Solução de Alta Disponibilidade para Banco de Dados MySql

Marcio Fernandes da Costa, Lucas Guilherme Diedrich, Wilson Varaschin

Coordenadoria de Tecnologia da Informação
Universidade Federal da Integração Latino-Americana (UNILA) – Foz do Iguaçu, PR – Brasil

{marcio.costa,lucas.diedrich,wilson.varaschin}@unila.edu.br

Abstract. *This article summarizes the database cluster solution in high availability implemented at the Federal University of Latin American Integration. The previous system was based on the use of only one database server, but even with several backups performed during the day, in the event of any failure that interrupted the operation of this service, all applications that depend on this data would be compromised. The new model brought more confidence in the whole system, including "zero downtime".*

Keywords: *cluster, high availability, database*

Resumo. *Este artigo apresenta de maneira resumida a solução de cluster de banco de dados em alta disponibilidade implementado na Universidade Federal da Integração Latino-Americana. O sistema anterior baseava-se na utilização de apenas um servidor de banco de dados, mas mesmo contando com várias cópias de segurança realizadas durante o dia, na ocorrência de qualquer falha que interrompesse o funcionamento deste serviço, todas aplicações que dependem destes dados seriam comprometidas. Com o novo modelo, obteve-se mais confiança no sistema como um todo, inclusive com downtime nulo.*

Palavras Chave: cluster, alta disponibilidade, banco de dados

1. Introdução

Ao disponibilizar serviços de missão crítica como banco de dados, além da necessidade de oferecê-los de forma segura, com boa performance, escalabilidade e confiabilidade, sem dúvida alguma a operação contínua é o fator mais importante para este tipo de serviço. Sob certo ponto de vista, podemos inferir que qualquer aplicação web atual utiliza algum tipo de banco de dados, mesmo que de forma transparente ao usuário, seja para armazenar poucos dados, como as configurações para o seu funcionamento, seja para armazenar volumes muito maiores, como o registro de autenticação dos usuários em uma rede.

Para oferecer serviços em alta disponibilidade, incluindo as capacidades de suportar e recuperar-se em casos de falhas ou manutenções, é obrigatório utilizar o princípio da redundância. Como afirma [COULOURIS et al, 2013, p.22], "os serviços podem se tornar tolerantes a falhas com o uso de componentes redundantes". Sem redundância não se pode falar em alta disponibilidade, porque o serviço apresentará o chamado **ponto único de falha** ou em inglês ***SPOF*** *(single point of failure)*, onde na ocorrência de sua falha, provocará a falha em todo o sistema.

Uma das soluções que podem ser utilizadas para a mitigação do ponto único de falha é a utilização de *clusters* (ou *clustering*), caracterizado por um sistema de dois ou mais computadores, trabalhando de maneira conjunta com a finalidade de processar tarefas, dividindo entre si as atividades de processamento e executando estes trabalhos de maneira simultânea.

Entre as justificativas para a disponibilização do serviço de banco de dados no formato de cluster, podemos citar:

– a aplicação de atualizações de segurança e correção de bugs para o sistema operacional ou aplicativos não interrompe o funcionamento do banco de dados, pois cada servidor do cluster funciona de maneira independente, podendo ser facilmente removido ou adicionado ao cluster;

– escalabilidade horizontal: a medida que a utilização aumenta, mais servidores podem ser adicionados ao cluster, de maneira que a carga de trabalho seja distribuída entre mais servidores;

– satisfação dos usuários: sistemas que estão sempre disponíveis e não apresentam interrupções melhoram a percepção dos usuários no que se refere ao uso de tecnologias para o desempenho das suas atividades diárias;

O presente trabalho trata da solução em uso na UNILA, onde foi implementada uma solução de alta disponibilidade para banco de dados MySql, utilizando as ferramentas open source de balanceamento de carga HAPROXY [1] e gerenciador de banco de dados Percona XtraDB Cluster (PXC) [2].

O balanceador de carga HAPROXY está disponível nas principais distribuições Linux do mercado e sua instalação e configuração são consideradas fáceis, oferecendo alta disponibilidade, balanceamento de carga e proxy para conexões do tipo TCP e HTTP.

O sistema gerenciador de banco de dados Percona XtraDB Cluster é uma solução ativo/ativo, de alta disponibilidade e escalabilidade, compatível com os sistemas de gerência de banco de dados MySql e MariaDB.

2. Métodos

O balanceador de carga HAPROXY foi escolhido para compor a solução por já estar em uso na universidade há algum tempo, provendo o balanceamento e alta disponibilidade a diversos serviços web, além de se mostrar estável, de fácil configuração e de baixo consumo de recursos computacionais. Ele não representa um ponto único de falha, visto que também está implementado em alta disponibilidade através das ferramentas open source **PCS/COROSYNC/PACEMAKER** [3].

O princípio de funcionamento destas ferramentas é o de entregar serviços críticos com alta disponibilidade, verificando constantemente se os recursos necessários estão em funcionamento e respondendo a requisições, e em caso de alguma falha ou recurso indisponível, fazer o start ou migração de serviços em outros nós da rede. Basicamente o HAPROXY recebe as requisições dos clientes e as distribui entre os servidores de banco de dados utilizando o algoritmo de balanceamento *leastconn*, recomendado para seções mais longas, como as de banco de dados [4].

Para a escolha do sistema de gerência de banco de dados foram levados em consideração alguns pontos, entre eles:

- deveria ser open source e de licença livre, de forma a não gerar custos de licenciamento à instituição;
- compatível com MySql e/ou MariaDB.
- ter *proxy protocol implementado*, que recebe do servidor proxy intermediário o endereço IP do cliente/aplicação que está acessando o banco de dados, em vez de receber o endereço IP do servidor proxy;
- suporte a IPv6, pois toda a comunicação entre servidores internos é preferencialmente realizada nesta versão do protocolo;

A escolha do Percona XtraDB Cluster foi considerada pelos seguintes motivos:

- com a instalação de apenas um arquivo (do tipo Linux/rpm), os softwares necessários já estariam disponíveis para o completo funcionamento do cluster, diferente, por exemplo, do MariaDB, onde seria necessária a instalação de mais alguns arquivos de plugins manualmente;
- o mesmo desenvolvedor conta com uma solução de backup chamada *Percona XtraBackup*, que permite realizar cópias de segurança completas ou incrementais em intervalos regulares, além da compressão dos dados, paralelismo na execução e facilidade para restore, inclusive podendo ser utilizada em banco de dados MySql/MariaDB que não estejam no formato de cluster;
- a existência para download de uma imagem do tipo conteiner (*Docker)*, com um conjunto de ferramentas e gráficos úteis para o monitoramento do cluster MySql, tanto no período inicial de testes e implantação, quanto para o uso diário. Esta imagem permite monitorar o

desempenho geral do cluster, uso dos processadores, memória e discos dos servidores, performance e parâmetros dos bancos de dados;

- manuais disponíveis para download no site do desenvolvedor;
- o desenvolvedor disponibiliza publicamente em seu site, em média, dois ou mais *webminars* técnicos mensalmente, demonstrando interesse no sentido de que seus produtos sejam mais e melhor utilizados;

Em relação à quantidade e aos equipamentos que compõem a solução de cluster, eles foram assim divididos:

- Três máquinas virtuais para o banco de dados, com 16GB de memória RAM e quatro processadores cada (Node1, Node2 e Node3);
- Duas máquinas virtuais para o HAPROXY (Haproxy1 e Haproxy2), com 2GB de memória RAM e um processador cada, sendo que ao cliente (usuário e/ou aplicação) é disponibilizado apenas o endereço IP virtual para a conexão ao banco de dados;

A figura 1 mostra a distribuição dos equipamentos da solução:

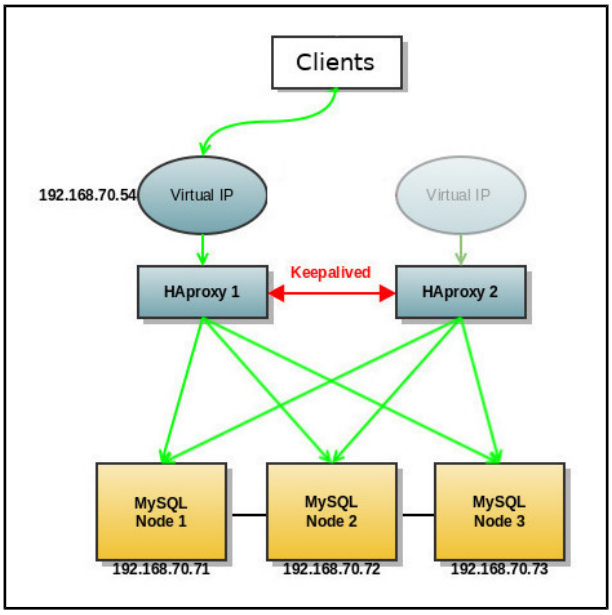

Figura 1. Distribuição dos equipamentos no cluster

Cada servidor de banco de dados trabalha de maneira independente: pode ter configurações de hardware e software diferentes, além de parâmetros de banco de dados que diferem dos outros servidores que compõem o cluster. Na situação em que são utilizadas configurações diferentes, pode-se atribuir pesos para cada servidor de banco de dados no HAPROXY, onde cada um deles poderá receber mais ou menos requisições de clientes dependendo do peso a ele atribuído.

Em relação ao armazenamento dos dados, cada servidor tem a sua própria base de dados local, efetuando as leituras e gravações de forma independente. Por não usar um ponto único e comum de gravação, e das bases de dados serem independentes, processos internos do sistema gerenciador de banco de dados se encarregam de replicar através da rede as transações entre cada nó, de modo que todos os bancos de dados de cada servidor tenham os mesmos dados, mantendo a integridade geral do cluster.

3. Resultados e conclusões

Com o uso do cluster MySql foram alcançados os seguintes resultados:

- *Downtime* de banco de dados nulo: considerando o pior cenário, onde apenas um dos três servidores de banco de dados continuar em funcionamento, o acesso aos dados ainda será possível;
- Distribuição da carga entre os servidores: se a utilização crescer de uma maneira inesperada, pode-se agregar mais equipamentos para um rebalanceamento da carga;
- Interrupções programadas ou não para manutenção: para a atualização dos pacotes do sistema operacional, aplicação do banco de dados, alteração de parâmetros ou alterações no hardware do servidor, basta que a aplicação do banco de dados seja parada. Assim, de maneira automática, o HAPROXY identificará que este servidor não está respondendo à requisições e não enviará mais pedidos para ele, deixando-o fora do cluster. Ao término da manutenção, após iniciar a aplicação de banco de dados, o administrador informa ao HAPROXY através de uma interface gráfica que o servidor já está pronto para receber requisições novamente;
- Monitoramento de métricas do banco de dados: através das informações disponibilizadas por ferramenta própria, eliminou-se um grande trabalho que seria necessário para configurar o monitoramento em outras aplicações.
- Cópias de segurança: são realizadas mais vezes durante o dia, em intervalos menores de tempo, mais rápidas no processamento e de tamanho menor, garantindo a salvaguarda dos dados.
- Além disso, vários SGBDs que estavam espalhados em diferentes servidores foram unificados no cluster, o que gerou um melhor balanceamento no hardware utilizado.

Levando em consideração a dependência das aplicações por banco de dados e eliminando pontos que poderiam levar a interrupções do serviço, entregamos de forma transparente aos clientes (neste cenário considerando aplicações ou pessoas), soluções de ponta para o desempenho das suas atividades.

Apesar de o modelo tratado nesse artigo necessitar uma maior utilização de recursos de hardware do que um servidor isolado, isso foi considerado irrelevante, pois a confiança gerada no novo modelo permitiu a migração de vários sistemas de gerência de bancos de dados espalhados para um único local, o *cluster*.

Referências

Coulouris, George. et al. (2013). Sistemas Distribuídos – Conceito e Projeto. 5 ed. São Paulo: Editora Bookman.

[1] http://www.haproxy.org/. Acesso em Março/2018.

[2] https://www.percona.com/software/mysql-database/percona-xtradb-cluster. Acesso em Março de 2018.

[3] https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/high_availability_add-on_overview/index. Acesso em Março de 2018.

[4] http://cbonte.github.io/haproxy-dconv/1.8/configuration.html#4.2-balance. Acesso em Março de 2018.