

## ***REPLICAÇÃO DE DADOS EM BANCOS DE DADOS DISTRIBUÍDOS***

### ***Distributed Databases: Technique of Replication Data***

Anderson de Melo Valadão e Rodrigo Martins de Faria

#### **RESUMO**

O presente trabalho apresenta algumas características dos Bancos de Dados Distribuídos e também a técnica de replicação de dados, utilizando-se o banco de dados Mysql como alternativa viável para quem deseja segurança, confiabilidade e agilidade no acesso e armazenamento de informações.

**Palavras-chave:** Banco de dados distribuídos; Mysql; SGBD.

#### **ABSTRACT**

This paper presents some characteristics of Distributed Databases and also the technique of replication data, using the Mysql database as a viable alternative for whom wants safety, reliability and speed of access and storage of information.

**Key words:** Distributed Databases; Mysql; SGBD.

#### **INTRODUÇÃO**

Com o avanço tecnológico e econômico, grandes empresas necessitam, cada vez mais, manipular informações de maneira segura e contínua. Para isso, necessitam de servidores que possam disponibilizar dados armazenados, de forma rápida e eficaz, aos usuários, de acordo com Date (2004), com os avanços nos estudos sobre Banco de Dados, muitas alternativas resultaram, e muitas técnicas de armazenamento de informações consolidaram-se, como oSBDD (Sistema de Banco de Dados Distribuídos). Em um SBDD, o banco de dados é armazenado em diversos computadores, que se comunicam por meio de redes de alta velocidade ou linhas telefônicas. Não compartilham memória

principal ou discos, e podem variar de tamanho e função.

Empresas de médio e pequeno porte, geralmente utilizam bancos de dados centralizados, em que o usuário faz uma requisição e um método coleta a informação no servidor local. No entanto, em algumas empresas de grande porte, o banco de dados centralizado vem sendo substituído pelo distribuído, por proporcionar mais autonomia, disponibilidade e crescimento incremental.

Segundo Date (2004), quanto maior a evolução no uso de redes, maior a possibilidade de se desenvolver novas e melhores soluções distribuídas. Observa-se esse fato ao se falar da internet que, dia a dia, expande seu quantitativo de usuários, permitindo a utilização em diversas aplicações distribuídas, além do aumento da largura da banda<sup>1</sup>, que possibilita uma comunicação mais fácil e rápida entre os sistemas e bancos distribuídos.

De acordo com Tanenbaum (2000), a tendência atual é de que a quantidade de dados também aumente, levando à necessidade de um maior processamento e acesso aos recursos das redes de dados. Esse crescimento fará com que o acesso aos dados seja mais lento.

Como, geralmente, a informação é de vital importância para o sucesso de qualquer empresa atual, é essencial que se utilizem as melhores tecnologias para proteger, tratar e guardar informações. Nesse contexto, está inserido também o conceito de replicação de dados: processo de copiar e manter várias cópias do mesmo dado em vários servidores, visando aumentar a disponibilidade das aplicações para garantir a integridade da informação.

Neste trabalho, trata-se de Banco de Dados Distribuído e da Replicação de Dados com o software MySQL, fazendo-se um levantamento das principais técnicas e dos principais conceitos utilizados, com exemplos para melhor entendimento e contextualização.

### **Banco de Dados Distribuído:**

Segundo Ozsu e Valduriez (2001), um banco de dados distribuído é uma

---

<sup>1</sup>Largura de banda é a medida em hertz da faixa de frequência, de um sistema ou sinal.

coleção de múltiplos bancos de dados, logicamente interligados e distribuídos, sobre uma rede de computadores.

Uma das principais vantagens de sua utilização seria a capacidade de partilhar informações pela distribuição de dados. Em Korth (1999), cada base, ou nó, da rede pode reter um grau de controle sobre os dados localmente armazenados. Se um nó falhar, em um sistema distribuído, os demais serão capazes de operar normalmente.

Outra vantagem é a aceleração das consultas. Se uma consulta envolve dados em diversos nós, é possível dividi-la em sub-consultas, que podem ser executadas em paralelo.

Esses bancos são mantidos por um sistema de gerenciamento de banco de dados (SGBD), que possibilita transparência para os usuários. Essa transparência, ao se falar de dados ou informações, pode ter três formas: transparência de localização, de fragmentação e de replicação.

A transparência de localização, também conhecida como transparência de rede, faz com que o usuário não precise saber onde se encontra o dado que acabou de solicitar. A transparência de fragmentação mantém a consistência e integridade dos dados fragmentados, e a transparência de replicação dá-se quando dados estão replicados em vários nós da rede, de forma sincronizada.

### **Replicação de Dados**

Segundo Ikematu (2001), o objetivo de um mecanismo de replicação de dados é permitir a manutenção de várias cópias idênticas de um mesmo dado, em vários sistemas gerenciadores de banco de dados.

A replicação aumenta o desempenho e a disponibilidade das aplicações, por existir mais de uma alternativa para acessar o dado. Com a replicação de dados, todas as cópias de uma tabela partem de um ambiente replicado, pode ser consultada ou mesmo atualizada por uma aplicação. Os SGBDs que formam o ambiente replicado, automaticamente, trabalham para convergir os dados de todas as réplicas de tabelas, garantindo, assim, a consistência e integridade transacional do ambiente.

Deve-se usar este recurso quando se tem a necessidade de replicar um ou mais banco de dados, ou alguns de seus objetos, e de sincronizá-los, de acordo com a demanda do negócio, para descentralizar aplicações e viabilizar arquiteturas e soluções de DataWarehousing<sup>2</sup> e a interação de sistemas heterogêneos.

Apesar da replicação elevar o desempenho de operações de leitura e aumentar a disponibilidade de dados, as transações de atualização incorrem em *overhead*. Controlar atualizações, concorrentes, de várias transações, para se fazer a replicação de dados, é um dispositivo mais complexo e, consequentemente, mais caro do que o controle de um banco de dados centralizado. O DBA ou o administrador do sistema terá trabalho extra para manter sistemas múltiplos diferentes, em vez de um único e grande; além da preocupação com a segurança dos dados, por não estarem centralizados.

### **Tipos de Replicação**

Considerando-se o tempo de atualização de um dado e de sua distribuição, entre as demais réplicas, a replicação é classificada, segundo Date (2004), como síncrona ou assíncrona.

#### **Replicação Síncrona**

No caso da replicação síncrona, todas as cópias de dados serão feitas no instante da sincronização. Se alguma cópia do banco for alterada, essa alteração será imediatamente aplicada a todos os outros bancos da transação. A replicação síncrona é recomendada para aplicações comerciais, em que a consistência exata das informações seja de suma importância.

#### **Replicação Assíncrona**

---

<sup>2</sup> Data Warehouse é o depósito de dados orientado por assunto, integrado, não volátil, variável com o tempo, para apoiar as decisões gerenciais

Na replicação assíncrona, a cópia de todos os dados fica fora de sincronia entre os bancos de dados. Se um banco é alterado, a alteração será propagada e aplicada para outro banco em uma segunda etapa, em uma transação separada; o que poderá ocorrer em segundos, minutos, horas ou até dias após. A cópia poderá ficar fora de sincronia. Contudo quando a sincronização ocorrer, os dados convergirão para todos os locais especificados.

A principal desvantagem, neste tipo de replicação, é que ela eleva, consideravelmente, o consumo de recursos das máquinas envolvidas no momento da replicação.

### **Técnicas de Replicação**

Buretta (1997) especifica que na replicação de dados, as técnicas mais utilizadas são: Replicação mestre/escravo (master / slave) e a Replicação em qualquer lugar.

#### **Replicação Mestre/Escravo**

No modelo mestre/escravo, há uma base de dados, chamada mestre, que pode ser armazenada de forma centralizada ou distribuída. Essa base mestre é a única passível de atualização. As demais réplicas, denominadas escravos, permitem apenas operações de leitura, e são atualizadas por meio de replicação assíncrona.

#### **Replicação em Qualquer Lugar**

No modelo replicação em qualquer lugar, não há o conceito de mestre. A replicação de dados pode ocorrer de forma síncrona, em que as réplicas são atualizadas logo após o término do procedimento de atualização, ou assíncrona em que as réplicas podem ser atualizadas após um determinado tempo, ou após a ocorrência de algum evento específico.

Segundo Buretta (1997), as desvantagens em relação ao modelo anteriormente citado incidiriam sobre a complexidade. Ao se utilizar-se

o modelo, com replicação síncrona, há a necessidade do uso de protocolos para garantir a consistência de dados. Na replicação assíncrona, é necessário utilizar um mecanismo de detecção e resolução de conflitos.

### **Aplicação Exemplo**

Pretende-se apresentar, neste tópico, um cenário, de natureza distribuída, de uma aplicação de banco de dados que torna o uso de banco de dados distribuído, com replicação de dados, uma opção viável, fazendo-se comparação entre consultas realizadas em base centralizada e base distribuída, e relatando-se os resultados com relação ao desempenho e à segurança.

Para implementação, utiliza-se o sistema operacional Linux Mandriva Spring 2008 e o banco de dados MySQL 4.1; ambos instalados em duas máquinas Intel Core 2 Duo 2.2 com 3GB de memória em cada uma.

O MySQL permite a replicação Mestre/Escravo (master/slave). Neste caso, tem-se um servidor MySQL, atuando como Master, e um outro servidor MySQL, atuando como Slave. O servidor Master grava, em um log binário de alterações, todos os comandos de atualização da base de dados. O Slave, por sua vez, conecta-se ao Master, lê o arquivo de log binário e executa os comandos encontrados no log. Dessa forma, as alterações ocorridas no Master são imediatamente replicadas para o outro servidor Slave.

A replicação Master-Slave é um processo assíncrono: poderá haver um atraso de informações entre o Master e o Slave, dependendo do meio de comunicação entre eles. Além disso, como o Slave copia as alterações do Master, todas as modificações dos dados devem ser aplicadas ao Master. Caso contrário, os servidores ficarão sem sincronismo.

Para configurar a replicação, cada servidor MySQL deverá ter um server-id\* único, e o Master deve estar com o log binário habilitado. É necessário também que se crie um usuário no Master, com o privilégio FILE ou REPLICATION SLAVE, para que o Slave possa se conectar a esse servidor e fazer a leitura de seu log binário. No Slave, é preciso indicar o endereço do

servidor Master e o usuário a ser utilizado na conexão.

### **Configuração dos Servidores:**

**IP MASTER:** 192.168.0.199

**IP SLAVE:** 192.168.0.200

Com o MySQL instalado e o respectivo cliente, deve-se copiar o arquivo de configuração do MySQL (my-medium.cnf) para o diretório /etc/, renomeando o arquivo como my.cnf.

Para configurar o servidor Master, é necessário que se interrompa o servidor mysql, uma vez que estará alterando seu arquivo de configuração por meio do comando:

**servicemysql stop**

Deve-se editar o arquivo my.cnf em qualquer editor de texto:

**vi my.cnf**

É necessário acrescentar as seguintes palavras-chave na seção "[mysqld]"

**[mysqld]**

**log-bin #** (Registra todas as consultas que alteram dados em arquivos)

**log**

**log-slow #** (Log de consultas que levam um determinado tempo de execução)

**log-slave-updates #** (Log de atualização para arquivos)

Isso habilitará os logs binários e os demais.

Agora, deve-se iniciar o MySQL:

**servicemysql start**

Após a inicialização do servidor, deve-se digitar:

**\$ mysql -u<usuario> -p<senha>**

Em seguida, cria-se um usuário "slave" para a replicação dos dados, definindo-se uma base de dados para testar a replicação.

**mysql> CREATE DATABASE replica;**

**mysql> GRANT REPLICATION SLAVE ON replica.\* TO slave@'%' IDENTIFIED BY 'slave';**



A base de dados "replica" estará criada e, também, o usuário "slave", com a senha "slave".

O usuário slave terá acesso para replicação, na base de dados replica, respectivamente. O server-id Slave deve ser diferente do server-id do Master, caso contrário o processo de replicação não funcionará.

Para a configuração do servidor slave, deve-se parar o MySQL, editar o my.cnf e, depois, iniciá-lo novamente.

\*Server-id é a identificação de cada servidor MySQL de sua rede.

É necessário que se abra o arquivo my.cnf e acrescente as seguintes linhas na seção [mysqld]:

```
#habilite o log
log
log-bin
master-host=(192.168.0.199)
master-user=slave
ster-password=slave
server-id=2
```

Para finalizar a configuração, é necessário criar um usuário no MySQLclient, com o seguinte privilégio:

```
mysql> GRANT SELECT ON *.* TO read @'%' IDENTIFIED BY 'read';
#(Este é um usuário apenas com direito à leitura em todos os banco de dados).
```

### Testes do Processo de Replicação

Com o servidor e o cliente configurados, no servidor cliente, executa-se o comando:

```
$ mysql -uread -pread
mysql> SLAVE START;
mysql> SHOW SLAVE STATUS;
```

Nos resultados obtidos, por meio do comando acima, procura-se por dois "YES" escritos em Slave I/O e Slave SQL. Caso os YES apareçam, o processo da replicação funcionou.

Agora, lista-se os banco de dados:



```
mysql> SHOW DATABASES;
```

A base "replica" deverá aparecer.

Para os testes e as comparações propostas, criou-se uma tabela no Master:

```
CREATE TABLE `func` (  
id` int(6) unsigned NOT NULL auto_increment,  
nome` varchar(255) NOT NULL default "",  
telefone` int(10) default NULL,  
PRIMARY KEY (`id`)) ;  
INSERT INTO replica VALUES (1);
```

Após a criação da tabela, foi utilizado um script para populá-la com mais de 5 mil registros.

Para testar se a replicação funcionou no servidor Slave, é necessário digitar os seguintes comandos.

```
mysql> USE replica;  
mysql> SELECT * FROM replicacao.teste;
```

Após as configurações e os testes, parte-se para os testes em que se utiliza as consultas SQL, como por exemplo:

```
SELECT * FROM func;  
SELECT * FROM func WHERE nome=('MARIA');  
SELECT * FROM func WHERE `fone` LIKE '99%';  
DELETE FROM func WHERE `fone` LIKE '99%';  
DELETE FROM func WHERE `nome` LIKE 'JO%';  
DELETE * FROM func;  
INSERT INTO func (id,nome,telefone) values ( , ' ', ' ');
```

## RESULTADOS OBTIDOS

Os resultados obtidos foram satisfatórios, como comprovaram os testes realizados com as consultas SQL. Notou-se, com relação ao desempenho, que todas as consultas feitas diretamente no Slave foram milésimos de segundo, mais rápidas, no que diz respeito ao tempo de resposta, do que as mesmas consultas realizadas no Master, por meio de rede. Nas consultas executadas no Master, notou-se também um ligeiro consumo dos recursos da máquina:

Master e Slave. As consultas feitas diretamente no Slave consumiram apenas os recursos da máquina.

Com relação à segurança, se, por uma eventualidade, acontecesse de se perder a base de dados contida em qualquer servidor, por causas desconhecidas, a outra cópia, gravada em outro servidor, estaria pronta para ser restaurada, o que comprova os benefícios do uso da replicação.

## **CONCLUSÃO**

O banco de dados MySql com o sistema operacional Linux apresentaram-se bastante consistentes, interativos e de fácil manipulação, no que se refere à configuração e utilização da replicação de dados. Usou-se uma aplicação pequena, como forma de exemplificar a replicação, e não foram realizados testes em ambientes controlados. Contudo, o exemplo apresentado permite deduzir, a partir dos resultados citados, que a replicação é uma proposta viável, que garante segurança, desempenho e escalabilidade.

Percebem-se várias vantagens, que têm atraído empresas, na utilização de banco de dados distribuídos e na replicação de dados. Dentre elas ressaltam-se o acesso a dados, de maneira eficiente e confiável; a possibilidade de obterem-se informações próximas de onde elas serão utilizadas; o aumento de velocidade no processamento de consultas, além de confiabilidade, transparência e autonomia em um controle distribuído.

Apesar das inúmeras vantagens, existem entraves que impedem o avanço dessa tecnologia: o custo e a complexidade. Referente ao custo no trabalho apresentado, os investimentos não incidiriam sobre a ferramenta ou software. Utilizou-se o sistema operacional Linux e o banco de dados MySQL, ambos ferramentas gratuitas. Os recursos incidiriam na parte física; aquisição de estruturas de redes, hubs, switches, roteadores e hardware. Caso a empresa prefira, há várias outras opções de ferramentas no mercado, como ORACLE e SQLSERVER; o que também geraria ônus para obtenção de suas

respectivas licenças de uso. Com relação à complexidade, um ambiente distribuído é bem mais trabalhoso de se controlar e de se manter do que um ambiente centralizado, em razão das informações estarem distribuídas e fragmentadas e também pelo maior quantitativo de máquinas demandado.

Diferentes aspectos poderão ser abordados em trabalhos futuros, tais como: a realização de testes em ambientes controlados, com a finalidade de medir o desempenho e a aplicabilidade da replicação; e a inclusão de recursos para detecção e tratamento de conflito, por apresentar um tipo de replicação que determina o tratamento de conflitos na própria aplicação.

## REFERÊNCIAS

BURETTA, M. *Data replication: tools and techniques for managing distributed information*. New York: John Wiley e Sons, Inc, 1997. 360p.

DATE, C. J. *Introdução a sistemas de bancos de dados*. 8 ed., Rio de Janeiro: Campus, 2004.

ELMASRI, R & NAVATHE, S. *Sistemas de banco de dados: fundamentos e aplicações* 3ed, SP: LTC, 2000.

IKEMATU, R. *Gestão de metadados: sua evolução na tecnologia da informação*. DataGramaZero – Revista de Ciência da Informação – v.2 n.6 dez/01.

KORTH, H.F, SUDARSHAN S & SILBERSCHATZ, A. *Sistemas de banco de dados*. 3ed, SP: Makron Books, 1999.

MANUAL DE REFERÊNCIA DO MySQL, Disponível em: <http://dev.mysql.com/doc/#manual>

OZSU, M.T; VALDURIEZ, P. *Princípios de sistemas de banco de dados distribuídos*. Tradução da 2ª edição americana, Rio de Janeiro: Campus, 2001.

TANENBAUM, A S. *Redes de computadores*. Rio de Janeiro: Campus, 2000.

## AUTORES

**Anderson de Melo Valadão** é professor da FEIT – UEMG, Campus de Ituiutaba-MG.  
[andersonvalada@gmail.com](mailto:andersonvalada@gmail.com)

**Rodrigo Martins de Faria** é analista de sistemas da FEIT – UEMG, Campus

de Ituiutaba-MG.  
[rodrigomfaria@hotmail.com](mailto:rodrigomfaria@hotmail.com)