

OpenBox: Ferramenta de Alta Disponibilidade e de Gerenciamento de Cópias de Segurança de Dados

Victor Dias de Oliveira^{1,2}, Matheus Bandini¹, Fábio Licht²,
Bruno Schulze¹, Antonio Mury¹

¹Laboratório Nacional de Computação Científica (LNCC)
Av. Getúlio Vargas, 333 – Quitandinha – 25660-004 – Petrópolis – RJ – Brasil

²Universidade Estácio de Sá (UNESA)
Rua Bingen, 45 – Bingen – 25651-075 – Petrópolis – RJ – Brasil

Abstract. *This paper presents a tool for distributed virtualized environments, aiming to ensure the high availability of information in Database Management Systems (DBMS) in the context of cloud computing. The tool is capable of monitoring high availability systems, assigning functions for the better management of backups files, with the ability of restoring them to different DBMS available for use.*

Resumo. *Este artigo apresenta uma ferramenta para ambientes distribuídos virtualizados, com o objetivo de garantir a alta disponibilidade de informações por parte dos Sistemas de Gerenciamento de Banco de Dados (SGBD) no contexto de computação em nuvem. A ferramenta é capaz de monitorar um sistema de alta disponibilidade, atribuindo funções para gerenciar a criação de cópias de segurança, com a capacidade de restauração a diversos SGBDS existentes no mercado.*

1. Introdução

O avanço tecnológico das últimas décadas proporcionou diversas abordagens fundamentadas em computação para a solução de sistemas complexos, desde a ampla utilização de supercomputadores até a difusão da Internet com conexão de alta velocidade, permitindo que ambientes de computação paralela e distribuída de alto desempenho pudessem ser idealizados e implantados, garantindo a alta disponibilidade dos sistemas computacionais.

Aliado a isto, o avanço das tecnologias de virtualização de recursos computacionais e de redes vieram a contribuir para o surgimento do conceito de computação em nuvem, possibilitando a replicação de infraestruturas (IaaS – *Infrastructure-as-a-Service*), plataformas (PaaS – *Platform-as-a-Service*) e aplicações (SaaS – *Software-as-a-Service*), por meio da virtualização de ambientes dedicados específicos, atendendo às demandas mais pontuais em função das necessidades dos usuários [Vecchiola et al. 2009].

Devido à demanda crescente no uso de ambientes computacionais de alto desempenho e, conseqüentemente, à grande massa de dados gerada pelas aplicações nesses ambientes, surge a necessidade de gerenciar essas informações de forma a garantir a sua segurança e integridade [Wiesmann et al. 2000], uma vez que a perda ou inconsistência de um banco de dados pode acarretar em perdas significativas de informações importantes e até mesmo perdas monetárias, no caso do mundo comercial [Ekanayake and Fox 2009].

No caso da alta disponibilidade, ela pode ser alcançada em diversos níveis, incluindo os de aplicação, infraestrutura, *datacenter* e até o de redundância geográfica [Filho 2004]. Na sua forma mais básica, configurações de alta disponibilidade em infraestrutura são constituídas por: balanceadores de carga, servidores *web* e servidores de banco de dados.

Dentro deste contexto, este trabalho tem como objetivos principais: 1) a implementação de um sistema robusto de banco de dados, capaz de garantir o armazenamento seguro das informações para ambientes de computação distribuída, de modo que uma falha qualquer não ocasione a interrupção de seu funcionamento; 2) garantir esse funcionamento através de *middlewares* de replicação de bancos de dados, de modo a ser transparente tanto para o usuário quanto para a aplicação e; 3) apresentar a ferramenta desenvolvida para gerenciar a geração de cópias de segurança das informações, compactando-as e permitindo a restauração para mais de um SGBD.

Este trabalho encontra-se organizado da seguinte forma: a seção 2 apresenta trabalhos relacionados com ferramentas de gerenciamento de cópias de segurança e restauração; a seção 3 descreve as funcionalidades e a arquitetura da ferramenta apresentada neste trabalho e a seção 4 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Apresentaremos a seguir alguns dos trabalhos existentes relacionados ao conceito de *backup*, restauração e gerenciamento de banco de dados, utilizados na proposta apresentada.

2.1. Ferramentas de Apoio e Gerenciamento de Banco de Dados

O PgDump é um utilitário capaz de criar e gerenciar cópias de segurança de um banco de dados PostgreSQL. É possível realizar cópias de segurança consistentes, mesmo que o banco de dados esteja sendo utilizado. Os *backups* podem ser feitos no formato de *script*, personalizado ou tar. As cópias de segurança no formato de *script* são arquivos de texto puro, contendo os comandos SQL necessários para reconstruir o banco de dados no estado em que este se encontrava quando foi salvo [Momjian 2001].

O S.O.S Backup EasyDB é um software que realiza *backups online* dos bancos de dados MySQL e PostgreSQL. Tem como características principais: 1) *Backup online* para as bases de dados MySQL e PostgreSQL e; 2) Gerenciamento remoto pelo S.O.S Backup Manager; 3) Compressão de dados no formato ZIP. Entretanto, algumas das suas limitações consistem em ser um *software* que funcione apenas em sistemas operacionais Windows, e por apenas apresentar uma versão paga do sistema [Virtos, SOSBackup 2010].

Bacula é um conjunto de programas que permite administrar *backup*, restauração e verificação dos dados de computadores em uma rede de sistemas mistos. Tem como características os seguintes itens: 1) Módulos para diferentes sistemas operacionais; 2) Infinitude de recursos para a customização de *backups*; 3) Ferramenta de *backup* multi-banco-de-dados; 4) Suporte para MySQL, PostgreSQL e SQLite. Entretanto, essa ferramenta realiza suas cópias de segurança através da cópia do diretório aonde os SGBDs armazenam toda a estrutura do banco de dados. Isso não permite que os dados de um SGBD sejam migrados para outro. [Di Francesco and Sweden 2012]

As ferramentas citadas demonstram ser bastante maduras em função de seu tempo de uso e também pela quantidade de investidores e desenvolvedores envolvidos. Estas ferramentas são complexas e demandam um estudo aprofundado por parte dos interessados. Além disso, não oferecem suporte à alta disponibilidade.

Procurou-se neste trabalho disponibilizar uma ferramenta simples, capaz de garantir a alta disponibilidade das informações e, ao mesmo tempo, facilitar o gerenciamento das cópias de segurança, por meio da transposição de dados entre diferentes SGBDs. Mais detalhes serão apresentados na Seção 3.

3. Funcionalidades e Arquitetura da Ferramenta

Nesta seção serão descritos detalhes da arquitetura, do funcionamento e benefícios do sistema proposto, bem como a ferramenta de gerenciamento de rotinas de criação de cópias de segurança - OpenBox (**disponível para acesso através da URL: <http://comcidiv01.lncc.br/openbox>**).

3.1. Ferramenta OpenBox

O OpenBox foi desenvolvido integrando as ferramentas apresentadas na Seção 2 com o objetivo de oferecer um ambiente funcional, robusto e de fácil uso, no âmbito de gerenciamento de banco de dados, podendo ser utilizado para o apoio de aplicações no contexto de Computação em Nuvem. O sistema é subdividido em dois módulos: No primeiro é proposto, no contexto de Computação em Nuvem, um serviço de *IaaS* com o objetivo de garantir a alta disponibilidade de dados em ambiente que utilize o SGBD PostgreSQL. O segundo tem por objetivo gerenciar remotamente os recursos disponibilizados, capaz de realizar a cópia de segurança das informações, a restauração e o agendamento de tarefas de forma agregada, simplificando a operação destas tarefas por parte do usuário. A Figura 1 demonstra de maneira simplificada a arquitetura do sistema OpenBox.



Figura 1. Arquitetura do Sistema

O primeiro módulo, denominado ambiente de alta disponibilidade, tem por objetivo tornar o armazenamento de dados robusto, consistente e seguro, utilizando-se de um *middleware* cuja função é se comportar como uma camada de abstração entre os usuários e o *cluster* de banco de dados que compõem o sistema. Para a obter essa abstração foi utilizada a ferramenta PgPool-II [Perkov et al. 2011].

O segundo módulo é responsável pelo gerenciamento do ambiente de alta disponibilidade provendo informações sobre cada máquina do sistema, como por exemplo, a quantidade de memória consumida e a disponível, a capacidade em disco, os dados trafegados pela interface de rede, entre outras funcionalidades. Essas informações são obtidas

pelo uso da ferramenta ICAL (*Intelligent Collector Algorithm*) [Alvarenga et al. 2012]. Entretanto, dada a necessidade de cópias de segurança automatizadas, a ferramenta também é capaz de gerar e agendar cópias padronizadas e restaurá-las para SGBDs diferentes (no momento para PostgreSQL e MySQL), tornando-se assim, uma ponte para a troca de informações. Todos esses recursos são acessadas a partir de uma interface *web* com funcionalidades específicas de acordo com o nível de permissão do usuário.

3.2. Módulo I - Ambiente de Alta Disponibilidade

Para garantir a segurança das informações, optou-se por utilizar o *middleware* de replicação de bases de dados denominado PgPool-II. Suas características principais, que influenciaram o seu uso neste trabalho, são: o modo de replicação síncrona, o balanceamento de carga, o *pool* de conexões, a capacidade de detecção de falhas, a remoção automática do nó defeituoso e a execução de *scripts* personalizados na realocação daquele nó que apresentava falhas.

De forma prática, a replicação do banco de dados com PgPool-II é composto por 3 máquinas virtuais. A primeira é denominada nó principal, e as outras duas são as máquinas que mantêm o servidor de banco de dados. Todos os nós representados pelas máquinas virtuais fazem parte da infraestrutura virtualizada do ambiente (*IaaS*).

O nó principal, o servidor que gerencia o ambiente através do PgPool, é a máquina intermediária entre a aplicação e os servidores PostgreSQL. Este age de forma transparente e suporta até 128 nós de replicação. Quando uma requisição é realizada por meio de uma aplicação, essa ação será retransmitida do nó principal até os servidores PostgreSQL. Embora o PgPool-II seja visto pela aplicação como um servidor PostgreSQL real, os servidores de *backend* o veem como um de seus clientes. Isso acontece devido à característica de transparência do *middleware*.

Se um banco de dados é replicado (configurado no modo de replicação), quando realizada uma consulta em qualquer um dos servidores, o resultado é sempre o mesmo, pois as bases de dados são mantidas sincronizadas. Dada essa característica, foi possível configurar o PgPool-II para aproveitar a funcionalidade de replicação, a fim de reduzir a carga em cada servidor PostgreSQL. Ele faz isso através da distribuição de consultas de leitura (SELECT) entre os servidores disponíveis, melhorando o rendimento global do sistema. Em um cenário ideal, o desempenho de leitura melhora a medida em que número de servidores de banco de dados aumenta. Esse método, denominado de Balanceamento de Carga, é indicado para um ambiente onde há um grande número de usuários que executam muitas transações de leitura ao mesmo tempo. A Figura 2 apresenta a estrutura de alta disponibilidade do sistema.

3.3. Módulo II - Interface do Gerenciamento de Cópia de Segurança

O módulo II foi desenvolvido com o objetivo de gerar cópias de segurança, gravando os dados selecionados pelo administrador em um formato padrão de arquivo definido pelo sistema de forma segura. Além disso, permite a restauração desses dados não apenas aos SGBDs de origem como também para outros, tais como o MySQL, SQLite, SQLServer, dentre outros, tornando o OpenBox uma ponte capaz de interligar diversos sistemas gerenciadores de banco de dados.

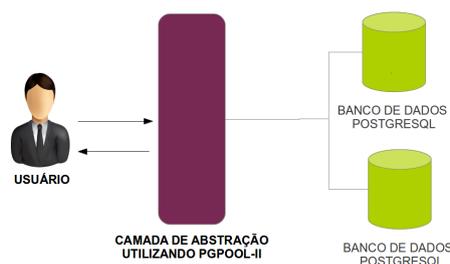


Figura 2. Ambiente de alta disponibilidade configurado pelo OpenBox

Este módulo é uma aplicação *SaaS* (*software* como serviço) no contexto de Computacional em Nuvem, permite um melhor gerenciamento das cópias de segurança, possibilitando a sua restauração para diferentes SGBDs, reduzindo o tamanho dos arquivos através da compactação e permitindo *backups* agendados. Os *backups* são armazenados na nuvem pelo OpenBox ficando a disposição do administrador sempre que necessário. O OpenBox foi desenvolvido utilizando a linguagem de programação JAVA em conjunto com a biblioteca JQuery (biblioteca javascript para realizar chamadas ao sistema de maneira assíncrona).

3.3.1. Núcleo do Sistema

O núcleo do OpenBox é o sistema encarregado de descobrir todas as características de um banco de dados. Ele tem como função varrer a estrutura de um SGBD, determinando as tabelas existentes, as estruturas de cada uma, suas chaves e metadados.

Para executar estas funções em vários SGBDs, foi desenvolvida uma ferramenta para criar uma camada de abstração entre os bancos de dados, denominadas de dialetos. Como cada sistema de gerenciamento de banco de dados possui suas funções e modo de interagir com seus respectivos banco de dados, o OpenBox, através da criação do dialetos para cada SGBD, permite que os diversos bancos de dados troquem informações entre si, funcionando como um canal de troca de mensagens entre eles.

O gerenciamento de arquivos, o gerenciamento de tarefas, a compactação das cópias de segurança, a funcionalidade de *backup* e a restauração são algumas das funcionalidades executadas pelo o núcleo desenvolvido nesse projeto.

3.3.2. Dialetos

Um dialeto é a maneira pela qual uma língua é falada em uma região específica. Logo, trata-se de uma variante linguística sobre um mesmo elemento, constituída de características próprias.

Os SGBDs fazem uso da linguagem estruturada denominada SQL. Esta é a linguagem padrão para interagir com bancos de dados relacionais. Entretanto, cada SGBD possui maneiras próprias de interagir com seu respectivo sistema. Nesse momento, o OpenBox (por meio da camada de dialetos) tem a capacidade da tradução dessas características próprias dos SGBDs, tornando possível a troca de dados entre o PostgreSQL e

o MySQL (podendo ser estendido para outros SGBDs).

3.3.3. Pool de Conexões

O *pool* de conexões é a parte encarregada de gerenciar todas as conexões criadas no OpenBox. Com essa funcionalidade é possível estabelecer conexões com PostgreSQL e o MySQL. Abrir conexões de modo excessivo em banco de dados é um processo custoso, podendo causar a diminuição da capacidade de atender a solicitações dos usuário. Por esse motivo, o OpenBox limita para até 5 o número máximo de conexões por usuário.

3.3.4. Cópia de Segurança

A cópia de segurança utiliza-se da estrutura padrão gerada em XML, formato para a criação de documentos com dados organizados de forma hierárquica. Através do uso de arquivos XML, o OpenBox é capaz de criar interfaces com outros bancos distintos permitindo então conter os mesmos dados. A hierarquia da estrutura XML é composta primeiramente da seguinte ordem: nome da base de origem, nome da tabela e finalizando, cada linha da tabela é separada pelo nome da coluna e seus respectivos valores.

3.3.5. Gerenciador de Tarefas

O gerenciador de tarefas, é o modulo capaz de identificar ações realizadas no ambiente OpenBox. Ao realizar um *backup* ou uma restauração, o gerenciador tem como função verificar o status das tarefas procurando saber se a mesma foi concluída com êxito, se está em andamento ou se em algum momento houve uma falha. Além de apresentar informações tais como: o nome do banco de dados, a tabela envolvida, o SGBD originado e o endereço IP.

3.3.6. Gerenciador de Arquivos

O gerenciador de arquivos é um programa *web* que serve para gerenciar e organizar os arquivos armazenados na nuvem do OpenBox. Tem como funções em sua interface: 1) Excluir as cópias de segurança; 2) Obter informações extras dos arquivos; 3) Compactar os backups; 4) Visualizar os arquivos no formato XML na web e; 5) Selecionar um cópia de segurança para restauração.

3.3.7. Restauração da Cópia de Segurança

Para utilizar-se da restauração de dados entre SGBDs diferentes oferecida pelo OpenBox, primeiramente o usuário deverá estabelecer uma conexão principal da qual fará uso. Após esta configuração, o usuário deve selecionar qual cópia de segurança deseja restaurar.

A restauração para uma nova tabela é realizada através da verificação da existência do nome digitado com as tabelas existentes no banco de dados. Entretanto, caso o nome já se encontra utilizado, o usuário é informado e a restauração é desabilitada. Caso

contrário, o núcleo do OpenBox se encarrega de gerar a tabela automaticamente e inserir as informações. Com o uso dos dialetos, é possível criar e executar comandos SQL automaticamente em tempo de execução conforme as características de cada SGBD.

3.3.8. Agendamento de Backup

O agendamento de *backup* é uma funcionalidade da ferramenta OpenBox na qual é especificada a tabela cuja cópia de segurança deseja-se criar, bem como a data e a hora para execução da tarefa. O OpenBox faz uso do Quartz, biblioteca desenvolvida em java para agendamentos de tarefas.

4. Conclusões e Trabalhos Futuros

Esta ferramenta foi desenvolvida com o objetivo de garantir a alta disponibilidade através de *clusters* de banco de dados, utilizando os recursos virtualizados disponíveis na nuvem.

O OpenBox é uma ferramenta que possibilita a recuperação de bases inconsistentes, utiliza-se do balanceamento de carga entre os servidores com o objetivo de obter ganho de desempenho, além de ser capaz de gerenciar a criação de cópias de segurança, de modo a permitir sua restauração para diferentes tipos de SGBDs, de forma simples, robusta, com isso apresenta-se como uma ferramenta de assessoramento aos seus usuários contribuindo para tomadas de decisões relativas à geração de *backup*.

Para monitorar e gerenciar os recursos disponíveis no ambiente, utiliza-se a ferramenta de monitoramento denominada ICAL (*Intelligent Collector Algorithm*), que devido a sua flexibilidade, pode de ser configurado de acordo com a situação e o nível de detalhamento das informações desejadas, reduzindo a sobrecarga dos servidores.

Como propostas para trabalhos futuros, surgem como possibilidades: a integração de mais SGBDs à ferramenta de gerenciamento de cópias de segurança, através do desenvolvendo de novos dialetos para tradução das características específicas de cada banco de dados. Atualmente, os esforços estão concentrados em desenvolver dialetos para os gerenciadores de banco de dados SQLite e o SQLServer. Além disso, a integração também, da ferramenta Linux Heartbeats, cuja função é monitorar o status do servidor e permitir que uma nova instância do nó principal assuma as funções em caso de falhas. Pretende-se também avaliar quantitativamente o ganho de desempenho resultante do balanceamento de carga obtido (item 2 parágrafo seguinte).

Dentre as principais contribuições alcançadas neste trabalho, destacam-se: 1) o desenvolvimento de uma ferramenta segura e de simples utilização, capaz de utilizar um *cluster* de banco de dados para garantir a alta disponibilidade das informações, por meio do uso de ferramentas de replicação de dados e balanceamento de carga. 2) com a realização de testes envolvendo o balanceamento de carga proveniente das aplicações, foi possível observar um ganho de desempenho dos servidores, devido à distribuição das requisições concorrentes; 3) a gerência de cópias de segurança do OpenBox possibilita, através de uma interface de fácil uso, o seu gerenciamento e a criação de uma ponte de troca de dados entre diferentes SGBDs; 3) o monitoramento dos recursos do *cluster* de alta disponibilidade, feito com o auxílio da ferramenta ICAL, oferece informações a respeito do consumo de CPU, memória e disco por parte dos serviços de banco de dados;

4) o desenvolvimento de Dialeto permite a integração de diversos sistemas gerenciadores de banco de dados e; 5) a incorporação de um módulo capaz de realizar *backups* compactados torna possível a redução do espaço em disco necessário para armazenar as informações.

Agradecimentos

Este trabalho teve o apoio do Ministério de Ciência, Tecnologia e Inovação (MCTI), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - PCI/LNCC) e da Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ - E-26/112.221/2008).

Referências

- Alvarenga, V., Yokoyama, D., Barbosa, J., Bandini, M., Schulze, B., and Mury, A. (2012). Ferramenta para monitoramento e gerência de ambientes virtuais de computação de alto desempenho. In *Salão de Ferramentas / XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 888–895.
- Di Francesco, P. and Sweden, V. (2012). Design and implementation of a mlfq scheduler for the bacula backup software.
- Ekanayake, J. and Fox, G. (2009). High performance parallel computing with clouds and cloud technologies. In *Cloud Computing - 1st Intl. Conference, CloudComp 2009, Munique, Alemanha, Out. 19-21, 2009 Revised Selected Papers*, volume 34 of *LNICST, Social Informatics and Telecommunications Engineering*, pages 20–38. Springer.
- Filho, N. A. P. (2004). Serviços de pertinência para clusters de alta disponibilidade. Technical report, Universidade de São Paulo, Disponível em: <http://www.teses.usp.br/teses/disponiveis/45/45134/tde-04102004-104700/publico/dissertacao.pdf>.
- Momjian, B. (2001). *PostgreSQL: introduction and concepts*, volume 192. Addison-Wesley.
- Perkov, L., Pavkovic, N., and Petrovic, J. (2011). High-availability using open source software. In *MIPRO, 2011 Proceedings of the 34th International Convention*, pages 167–170. IEEE.
- Vecchiola, C., P, S., and Buyya, R. (2009). High-performance cloud computing: A view of scientific applications. In *ISPAN*, pages 4–16. IEEE CS.
- Virtos, SOSBackup (2010). Virtos: Backing Up Data at Lightning Speed. Disponível em <http://software.intel.com/en-us/articles/virtos-backing-up-data-at-lightning-speed>.
- Wiesmann, M., Pedone, F., Schiper, A., Kemme, B., and Alonso, G. (2000). Understanding replication in databases and distributed systems. In *Proceedings of the 20th International Conference on Distributed Computing Systems, 2000*, 1:464–474.