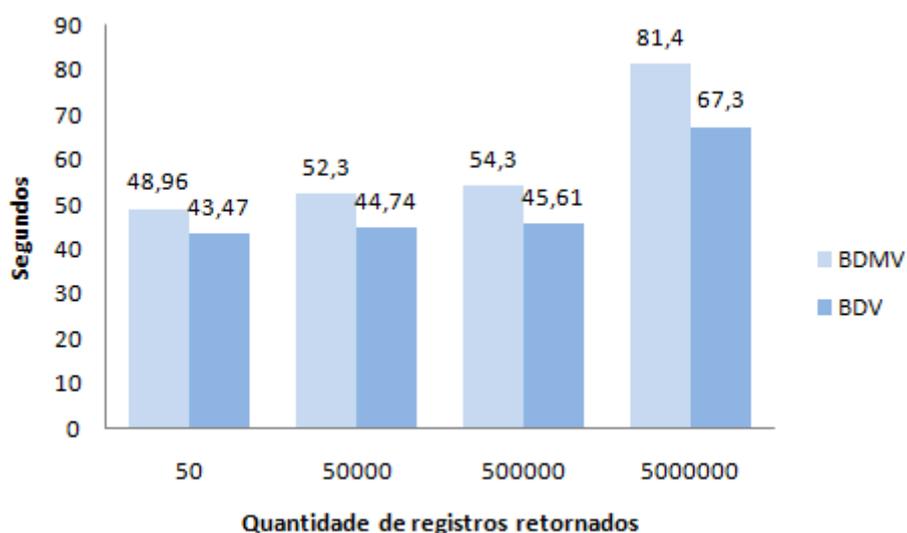


Figura 5.14 – Estudo de caso: comparação do desempenho do ambiente atual (BDMV) e BDV

Para garantir o desvio padrão de 95%, foram executadas 7 vezes cada *query* em cada ambiente, sendo calculado o valor médio do tempo de execução. A Figura 5.14 mostra o resultado obtido da execução da *query* de pesquisa realizada no ambiente atual (BDMV) e no BDV.

Observando a Figura 5.14 é possível notar o ganho de desempenho alcançado com a implementação do BDV. Este ganho foi de 12% para a execução com 50 tuplas de resultado. Para 50000, 500000 e 5000000 tuplas, o ganho passou para 15%, 17% e 18% respectivamente, mantendo estável em 18% para resultados maiores.

Observa-se que o ganho de desenho do BDV em relação ao ambiente atual é crescente, dado o número de tuplas retornadas. Este comportamento pode ser explicado pelos seguintes fatores:

Desempenho: o BDV utiliza os recursos de *hardware* em torno de 30% a menos.

Entrada e Saída (rede e disco): a taxa de transferência dos dados de rede e disco do BDV é menor em 8%, isso possibilitou ao ambiente BDV processar um maior número de registros, comparado ao ambiente atual.

Operações de memória: o BDV executou mais operações de memória, garantindo assim o melhor resultado. Considerando que o total de *Page-Faults* do BDV é inferior ao ambiente atual, isso é refletido no desempenho.

Gerência de recursos e alta disponibilidade: os dois ambiente são providos destes recursos e conforme demonstrado neste trabalho, durante o processo de migração em tempo real do BDV, este não sofre perda de desempenho.

Isolamento: Os dois ambientes são providos de isolamento porém este recurso no ambiente atual é gerenciado pela MV, enquanto que no BDV este recurso é assegurado pela execução dos processos utilizando a tecnologia de *containers*.

Este estudo de caso complementa este trabalho, onde foram realizadas avaliações de desempenho, gerência de recursos e alta disponibilidade. Desta forma, é possível relacionar os dados obtidos na avaliação das abordagens tradicionais de virtualização, sugerindo como uma alternativa a utilização do BDV, baseada na utilização da MC gerada a partir dos estudos realizados neste trabalho.

5.5 Conclusões deste capítulo

Este Capítulo é complementar a outros trabalhos disponíveis na literatura que comparam a abordagem de virtualização de sistemas de banco de dados [AAS08] [CJMB11] [MYAS08]. Nos estudos realizados, a utilização do BDV apresenta um melhor desempenho se comparado com a execução do BDMV, principalmente pelo fato de que o *hypervisor* da MV exige a tradução de cada instrução que está sendo executada. Porém existem outros fatores que também podem influenciar no desempenho, tais como, o custo de migração e a possibilidade de consolidação de servidores. Embora cada solução possua seus benefícios e limitações um problema que é compartilhado por todas é o desempenho.

A abordagem BDV pode ser considerada a nova forma de virtualizar um sistema de banco e dados. Neste tipo de implementação, somente a instância do banco de dados é virtualizada mediante o compartilhamento de recursos pelo *hypervisor* a nível de *kernel*. O gerenciamento de

recursos pode ser individual por instância de banco de dados. Um dos benefícios de um cenário virtualizado é que múltiplas aplicações de banco de dados podem ser executadas simultaneamente em um servidor com isolamento total.

6. CONCLUSÃO

Este trabalho teve como objetivo estudar e avaliar o desempenho de um sistema de banco de dados virtual e comparar com as abordagens que um ambiente de TI tradicionalmente utiliza: abordagem sem a camada de virtualização, instalado diretamente sobre o *hardware* nativo ou utilizando a tradicional abordagem de virtualização, que faz uso de MV.

Para alcançar tais objetivos, primeiramente procurou-se na literatura por conceitos e abordagens que auxiliassem na satisfação desses desafios ainda não alcançados. Em seguida, foram elencados os pilares de comparação e estudo, quais sejam: **Desempenho, Gerência de Recursos e Alta Disponibilidade.**

O próximo passo foi identificar as abordagens para estudar e satisfazer tais requisitos comuns em todas as implementações, possibilitando alcançar os objetivos definidos neste trabalho.

Além disso, foi ofertado um estudo de caso de um ambiente real, utilizando o conceito de banco de dados virtual. Este estudo de caso utilizou a Matriz de Comparação gerada a partir dos dados levantados por este trabalho.

Desta maneira, as seguintes questões de pesquisa foram respondidas:

1. **Existe uma alternativa à área de banco de dados para diminuir o *overhead* causado pela inclusão da camada adicional de virtualização, porém mantendo os benefícios trazidos pela mesma?**

Tendo a necessidade de inclusão de uma camada adicional de virtualização através de utilização de MV e a perda de desempenho comprovado nos estudos apresentados, surge como uma alternativa à abordagem tradicional de virtualização à área de banco de dados a utilização de BDV, implementado através do conceitos de *containers*, onde cada instância de banco de dados pode ser administrada como um ambiente virtual, porém com uma desempenho semelhante ao ambiente nativo.

2. **Qual é a eficiência da abordagem de banco de dados virtual?**

A eficiência da utilização da abordagem de banco de dados virtual pode ser avaliada nos seguintes aspectos:

Desempenho: Foi possível identificar e comprovar que o desempenho do número de transações do BDV é superior em 9% à abordagem que utiliza MV. Além disso, foi possível observar que a utilização de recursos, tais como memória, discos e rede, foram menores devido a baixa utilização do *hardware*, ocasionando também uma redução do consumo de energia.

Gerência de Recursos: Neste quesito, o BDV é tão eficiente quanto a abordagem que utiliza uma MV, oferecendo funções semelhantes, tais como: migração ao vivo, isolamento de recursos e reconfiguração dos parâmetros de execução.

Alta Disponibilidade: O BDV é tão eficiente quanto a utilização de MVs, garantindo a alta disponibilidade.

3. **Qual é o custo do processo de migração utilizada na consolidação das abordagens tratadas neste trabalho?**

O custo do processo de migração apresentado na Seção 5.5 mostra que a migração de uma instância de banco de dados virtual é mais eficiente do que a migração de uma MV com um BD. Este trabalho, evidenciou que a migração de uma MV é onerosa devido ao transporte da MV, do SO e do BD, sendo que o mesmo transporte do BDV é mais eficiente seja em requisito de tempo, transferência de dados e disponibilidade do serviço de BD.

4. **É possível executar múltiplas instâncias de banco de dados virtuais em um mesmo servidor físico sem que uma prejudique o desempenho da outra, garantindo o isolamento durante a execução?**

Ambientes virtuais normalmente executam múltiplas instâncias de MVs garantindo o total isolamento entre as MV. Avaliando o BDV foi possível observar que as instâncias de banco e dados virtuais são executadas de forma isolada uma das outras. A execução de uma instância não afeta a execução de outras, garantindo assim um desempenho estável, seguro e com recursos reservados de *hardware* para cada uma delas.

6.1 Resumo das Contribuições

Um das contribuições que este trabalho apresenta é que a utilização de banco de dados virtual como recurso de TI mantém os benefícios da virtualização sem a perda de desempenho, ocasionado pela inclusão de uma camada adicional de virtualização. A execução do sistema de banco de dados virtual (BDV) é tão eficiente quanto a execução diretamente no sistema operacional, e ainda oferece todas as vantagens da virtualização. O BDMV utiliza mais recursos de *hardware*, impactando também no consumo de energia e ocasionando um custo mais elevado.

Além da contribuição citada acima, outras contribuições específicas deste trabalho são:

- **Introduzir o conceito de virtualização de banco de dados utilizando a tecnologia de *containers*:** Este trabalho propiciou a inclusão de uma nova alternativa à abordagem tradicional de virtualização de banco de dados, abrindo uma discussão para futuras pesquisas referente a utilização de banco de dados virtuais através da tecnologia de *containers*.
- **Teste, avaliação e comparação de particularidades de funcionamento tais como valores de desempenho, consumo de recursos e características de gerência das abordagens mais comuns, comparando com abordagem introduzida neste trabalho:** Trabalhos anteriores avaliaram o desempenho de BD sobre MV e compararam com a execução nativa. Este trabalho torna-se complementar aos demais, pois não só compara métricas de desempenho e utilização de recursos, mas também avalia uma nova alternativa, comparando com alternativas já existentes.

- **Desenvolvimento de uma matriz de comparação para identificar o modelo adequado a ser aplicado em um ambiente de TI, variando as necessidades para cada implementação:** Dadas as necessidades individuais de cada ambiente e do administrador do banco de dados, a matriz de comparação pode ser utilizada para auxiliar tais gestores a implementar a melhor alternativa baseada em uma matriz construída mediante a análise de resultados obtidos durante a realização deste trabalho.
- **Aplicação de um estudo de caso de um ambiente real, aplicando a matriz de comparação e observando o desempenho do ambiente real:** Dado o estudo realizado, foi possível aplicar um estudo de caso no que comprovou-se os resultados identificados neste trabalho, alcançados através da aplicação da MC.

6.2 Trabalhos Futuros

Considerando os benefícios proporcionados neste trabalho, os seguintes tópicos podem ser candidatos de um estudo em maior profundidade e apresentados em trabalhos futuros:

- Elaboração de um algoritmo de consolidação e gerenciamento dos bancos virtuais, possibilitando assim que o próprio banco de dados virtual se adapte às necessidades das aplicações nos ambientes de nuvem computacional, assegurando desta forma as características que as nuvens computacionais oferecem, tais como elasticidade e reconfigurações com a mínima interação do provedor de serviço.
- Considerando a constante evolução da tecnologia de banco de dados, como trabalho futuro torna-se interessante a realização de novas pesquisas para identificar outras tecnologias de BDV que possam vir a fazer parte deste tipo de implementação em ambientes *heterogêneos* de SO.
- Criar mecanismos de integração com soluções para ambiente de nuvem, a fim de proporcionar soluções para demandas integradas de ferramentas de administração tais como, Openstack, Eucalyptus, OpenNebula e Nimbus.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AAS08] Aboulnaga, A.; Amza, C.; Salem, K. "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, 2008, pp. 746–747.
- [AM10] Ahmadi, M.; Maleki, D. "Performance evaluation of server virtualization in data center applications". In: Telecommunications (IST), 2010 5th International Symposium on, 2010, pp. 638–644.
- [AMN06] Apparao, P.; Makineni, S.; Newell, D. "Characterization of network processing overheads in xen". In: Virtualization Technology in Distributed Computing, 2006. VTDC 2006. First International Workshop on, 2006, pp. 2–2.
- [AZR⁺12] Almari, F.; Zavorsky, P.; Ruhl, R.; Lindskog, D.; Aljaedi, A. "Performance analysis of oracle database in virtual environments." In: AINA Workshops, Barolli, L.; Enokido, T.; Xhafa, F.; Takizawa, M. (Editores), 2012, pp. 1238–1245.
- [BDF⁺03] Barham, P.; Dragovic, B.; Fraser, K.; Hand, S.; Harris, T.; Ho, A.; Neugebauer, R.; Pratt, I.; Warfield, A. "Xen and the art of virtualization". In: Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003, pp. 164–177.
- [Car08] Carissimi, A. "Virtualização: da teoria a soluções". In: Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRCt2008,, 2008, pp. 173–207.
- [CJ06] Campbell, S.; Jeronimo, M. "Applied Virtualization Technology". Intel Press, 2006.
- [CJMB11] Curino, C.; Jones, E. P.; Madden, S.; Balakrishnan, H. "Workload-aware database monitoring and consolidation". In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, 2011, pp. 313–324.
- [CRFD07] Calheiros, R. N.; Rodrigues, G.; Ferreto, T.; De Rose, C. A. F. "Avaliando o ambiente de virtualização xen utilizando aplicações de bancos de dados". In: VIII Workshop em Sistemas Computacionais de Alto Desempenho (WSCAD'07), 2007, pp. 171–178.
- [Dat04] Date, C. "Introdução a sistemas de bancos de dados". Ed. Campus, 2004.
- [EN11] Elmasri, R.; Navathe, S. "Sistemas de banco de dados". Ed. Pearson Addison Wesley, 2011.
- [Fre12] FresBSD. "Freebsd jails". [Online; acessado em 19 de julho de 2012], Capturado em: <http://www.freebsd.org>, 2012.
- [ham13] "Hammerora". [Online; acessado em 19 de janeiro de 2013], Capturado em: <http://hammerora.sourceforge.net>, 2013.

- [HLAP06] Huang, W.; Liu, J.; Abali, B.; Panda, D. K. "A case for high performance computing with virtual machines". In: Proceedings of the 20th annual international conference on Supercomputing, 2006, pp. 125–134.
- [Jon06] Jones, M. T. "Virtual linux an overview of virtualization methods, architectures, and implementations", 2006.
- [LCR⁺12] Lange, T. A. P.; Cemim, P.; Rossi, F. D.; Xavier, M. G.; Belle, R. L.; Ferreto, T. C.; Rose, C. A. F. "Performance evaluation of virtualization technologies for databases in hpc environments". In: XIII Simpósio em Sistemas Computacionais (WSCAD-SSC), 2012, pp. 88–89.
- [Lin12] "Lxc linux containers". [Online; acessado em 19 de julho de 2012], Capturado em: <http://lxc.sourceforge.net>, 2012.
- [LPD⁺11] Lange, J. R.; Pedretti, K.; Dinda, P.; Bridges, P. G.; Bae, C.; Soltero, P.; Merritt, A. "Minimal-overhead virtualization of a large scale supercomputer". In: Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, 2011, pp. 169–180.
- [MST⁺05] Menon, A.; Santos, J. R.; Turner, Y.; Janakiraman, G. J.; Zwaenepoel, W. "Diagnosing performance overheads in the xen virtual machine environment". In: Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments, 2005, pp. 13–23.
- [MYAS08] Minhas, U.; Yadav, J.; Abounaga, A.; Salem, K. "Database systems on virtual machines: How much do you lose?" In: Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on, 2008, pp. 35 –41.
- [PPS⁺07] Payne, B. D.; Perez, R.; Sailer, R.; Lee, W.; Cáceres, R. "A layered approach to simplified access control in virtualized systems", *Operating Systems Review*, vol. 41, pp 12-19, 2007.
- [RS07] Raj, H.; Schwan, K. "High performance and scalable i/o virtualization via self-virtualized devices". In: Proceedings of the 16th international symposium on High performance distributed computing, 2007, pp. 179–188.
- [SDG⁺07] Shivam, P.; Demberel, A.; Gunda, P.; Irwin, D.; Grit, L.; Yumerefendi, A.; Babu, S.; Chase, J. "Automated and on-demand provisioning of virtual machines for database applications". In: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, 2007, pp. 1079–1081.
- [SKM08] Singh, A.; Korupolu, M.; Mohapatra, D. "Server-storage virtualization: integration and load balancing in data centers". In: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, 2008, pp. 53:1–53:12.

- [SMA⁺08] Soror, A. A.; Minhas, U. F.; Abounaga, A.; Salem, K.; Kokosielis, P.; Kamath, S. "Automatic virtual machine configuration for database workloads". In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, pp. 953–966.
- [Sol12] "Solaris containers". [Online; acessado em 19 de julho de 2012], Capturado em: <http://www.oracle.com/technetwork/server-storage/solaris/containers-169727.html>, 2012.
- [tpc13] "Tpc consortion". [Online; acessado em 19 de janeiro de 2013], Capturado em: <http://www.tpc.org/tpcc/detail.asp>, 2013.
- [Tya09] Tyagi, N. K. "Virtualization of data base layer (vdbl)". In: GLOBAL JOURNAL OF ENTERPRISE INFORMATION SYSTEM, 2009, pp. 113–118.
- [VBVB09] Voorsluys, W.; Broberg, J.; Venugopal, S.; Buyya, R. "Cost of virtual machine live migration in clouds: A performance evaluation". In: CloudCom, 2009, pp. 254–265.
- [WWS⁺10] Wada, Y.; Watanabe, Y.; Syoubu, K.; Sawamoto, J.; Katoh, T. "Virtual database technology for distributed database", *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 0, 2010, pp. 214–219.
- [XNR⁺13] Xavier, M.; Neves, M.; Rossi, F.; Ferreto, T.; Lange, T.; De Rose, C. "Performance evaluation of container-based virtualization for high performance computing environments". In: Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on, 2013, pp. 233–240.