

Replicação em MySQL

Como fazer backup ao vivo das suas bases de dados usando a replicação do MySQL

POR CARLOS EDUARDO AZEVEDO ARANHA DE CASTRO

Muitos DBAs já passaram pela terrível situação de um crash no servidor de banco de dados, seja culpa de um problema de hardware ou uma pane no sistema operacional. Certamente já existe tecnologia de backup hoje em dia, bem mais acessível do que foi há alguns anos atrás, que permite não uma, mas diversas cópias da referida base de dados, mas o que acontece quando nossa cópia não está atualizada? Muitos DBAs preferem fazer cópia de segurança das suas bases de dados durante a madrugada quando nenhum usuário está alterando dados. Se a situação for crítica, fazem de doze em doze horas ou até mesmo seis em seis horas. Nesse meio tempo, a base de dados fica compartilhada entre a aplicação que faz a leitura para backup dos dados e a leitura de dados pelos usuários, tornando ambos os processos mais lentos.

Mas, e se houver uma pane dez minutos antes da tal cópia de segurança começar? A resposta é simples: Caos. Geralmente durante uma pane, os usuários param de consultar ou alterar dados, clientes deixam de ser atendidos, aplicações travam, empresas podem até mesmo parar. E o pior de tudo: A pane pode causar perda de dados, gerando inconsistências, funcionários furiosos por ter de refazer o que já haviam feito e até mesmo prejuízos financeiros.

Felizmente, já existem diversas soluções que contornam esse tipo de situação. Uma delas, que pretendo ilustrar aqui, é fazer uma replicação da base de dados de um servidor para uma outra máquina. Esta solução está disponível no MySQL e é bastante prática.

Requisitos

Para criarmos um ambiente de replicação precisamos de:

- 1) Um servidor MySQL, que passarei a chamar aqui de **Mestre**.
- 2) Uma segunda máquina, que receberá a cópia dos dados do Mestre, que chamarei a partir de agora de **Escravo**. Esta segunda máquina deve possuir também uma cópia instalada do MySQL. Geralmente é usada a estação de trabalho do próprio DBA, no caso da ausência de um segundo servidor.
- 3) Um descarregamento (dump) do banco de dados na qual desejamos fazer cópia de segurança.
- 4) Acesso aos arquivos de configuração da base de dados, no windows my.ini (geralmente fica no diretório da instalação do MySQL) e no linux, dentro do diretório /etc.
- 5) Poder de criação de contas e delegação de privilégios a essa nova conta.

Qualquer que seja o seu ambiente, eu recomendo que as duas máquinas tenham um IP fixo na rede, já que a cópia será feita através da rede. É importante que ambos os MySQLs também sejam da mesma versão. A versão que usei para elaborar este documento foi a versão 4.1.15.

Preparando a Máquina Mestre

Inicialmente, vamos preparar a máquina mestre. O primeiro passo é fazer um dump da(s) base(s) de dados que queremos replicar. Isso serve para criarmos uma cópia exata do nosso banco de dados para colocarmos no servidor Escravo. Esse passo é muito importante!

Para fazer um dump de uma base de dados, use o seguinte comando:

```
S.O. Prompt>mysqldump --database <NOME DA BASE DE DADOS> -u <USUARIO> -p > c:\meu_db.sql
```

Ou para fazer dump de todas as suas bases:

```
S.O. Prompt>mysqldump --all-databases -u <USUARIO> -p > meu_mysql.sql
```

O Sistema irá gerar um arquivo texto, que pode ser bastante grande, dependendo apenas do tamanho de sua base de dados. Se sua(s) base(s) forem grandes, recomendo fazer um dump para cada uma em vez de um grande arquivo contendo todas elas. Dentro desse arquivo texto existem as instruções, em linguagem SQL, para criar e popular sua base de dados em outra máquina.

Recomendo realizar o processo de dump acima ao invés de outras formas de cópia de dados como copiar os arquivos do diretório *data* diretamente através da rede, onde voce pode ter problemas caso use tabelas do tipo InnoDB.

Depois que voce tiver o arquivo SQL do dump da máquina Mestre em mãos, vamos precisar criar o usuário que fará a replicação entre as duas máquinas. Para isso, vamos entrar no mysql com uma conta que permita a criação de privilégios. A conta do root do mysql é uma excelente pedida.

```
mysql> GRANT FILE, SUPER, REPLICATION CLIENT, REPLICATION SLAVE, RELOAD ON *.* TO
'replica'@'%' IDENTIFIED BY 'senhareplica';
```

No exemplo acima, usei como usuário de replicação a conta *'replica'*, e como senha usei a palavra *'senhareplica'*. Modifique esses parâmetros a seu gosto. Um comentário que faço a respeito da linha acima é que ela permite que o usuario *replica* copie os dados de qualquer máquina da rede. Se voce preferir restringir isso, utilize apenas o IP da máquina escravo ao invés do simbolo de '%' usado no exemplo. Um outro detalhe que vale a pena ser observado é que o comprimento máximo da senha é de 16 caracteres.

Agora vamos configurar o arquivo de inicialização do MySQL do servidor Mestre. Abra o arquivo my.ini (windows) ou my.cnf (linux) e localize a seção [mysqld]. No arquivo do linux, já existem informações prontas para uso enquanto que o arquivo do windows é mais enxuto. O importante é acrescentar no final da seção [mysqld] as seguintes linhas:

```
log-bin
server-id = 1
```

Salve o arquivo e reinicie o seu servidor mestre. As linhas servem para iniciar o log binário do mysql e o server-id serve para designar um número único para o seu servidor. Este número não deve se repetir.

Uma vez reiniciado, faça login e digite no servidor mestre:

```
mysql> RESET MASTER;
```

Este comando serve para reiniciar os logs binários já existentes.

Preparando a Máquina Escravo

Agora iremos preparar a máquina escravo. Antes de qualquer passo, devemos restabelecer as bases de dados do servidor mestre na máquina escravo através do dump que fizemos no primeiro passo.

Para isso, usamos o seguinte comando:

```
S.O. prompt> mysql -u <USUÁRIO> -p < c:\meu_arquivo.sql
```

Substitua o nome do arquivo pelo que voce criou na etapa de preparação da máquina Mestre.

Agora vamos alterar o arquivo my.cnf ou my.ini. Novamente localizamos a seção [mysqld] dentro do arquivo. Porém os parâmetros agora são diferentes.

```
server-id = 2
master-host = Endereço IP do servidor MESTRE
master-user = replica
master-password = senhareplica
replicate-do-db = minha_database1
```

Chamo a atenção para a linha `server-id=2`. Repare que aqui no escravo eu usei o numero 2 e no mestre usei o numero 1. Se tivéssemos uma terceira máquina replicando dados, ela deveria ser a `server-id=3`. Estes números não devem se repetir e são usados pelo MySQL se orientar entre as replicações.

Os parâmetros `master-user` e `master-password` são as mesmas das configurações feitas no Mestre. Ou seja, no meu caso, `replica` e `senhareplica`.

O último parâmetro é o comando `replicate-do-db`. Esse parâmetro serve para especificar qual a base de dados que queremos replicar. Se tivéssemos mais bases de dados sendo replicadas, devemos inclui-las da seguinte forma:

```
server-id = 2
master-host = Endereço IP do servidor MESTRE
master-user = replica
master-password = senhareplica
replicate-do-db = minha_database1
replicate-do-db = minha_database2
replicate-do-db = minha_database3
```

Ou, caso aconteça de eu não querer apenas uma base de dados e sim, todas, apenas não especifico a linha `replicate-do-db` e o MySQL entenderá que todas as bases devem ser replicadas.

Após esse passo, vou no prompt do `mysql` da maquina escravo e digito:

```
mysql> RESET SLAVE;
```

Esse comando serve para deletar os logs binários já existentes, para evitar inconsistência com os logs binários já existentes no escravo.

Com isso, nossa configuração de replicação está encerrada. Não devemos nos esquecer de reiniciar o serviço do `mysql` no escravo. Assim que for reiniciado ele tentará se conectar ao mestre, conforme especificado no arquivo `my.cnf` ou `my.ini`.

Para verificar se está tudo bem no seu escravo e se os dados estão realmente sendo replicados, basta usar o comando

```
mysql> SHOW SLAVE STATUS;
```

Repare que na saída do comando será mostrado uma coluna chamada "Slave Running", cujo conteúdo deve ser "yes".

Caso haja algum problema, verifique as permissões da conta `replica`. Ela é geralmente culpada pela maioria dos problemas de replicação. Outra fonte de informação interessante é o log de erro do MySQL, localizado no linux geralmente no diretório `/var/lib/mysql` e no windows no diretório onde ele foi instalado.

Caso voce tenha configurado um endereço IP incorretamente, o escravo obviamente não conseguirá se conectar ao mestre. Para consertar o problema, voce deve reconfigurar o arquivo `my.cnf` ou `my.ini` com o IP correto e reiniciar o escravo.

Conclusão

A replicação no MySQL é algo simples e prático. Gosto sempre de poder contar com a replicação para me fornecer um backup ao vivo do que meus usuários estão fazendo. O fato de eu usar um `mysql` no servidor e outro na minha máquina como DBA, me permite uma flexibilidade muito grande.

Há algum tempo, tive uma pane no meu servidor de banco de dados. Apontei as aplicações existentes para minha máquina, com meu banco replicado e o sistema voltou sem nenhuma perda de dados. Com isso, pude consertar com calma o servidor e depois de corrigido o problema, simplesmente passei a dizer que o escravo era o servidor original e o mestre, minha estação. Os dados foram atualizados rapidamente e pude desfazer a troca mais tarde.

Outro caso que gosto de ilustrar do uso da replicação é o de uso do meu banco de dados de demonstração de sistemas que desenvolvi no meu laptop. Quando estou fazendo atualizações em clientes,

geralmente as testo no meu laptop e depois as transfiro ao servidor de produção. Sempre que crio um campo novo, altero um existente ou insiro novas tuplas, ao chegar em casa, tenho outra máquina configurada como escravo do meu laptop, então sempre que me conecto à rede, tenho um backup instantâneo e automático.

Referências

<http://dev.mysql.com/doc/refman/5.0/en/replication-howto.html>

Carlos Aranha (mysql@carlosaranha.net) – Estudante de engenharia de computação da PUC-Rio, DBA de 2 empresas de engenharia e presto consultoria a outros projetos que usam bancos de dados em Bancos de Dados open source..