

# CLUSTER DE ALTA DISPONIBILIDADE DE BANCO DE DADOS POSTGRESQL

Rodrigo Welter\*  
Otilia Donato Barbosa\*\*

## Resumo

Este artigo apresentou um estudo sobre ferramentas para promover a alta disponibilidade do serviço de banco de dados em uma rede de computadores. O estudo foi realizado sobre o Banco de dados PostgreSQL na versão 9.2.3 com o uso da ferramenta de replicação de banco de dados nativa deste banco, *streaming replication*. Além desta ferramenta, foi utilizada a *Heartbeat*, para que se possa deixar o serviço com alta disponibilidade sem a intervenção do administrador de banco de dados. O estudo considerou o uso de *softwares* livres, mostrando que é possível criar uma estrutura de alta disponibilidade do serviço de banco de dados sem a necessidade de altos investimentos.

Palavras-chave: Banco de dados. *Streaming Replication*. *Heartbeat*.

## 1 INTRODUÇÃO

Considere uma pequena empresa que presta serviços de contabilidade, por exemplo, com 10 funcionários os quais dependem do perfeito funcionamento do serviço de banco de dados para que possam usar o sistema para realizar suas atividades diárias, como a apuração dos impostos devidos das empresas ao qual o escritório de contabilidade representa. Se o serviço de banco de dados falhar e indisponibilizar o acesso ao sistema para a apuração dos impostos no dia em que vence o prazo para pagamento dos tributos, o que poderia acontecer com esse escritório? Uma solução de alta disponibilidade do serviço de banco de dados evitaria problemas.

Este artigo apresenta uma solução de alta disponibilidade em um Sistema Gerenciador de Banco de Dados (SGBD) e outras ferramentas de replicação e disponibilidade de serviços gratuitos, ou seja, suas licenças de uso não exigem investimentos financeiros. Esta solução pode ajudar as pequenas empresas que utilizam banco de dados a evitarem problemas de indisponibilidade deste serviço.

Entre os tipos de *clusters* existentes (*cluster* de alta disponibilidade, de balanceamento de carga, de alto desempenho, etc.), optou-se por apresentar o primeiro. Como a proposta para a alta disponibilidade do serviço se destina ao serviço de banco de dados, foi necessário escolher uma técnica de replicação de dados entre os nós do *cluster*. Entre os diversos tipos de replicação existentes (*off-line*, *on-line*, etc.), escolheu-se a replicação assíncrona por *streaming replication* em razão do índice de impacto menor para os usuários dos sistemas conectados aos SGBDs.

Existem diversos tipos de replicação de dados, a citar os principais: *off-line*, *on-line*, síncrona, assíncrona, etc. Entre as estudadas, a que se destaca pela facilidade de implementação, bem como pelo seu desempenho em ambientes de empresas de pequeno porte, é a replicação assíncrona.

Os SGBDs *freeware* são usados na maioria das empresas de pequeno porte que geralmente não dispõem de um profissional dedicado à administração do banco de dados que possa configurar os serviços de alta disponibilidade. O que muitos administradores de redes de computadores não sabem é que existem soluções fáceis e baratas que podem ser implementadas unicamente com *softwares* livres, como se pretende demonstrar com este trabalho.

\* Pós-graduado em Administração de Banco de dados e *Business Intelligence* pela Universidade do Oeste de Santa Catarina de São Miguel do Oeste; rodrigo007@gmail.com

\*\* Mestre em Engenharia Biomédica e Informática Industrial; Professora dos Cursos de Sistemas de Informação e Ciência da Computação da Universidade do Oeste de Santa Catarina de São Miguel do Oeste; Rua Oiapoc, 211, São Miguel do Oeste, SC; otília.barbosa@unoesc.edu.br

## 2 DESENVOLVIMENTO

### 2.1 REFERENCIAL TEÓRICO

Os Sistemas Gerenciadores de Banco de Dados (SGBDs) Relacional são utilizados para armazenar informações de soluções de informática em todas as áreas de negócios existentes, bem como para administrar o acesso a tais informações. O PostgreSQL é um SGBD Relacional de código aberto, muito utilizado por sua confiabilidade e integridade dos dados (POSTGRESQL, 2013).

A *High Availability* (HA) ou alta disponibilidade é a característica de um sistema em possuir um grau elevado de continuidade da sua operação, ou seja, em manter a disponibilidade de seu serviço o maior tempo possível. Atualmente esta característica é exigida da maioria dos sistemas computacionais e os investimentos para prover a disponibilidade vêm sendo proporcionais à importância deste serviço para a empresa (DEXTRANING, 1999-2013).

Soluções de alta disponibilidade de banco de dados requerem basicamente dois cuidados básicos ao seu perfeito funcionamento. O primeiro e principal é a integridade das informações do banco de dados durante a sua replicação; o segundo, não menos importante, é a solução que identifica algum problema no nó que compõe a solução e transfere o serviço para o nó do segundo servidor, chamado de servidor *slave* (DEXTRANING, 1999-2013).

Diretamente ligado ao conceito de alta disponibilidade está o conceito de *downtime* que significa o tempo de indisponibilidade de um serviço para uma empresa e é usado para se ter uma métrica da sua disponibilidade. As causas mais comuns de *downtime* de serviços estão relacionados a 27% *softwares*, 23% *hardware*, 18% erro humano, 17% rede, 8% desastres naturais e 7% erros desconhecidos (DEXTRANING, 1999-2013). Especialmente sobre a rede, conforme citam Lopes, Sauve e Nicolletti (2003), mesmo o melhor sistema de gerência de redes não pode evitar todas as falhas.

Também relacionado ao conceito de alta disponibilidade está o conceito de *cluster* que é definido como um sistema que une dois ou mais computadores ou sistemas (denominados nós), que trabalham em conjunto para executar tarefas, de forma que os usuários tenham a impressão de que um único computador responde a eles, criando a ilusão de um recurso único; tal definição é denominada transparência do sistema (DONATO, 2009).

Existem, basicamente, três tipos de *cluster*: de alta disponibilidade, de balanceamento de carga e de alto desempenho.

O *cluster* de alta disponibilidade tem como objetivo principal manter o sistema disponível por meio de técnicas de tolerância a falhas. Para atender a esta técnica, utiliza-se da redundância de *hardware* e *software* quando geralmente existe um *software* a ser configurado com o objetivo de monitorar e identificar a falha em algum nó e para permitir o *failover*, ou seja, executar ações necessárias para manter os recursos disponíveis em outro(s) nó(s).

O *software Heartbeat*, *software* livre e não nativo dos SGBDs PostgreSQL, é um exemplo de *software* que possui essas funcionalidades e permite a implementação deste tipo de *cluster*. A redundância de recursos que necessitam de atualização constante, como é o caso dos SGBDs, implica o uso de ferramentas de replicação (DEXTRANING, 1999-2013).

A replicação de banco de dados é o processo de duplicação ou compartilhamento de informações de maneira consistente entre os servidores envolvidos em um *cluster*. Segundo Milani (2008), bem como os demais bancos de dados, o PostgreSQL oferece os recursos necessários para realizar a replicação entre servidores. A vantagem sobre outros bancos é que sua licença abre este recurso ao uso gratuito até para aplicações comerciais, diferentemente de outras licenças de *software* livre utilizadas por outros bancos de dados.

Uma sincronização de dados acontece a partir do momento em que a informação se torna consistente no SGBD e pode ser feita de maneira síncrona ou assíncrona (SMANIOTO, 2008).

Na sincronização síncrona, a replicação da ação é feita instantaneamente. Se alguma cópia do banco é alterada, esta alteração será imediatamente aplicada a todos os outros bancos dentro da transação. A replicação síncrona é apropriada em aplicações comerciais em que é exigido um nível de atualização muito preciso em todos os servidores envolvidos. A desvantagem deste modelo é a perda sensível da *performance* (SMANIOTO, 2008).

No modelo assíncrono, o replicador monta um histórico das ações a serem replicadas e em um determinado momento é feita a replicação entre as bases de dados relacionadas. A alteração será propagada e aplicada para outra base em um segundo passo, dentro de uma transação separada. Esta poderá ocorrer em segundos, minutos, horas ou até dias depois, dependendo da configuração preestabelecida (SMANIOTO, 2008).

A ferramenta *Heartbeat* é o principal componente do projeto Linux-HA (*High Availability*). O processo deste *software* é executado em segundo plano no sistema operacional e permite a comunicação entre os nós de um *cluster* para identificar e tolerar as falhas. Segundo Dextraining (1999-2013), o projeto Linux-HA, ao qual pertence o *software Heartbeat* é compatível com as plataformas Linux, FreeBSD, Solaris e Mac OS/X, fazendo parte de algumas das seguintes distribuições: Debian, RedHat, SUSE, Mandriva, TurboLinux, Gentoo, entre outras. O *Heartbeat* conta com uma interface gráfica chamada *Heartbeat GUI*.

## 2.2 MATERIAIS E MÉTODOS

Para que fosse possível realizar este estudo, foi necessário utilizar dois servidores, ambos com sistema operacional Linux com a distribuição OpenSuse versão 12.2. Estes dois servidores foram criados utilizando o *software* de virtualização VMWare Workstation versão 8.0 da empresa VMWare para emular o ambiente do servidor.

Um dos servidores recebeu o nome de servidor *master* e o outro, servidor *slave*. Em ambos foi instalado e configurado o SGBD PostgreSQL versão 9.2.3.

O próximo passo foi realizar a configuração da técnica de replicação dos dados do servidor *master* para o *slave*. Não houve a necessidade de instalação de nenhuma ferramenta para esta funcionalidade, pois a ferramenta de replicação usada é nativa do PostgreSQL, ou seja, foi instalada com a instalação deste.

Entre os diversos tipos de replicação disponíveis a escolhida para o desenvolvimento desta pesquisa foi a de *streaming replication* em modo de operação *hot standby*. A *streaming replication*, que permite que um servidor *slave* permaneça em comunicação constante com o servidor *master* facilitando o recebimento das alterações de base geradas na base de dados do servidor *master*. Isso é possível em virtude dos dois processos:

- a) Processo *walreceiver*: iniciado no servidor *slave*;
- b) Processo *walsender*: iniciado no servidor *master*.

Estes dois processos utilizam-se de um canal de comunicação criado via protocolo TCP/IP. A criação deste canal inicia-se quando o PostgreSQL envia uma requisição de conexão ao servidor *master*; se a conexão for estabelecida, o processo *walsender* é criado. A partir de então, o processo *walsender* envia ao *walreceiver* as alterações efetivadas no servidor *master* para aplicá-las imediatamente no servidor *slave*.

A replicação com *streaming replication* em modo *hot standby* faz com que a base de dados do servidor *slave* seja disponibilizada somente em modo leitura à rede, para que haja integridade dos dados.

Basicamente, as configurações necessárias para prover a replicação entre os servidores *master* e *slave* foram realizadas nos arquivos *pg\_hba.conf*, *postgresql.conf* e *recovery.conf*, em ambos os servidores.

Finalizadas as configurações necessárias para que houvesse a replicação de dados, o próximo passo foi a instalação e a configuração do *software Heartbeat*.

As configurações do *Heartbeat* envolveram alguns arquivos, como:

- a) *authkeys*: especifica o tipo de autenticação;
- b) *há.cf*: responsável pela configuração global do *cluster*;
- c) *cib.xml*: contém informações sobre o estado dos recursos.

Para testar a interoperabilidade dos serviços, foi criada uma nova tabela no banco de dados *master* e aguardada a sua replicação para o servidor *slave*. Depois de conferida a replicação desta tabela para o servidor *slave*, foi parada a interface de rede do servidor *master*, com o objetivo de simular uma falha *hardware* e ou *software* deste servidor. Como a solução *Heartbeat* é responsável pela detecção das falhas, este serviço automaticamente alocou o endereço IP usado para a conexão com o banco de dados ao servidor *slave* e promoveu este a servidor *master*.

## 2.3 RESULTADOS DA PESQUISA

A instalação dos dois sistemas operacionais Opensuse em ambiente de virtualização não foi uma tarefa difícil. Foram selecionados apenas pacotes básicos para a instalação do Sistema Operacional e adotado o particionamento de disco sugerido pelo próprio instalador. Em ambientes de produção, ou seja, em situações reais, o particionamento de disco deve receber atenção especial, pois um mau particionamento e distribuição de carga nos discos pode diminuir consideravelmente a *performance* do servidor em razão de problemas de gargalo de disco. Servidores com tecnologias de RAID ajudam a manter a disponibilidade de um servidor quando os problemas estiverem ligados a falhas de disco, bem como resolvem problemas de gargalo de acessos a discos.

Para evitar incompatibilidades de *software*, optou-se por instalar o SGBD PostgreSQL nas mesmas versões em ambos os servidores. Conforme cita Almeida (2009), a replicações entre versões mínimas (8.3.0 para 8.3.1 ou 8.3.2 ou etc. e vice-versa) podem funcionar corretamente, mas o Grupo de Desenvolvimento Global do PostgreSQL aconselha utilizar servidores com tal número de versão.

A configuração da estrutura de replicação foi realizada somente depois que a rede de comunicação entre os dois servidores estava operando corretamente após os dois SGBDs, um no servidor *master* e outro no servidor *slave* estarem funcionando. Para testar a disponibilidade dos dois bancos foram criadas tabelas em ambos os bancos, conforme orientações das fontes pesquisadas.

A fase da configuração da replicação dos bancos de dados exigiu bastante cuidado, pois é nesta fase que os dados existentes na base de dados do servidor *master* são replicados ao servidor *slave*. Neste processo, caso existissem informações valiosas no servidor *slave*, estas seriam perdidas após o início da replicação. Este cuidado vale para configurar a replicação em ambiente de produção. Para assegurar a integridade do ambiente, foram feitas as devidas cópias de segurança (*dumps* de base, clones dos Sistemas Operacionais) antes de iniciar o processo.

A instalação da ferramenta *Heartbeat* foi realizada por meio do próprio sistema operacional por intermédio de *download* dos pacotes (pacotes .rpm) disponíveis no *site* do projeto Linux-HA. A configuração foi realizada com o *Heartbeat GUI*, ferramenta de interface gráfica sempre seguindo a documentação sugerida por DEXTRANING(1999-2013). É possível configurar o *Heartbeat* diretamente nos seus arquivos de configuração, mas este procedimento exige um pouco mais de conhecimento da ferramenta e de editores de texto para as alterações.

Toda a estrutura da pesquisa foi realizada usando apenas um *hardware* físico. Isso foi possível por causa do uso de *softwares* de virtualização de ambientes os quais permitiram que três sistemas operacionais pudessem ser executados em um só Sistema Operacional (Servidor hospedeiro com Windows 7, servidor *master* com OpenSuse 12.2 e servidor *slave*, também com OpenSuse 12.2). Durante o desenvolvimento da pesquisa, percebeu-se que o uso de recursos de virtualização é uma forte tendência em empresas, mas se tais recursos não forem usados de forma correta, todo um projeto pode estar em risco.

Para todas as etapas da implementação foram encontradas documentações, disponíveis, inclusive, no idioma Português Brasileiro.

## 3 CONCLUSÃO

Após a finalização da implementação da estrutura de alta disponibilidade, usando a replicação por *streaming* em modo *hot standby* com o uso do *software Heartbeat*, percebeu-se que existem soluções de simples implementação e

que não exigem altos investimentos financeiros para serem colocadas em prática. O grau de complexidade para a sua implementação não é considerado alto, o que permitiria que qualquer administrador de redes de computadores, dotado do conhecimento de alguns conceitos, pudesse implementar a solução em sua empresa.

Se uma empresa hoje usa um sistema cuja base de dados esteja hospedada em um SGBD PostgreSQL em versão superior a 9.0, o maior investimento ainda estará na disponibilização de um segundo servidor, o servidor *slave*, para que se possa montar a estrutura de alta disponibilidade dos serviços de banco de dados.

As possibilidades de investimentos são inúmeras para que se possa montar a estrutura de alta disponibilidade. Uma vez analisadas as necessidades, a escolha do *hardware* para o servidor *slave* poderá depender de alguns fatores, como em caso de falha do servidor *master*. Desse modo, haveria a necessidade de direcionar todas as conexões de usuários para o servidor *slave*? Se a resposta a esta pergunta fosse de que apenas uma pequena quantidade destas conexões precisassem ser direcionadas ao servidor *slave* sem que isso prejudicasse as atividades dos funcionários desta empresa, o *hardware* a ser adquirido não necessitaria possuir tais características de processamento do servidor *master*. Talvez, poderia ser usado algum *hardware* já disponível na empresa e que pudesse ser aproveitado para essa funcionalidade, evitando um novo investimento.

Em ambientes corporativos, o uso de máquinas virtuais para promover a alta disponibilidade é opcional. Se a empresa que deseja implementar a solução dispõe, em sua estrutura, um servidor físico e quer manter esta estrutura, basta providenciar a aquisição de mais um servidor com características de *hardware* parecidas com o servidor atual. Há também a possibilidade de usar um ambiente virtualizado para promover a alta disponibilidade (ao servidor *slave*).

Se a opção para se criar o servidor *slave* for com o uso de recurso de virtualização, os custos podem ser ainda menores ou até zerados. Se a empresa já possui a estrutura de virtualização e recursos de *hardware* suficientes para se criar o novo servidor *slave*, não haverá a necessidade de novos investimentos. Nesse caso, o investimento seria das horas de trabalho necessárias para a implementação da nova estrutura.

Com o desenvolvimento desta pesquisa, percebeu-se que existem inúmeras formas de promover a alta disponibilidade de serviços em uma rede de computadores para banco de dados. A melhor estratégia a ser adotada e implementada deve considerar as condições de investimento de uma empresa para a aquisição de *hardware* e contratação de profissionais para realizar as configurações quando estes profissionais não estiverem disponíveis na empresa.

A preocupação por disponibilidade de serviços em uma rede de computadores sempre foi e sempre será uma preocupação para administradores de tecnologia da informação. Mesmo as soluções sendo *freeware*, existem inúmeras empresas e desenvolvedores trabalhando para que sejam disponibilizadas novas ferramentas que promovam estas soluções. O que os administradores precisam é buscar tais soluções analisando as que mais se encaixam em sua estrutura.

### ***High availability cluster for PostgreSQL database***

#### *Abstract*

*This article presents a study on tools to promote high availability of database service in a computer network. The study was conducted on the PostgreSQL database version 9.2.3 using the replication tool of native database, streaming replication. Besides this tool, Heartbeat tool was used, to leave the service with high availability without database administrator intervention. The study considered the use of free software and took care to show that it is possible to create a structure of high availability database without the need for high investment service.*

*Keywords: Database. Streaming replication. Heartbeat.*

### **REFERÊNCIAS**

ALMEIDA, C. Replicação no PostgreSQL. **Revista SQLMagazine**, 2009. Disponível em: <<http://www.devmedia.com.br/articles/viewcomp.asp?comp=12687>>. Acesso em: 25 set. 2013.

DEXTRAINING. **PostgreSQL**: alta disponibilidade. São Paulo, 1999-2013.

DONATO, R. Gerenciando um cluster PostgreSQL com PGCluster. **Revista SQLMagazine**. 2009. Disponível em: <<http://www.devmedia.com.br/articles/viewcomp.asp?comp=7007&hl=>>. Acesso em: 25 maio 2013.

LOPES, R. V.; SAUVÉ, J. P.; NICOLLETTI, P. S. **Melhores práticas para gerência de redes de computadores**. Rio de Janeiro: Campus, 2003.

MILANI, A. **PostgreSQL**: guia do programador. São Paulo: Novatec, 2008.

POSTGRESQL. **Documentação**. 2013. Disponível em: <<http://www.postgresql.org.br/docs>>. Acesso em: 23 set. 2013.

POSTGRESQL. **Sobre o PostgreSQL**. 2013. Disponível em: <<http://postgresql.org.br/sobre>>. Acesso em: 25 set. 2013.

SMANIOTO, C. E. Replicação e alta disponibilidade no PostgreSQL. **Revista SQLMagazine**, 2008. Disponível em: <<http://www.devmedia.com.br/post-6140-Artigo-da-SQL-Magazine-24-Replicacao-e-alta-disponibilidade-no-PostgreSQL.html>>. Acesso em: 25 set. 2013.