



Artigo

# MySQL: Replicação de Dados

Veja neste artigo como configurar replicação no mysql. Veremos o conceito e a prática sobre replicação de dados.

Marcar como lido



Anotar



## INTRODUÇÃO

O objetivo de um mecanismo de replicação de dados é permitir a manutenção de várias cópias idênticas de um mesmo dado em vários servidores de bancos de dados (SGBD). Os principais benefícios da replicação de dados são a redundância, o que torna o sistema tolerante a falhas, a possibilidade de um balanceamento de carga do sistema, já que o acesso pode ser distribuído entre as réplicas, e finalmente, ter-ser o backup online dos dados, já que todas as replicas estariam sincronizadas. Este artigo, apresenta uma introdução ao mecanismo de replicação do MySQL, bem como as configurações básicas para realização desta tarefa.

## VISÃO GERAL DA REPLICAÇÃO

O MySQL permite um tipo de replicação conhecido como Master-Slave,



13



master são imediatamente replicadas para os outros servidores slave.

A replicação no mysql é principalmente compatível com a anterior, isto é, um servidor mais novo pode normalmente ser um escravo de um servidor mais velho sem nenhum problema. Porém, versões mais antigas dos servidores são, freqüentemente, incapazes de servir como slaves de versões mais novas, pois eles não podem entender novas características ou a sintaxe SQL que o servidor mais novo utiliza, e pode haver diferenças no formato dos arquivos que a replicação usa, por exemplo, você não pode replicar de um master MySQL 5.0 para um slave MySQL 4.0.

Com a replicação você possui uma série de vantagens como:

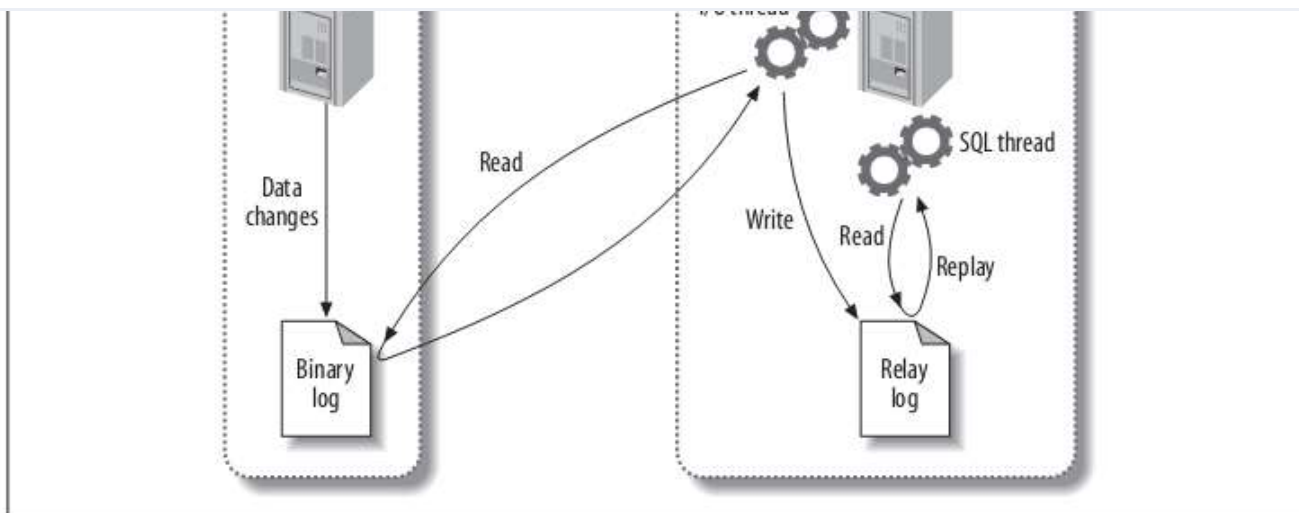
- *Equilíbrio de carga:* a replicação, geralmente não precisa de uma largura de banda muito intensiva, você pode inicia e parar conforme a sua vontade e o slave pode trabalhar distante do master.
- *Distribuição de dados:* a replicação pode ajudar a distribuir as consultas de leitura através de diversos servidores, ou seja, o mesmo dado pode está armazenados em todos os servidores envolvidos da replicação.
- *Backup:* a replicação é uma técnica valiosa para ajudar com backups, porém um slave não é nem um backup nem um substituto para backups.
- *Alta disponibilidade:* se um servidor falhar tem outro servidor para suprir aquele servidor que falhou.

## COMO A REPLICAÇÃO FUNCIONA

O MySQL realiza a replicação em um simples processo de três fases, a Figura 1



13



- O master registra alterações aos seus dados no seu log binário (estes registros são chamados de evento de log binário): antes de cada transação que atualiza dados no master, o mesmo registra as alterações no seu log binário, ou seja, o mysql escreve as transações no seu log binário e logo em seguida o master diz as ferramentas de armazenamento para comitar as transações.
- O slave copia os eventos de log binário do master no seu relay log (log de vigilância): o servidor slave, através de uma thread de I/O, abre uma conexão com o master e logo inicia o processo de esvaziamento de binlog. O esvaziamento de binlog consiste na leitura do evento a partir do log binário do master. A Thread de I/O escreve os eventos no relay log do slave.
- O slave repete aos eventos no relay log: a thread lê e repete eventos a partir do relay log, dessa forma atualizando os dados do slave para combinarem com os do master.

## CONFIGURANDO A REPLICAÇÃO

A configuração do MySQL é bem simples, mas dependendo do cenário pode ocorrer alguma variação. O cenário mais básico é um servidor master e um slave.





- E instruir o slave para conectar ao master e replicar a partir dele também.

*Obs.: Esta configuração foi feita em duas máquinas virtuais onde as mesmas tinham instaladas o sistema operacional GNU/Linux CentOS 5.7 final. Este artigo parte do princípio de que você já tenha instalado o servidor mysql tanto no servidor slave quanto no servidor master.*

### *Criando contas de Replicação*

Primeiramente deve-se criar uma conta de usuário com permissão de replicação para os servidores tanto no master quanto no slave, ou seja, o mesmo usuário criado para os servidores em questão. Aqui está como criar a conta de usuário, que chamaremos de *replicador*:

```
mysql> create user replicador identified by 'replica';  
Query OK, 0 rows affected (0.00 sec)  
mysql> grant replication slave, replication client on *.* to replicador@'192.168.56.  
Query OK, 0 rows affected (0.00 sec)
```

Foi criada esta conta de usuario no servidor master e no slave. Log em seguida foi dado as permissões de replicação (grant replication slave, replication client) para todas as bases de dados do SGBD MySQL (on \*.\* ) para o usuário que acabamos de criar restringindo ao usuário o acesso somente da rede local (replicador@'192.168.56.%' identified by 'replica';).

### *Configurando o Master e o Slave*

Depois de criado as contas o próximo passo é ativar algumas configurações no master. É preciso ativar o log binário e especificar um ID do servidor. Este





```
log-bin = mysql-bin  
server-id = 1 #pode ser qualquer identificador
```

Depois de realizado essa configuração no master é preciso reiniciar o mysql para o servidor ativar as configurações e criar os logs binários. Para verificar se o arquivo de log binário está criado no master execute o seguinte comando “SHOW MASTER STATUS” a Figura 2 mostra o resultado desse comando:

```
mysql> SHOW MASTER STATUS;  
+-----+-----+-----+-----+  
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |  
+-----+-----+-----+-----+  
| mysql-bin.000002 |      333 |              |                  |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

Figura 2 – SHOW MASTER STATUS

*Obs: O exemplo que foi citado acima eu tirei o print da tela depois que eu já tinha feito a configuração e já tinha realizado algumas instruções em sql que gerou esse log binário nessa posição. Suponho que você acabou de instalar o mysql e de configurar o master para replicação quando você digitar esse comando na coluna File irá aparecer **mysql-bin.000001** e na coluna position o número **98**.*

Seguindo no processo de configuração da replicação de dados no mysql, o servidor slave precisa ser configurado pelo arquivo de configuração do mysql (my.cnf), para isso insira as seguintes linhas no arquivo:

```
log-bin = mysql-bin  
server-id = 2  
relay-log = mysql-relay-bin  
log-slave-updates = 1
```



13



a localização e o nome do relay log) e o log-slave-updates (para fazer o slave logar os eventos replicados no seu próprio log binário.

*Obs.: não coloque opções de configuração de replicação como master\_host e master\_port no arquivo my.cnf do slave, pois esta é uma maneira ultrapassada de configurar e ainda pode causar problemas.*

## INICIANDO O SERVIDOR SLAVE

Para finalizar a configuração, o próximo passo é dizer ao slave como conectar-se ao master e começar a replicação. Para isso não utilize o arquivo my.cnf, use a expressão CHANGE MASTER TO. Esta expressão substitui as configurações do my.cnf correspondente e permite que você aponte o slave a um master diferente, sem parar o servidor, segue abaixo a expressão básica para iniciar a replicação:

```
mysql> CHANGE MASTER TO MASTER_HOST='192.168.56.101',  
-> MASTER_USER='replicador',  
-> MASTER_PASSWORD='replica',  
-> MASTER_LOG_FILE='mysql-bin.000001',  
-> MASTER_LOG_POS=0;
```

Esta configuração utilizar o usuário e a senha que foi criado no início da configuração. O MASTER\_HOST é onde colocamos o ip ou nome do servidor master, neste exemplo utilizamos o ip '192.168.56.191'. O MASTER\_LOG\_POS é configurado em 0 porque é o início do log.

Depois de realizada essa configuração, basta só iniciar o slave



13



Pronto! A partir de agora a replicação acaba de ser iniciada. Nessa configuração não é para aparecer erros, para verificar se a replicação está funcionando execute o seguinte comando:

```
mysql> SHOW SLAVE STATUS\G;
```

A Figura 3 mostra o resultado do comando digitado acima:

```
mysql> SHOW SLAVE STATUS\G;
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 192.168.56.101
      Master_User: replicador
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000002
      Read_Master_Log_Pos: 422
      Relay_Log_File: mysql-relay-bin.000007
      Relay_Log_Pos: 559
      Relay_Master_Log_File: mysql-bin.000002
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
*****
```

Figura 3 – SHOW SLAVE STATUS\G;

Note que a replicação está funcionando perfeitamente, portanto a tudo que for feito no servidor master será replicado para o servidor slave.

Finalizando este artigo espero que todos vocês compreendam a mensagem que eu queria passar qualquer coisa é só entrar em contato comigo.

Marcar como lido



Anotar



13



## RECEBA NOSSAS NOVIDADES

Informe o seu e-mail

Receber Newsletter

Suporte ao aluno - Deixe a sua dúvida.



Plataforma para Programadores

Comunidade

Revistas

Baixe o App

Fale conosco

Assinatura Empresarial





