Ferramenta nbackup do Firebird 2.0

PAUL VINKENOOG

O *nbackup* é um novo utilitário de backup que vem junto com o Firebird 2.0 e oferece funcionalidades inexistentes no *gbak*, ferramenta de backup pré-existente no Firebird, porém, não substitui esse último. Ambos os programas têm as suas vantagens e desvantagens e irão ainda coexistir provavelmente por algum tempo.

Funcionalidades do nbackup - um resumo

Com o *nbackup*, podemos executar dois grupos diferentes de tarefas:

1. Realizar e restaurar tanto backups completos quanto *incrementais*. Um backup incremental contém somente as alterações ocorridas no banco desde a realização de um backup prévio específico;

2. Bloquear o arquivo do banco de dados principal, o que permite posteriormente realizar o backup, utilizando ferramentas de cópia ou de backup da nossa própria escolha. Nesse caso, o *nbackup* não realiza nenhum backup, apenas cria as condições necessárias para que possamos realizar o backup com segurança. Aqui também existe uma previsão para restauração.

Ambos os modos podem funcionar em um banco de dados ativo, sem interferir com os usuários conectados. O backup criado sempre refletirá o estado do banco de dados no início da operação. Nesse sentido, o *nbackup* não tem diferença com relação ao *gbak*.

Vantagens do nbackup

• *Ambos os modos:* alta velocidade (tão alta quanto o hardware e OS permitam), pois o *nbackup* não enxerga os dados reais. No modo de backup o conteúdo é escrito de forma mais ou menos "cega" no arquivo backup;

• *Modo backup/restore:* economia de tempo e de espaço em disco, pois não precisamos fazer toda vez um backup completo. Essa modalidade pode fazer uma enorme diferença com ban-

Clube83.indb 30 04.04.07 17:38:31

cos de dados com tamanho de gigabytes;

• *Modo bloqueio/desbloqueio (lock/unlock):* liberdade total de escolha das ferramentas de backup, de cópia e/ou de compressão.

Limitações do nbackup

• O *nbackup* não efetuará uma varredura e compactará o banco de dados da maneira como o *gbak* o faz;

• Não poderemos alterar o dono do banco de dados dentro de um ciclo *nbackup* de backup/restauração, tal como faríamos com o *gbak*;

• O *nbackup* não pode construir backups transportáveis (*transportable backups*), isso é: backups passíveis de serem restaurados em uma plataforma incompatível ou sob outra versão de servidor;

• No momento, o *nbackup* não pode ser utilizado em bancos de dados multi-arquivo;

• O *nbackup* consegue suportar apenas bancos de dados locais.

Nas próximas seções, discutiremos extensivamente as várias funcionalidades do *nbackup*.

Realizando e restaurando backups

Em primeiro lugar: o *nbackup.exe* encontra-se no diretório *bin*, na pasta de instalação do Firebird. Assim como na maioria das ferramentas que vem com o Firebird, o *nbackup* não possui interface gráfica, para executá-lo, devemos utilizar a linha de comando (ou a partir de um arquivo lote, *batch file*, ou de uma aplicação).

Realizando backups completos

Para realizar um backup completo do banco de dados, a sintaxe de comando é:

```
nbackup [-U <usuário>-P <senha>]-B 0
  <banco de dados> [<backupfile>]
```

Por exemplo:

```
C:\Databases> nbackup -B 0 inventory.fdb
  inventory_1-Mar-2006.nbk
```

Comentários:

• O parâmetro *-B* significa backup (nossa!). *O nível de backup* 0, indica um backup completo. Backups de níveis maiores do que 0 são utilizados para backups incrementais, discutiremos mais tarde;

• Em lugar de um nome de arquivo do banco de dados, podemos também especificar um alias;

• Em lugar de um nome de arquivo backup, podemos também especificar *stdout*. A especificação dessa opção, direcionará o backup para a saída padrão, de onde poderá ser redirecionada para, por exemplo, um dispositivo de armazenamento em fita ou uma ferramenta de compressão;

• Os parâmetros *-U* (usuário) e *-P* (senha) podem ser omitidos nos seguintes casos:

- Quando fizermos o login como super-usuário (root, administrador.) ou
- Quando as variáveis de ambiente ISC_USER e ISC_PASSWORD estiverem configuradas.

Para maior clareza e brevidade, esses parâmetros não serão utilizados nos exemplos;

• Os parâmetros diferentes (*-B*, *-U* e *-P*) poderão ocorrer em qualquer ordem. Naturalmente, cada parâmetro deve ser imediatamente seguido pelo(s) seu(s) próprio(s) argumento(s). No caso de *–B* existem três deles: nível de backup, banco de dados e arquivo backup, nessa ordem;

• Caso o parâmetro *-B* for especificado por último, poderemos omitir o nome do arquivo backup. Nesse caso, o *nbackup* gerará um nome de arquivo baseado no nome do banco de dados, nível de backup e a data e hora atuais. Isso poderá resultar em um conflito de nomes (e uma falha do backup), caso dois comandos de backup do mesmo nível, forem emitidos simultaneamente.

Importante: *Não utilizar o **nbackup** com bancos de dados multi-arquivo, pois poderá resultar em corrupção e perda de dados, apesar do fato de que o **nbackup** não alertará a respeito de tal comando.*

Algumas observações com relação ao funcionamento interno

O que vem a seguir não representa o conhecimento mínimo necessário para a utilização do *nbackup*, apenas fornece uma idéia básica do que acontece por trás dos panos durante a execução do *nbackup -B*:

1. Em primeiro lugar, o arquivo do banco de dados principal é bloqueado mediante a alteração de um *flag* de estado interno. A partir desse momento, qualquer mudança no banco de dados será escrita em um arquivo temporário, o arquivo diferencial ou arquivo delta;

Clube83.indb 31 04.04.07 17:38:34

2. A seguir, o backup real é construído. Esse backup não é uma cópia de arquivo que possa ser utilizada diretamente, a restauração deve ser realizada também pelo *nbackup*;

3. Assim que o backup tenha sido completado, o conteúdo do arquivo delta estará integrado ao arquivo de banco de dados principal. Nesse momento, o banco de dados é desbloqueado (o *flag* volta para normal) e o delta é removido.

As funcionalidades dos passos 1 e 3 são fornecidas por duas novas cláusulas SQL: ALTER DATABASE BEGIN BACKUP e ALTER DATABASE END BACKUP. Ao contrário do que os nomes sugerem essas cláusulas não realizam o backup real, na verdade, apenas criam as condições para que o backup do arquivo do banco de dados principal possa ser realizado com segurança.

Ainda mais: não precisamos mesmo emitir esses comandos; o *nbackup* o fará automaticamente, no momento apropriado.

Restauração de um backup completo

Um backup completo é restaurado da seguinte maneira:

```
nbackup [-U <usuário>-P <senha>]-R  <banco de dados> [<backupfile>]
```

Por exemplo:

```
C:\Databases> nbackup -R inventory.fdb inventory_1-Mar-2006.nbk
```

Comentários:

• Não especificamos um nível para a restauração;

• Na restauração, o parâmetro *-R* **deve** vir último, por motivos que se tornarão claros posteriormente;

• Se o banco de dados especificado já existir e não houver nenhuma conexão, o mesmo será sobrescrito sem qualquer mensagem de alerta. Se houver usuários conectados, a restauração falhará e receberemos uma mensagem de erro;

• Note também, que poderemos omitir o nome do arquivo de backup. Caso o omitirmos, o *nbackup* o solicitará.

Realizando backups incrementais

Para realizarmos um backup incremental (diferencial), especificamos um nível de backup maior do que 0. Um backup incremental de nível *N*, sempre contém as modificações do banco de dados desde o nível de backup mais recente *N -1*. Exemplos:

Um dia depois do backup completo (nível 0), realizamos outro de nível 1:

```
C:\Databases> nbackup -B 1 inventory.fdb
  inventory_2-Mar-2006.nbk
```

Esse backup só conterá as modificações do dia anterior. No dia seguinte, fazemos novamente um outro de nível 1:

```
C:\Databases> nbackup -B 1 inventory.fdb
  inventory_3-Mar-2006.nbk
```

Esse backup conterá as modificações dos dois dias anteriores desde o backup completo e não somente as do backup de nível 1 prévio. Duas horas depois, realizamos um backup de nível 2:

```
C:\Databases> nbackup -B 2 inventory.fdb
  inventory_3-Mar-2006_2.nbk
```

Esse backup mais recente conterá somente as modificações desde o backup de nível 1 mais recente, isso é: das últimas duas horas.

Obs.: *Todos os comentários a respeito dos backups completos se aplicam também aos backups incrementais. Novamente: não utilize o* ***nbackup*** *com bancos de dados multi-arquivo.*

Restauração de backups incrementais

Para restaurarmos backups incrementais, devemos especificar a cadeia completa de arquivos de backup, desde o nível 0, até aquele que se deseja restaurar. O banco de dados será sempre restaurado desde o começo, passo a passo (é essa adição gradual até que o banco de dados esteja restaurado, que deu a origem à denominação backup incremental).

A sintaxe formal é:

```
nbackup [-U <user> -P <password>] -R
  <database> [<backup0> [<backup1> [...]]]
```

Portanto, para restaurar o backup de nível 2 do exemplo anterior, procederemos da seguinte maneira:

```
C:\Databases> nbackup -R inventory.fdb
  inventory_1-Mar-2007.nbk
  inventory_3-Mar-2007.nbk
  inventory_3-Mar-2007_2.nbk
```

Naturalmente, a linha foi dividida aqui apenas por razões de *layout*, na verdade escrevemos o comando completo e só pressionamos ENTER ao final do mesmo. Comentários (adicionais aos

Clube83.indb 32 04.04.07 17:38:35

comentários relativos à restauração de um backup completo):

• Como não sabemos antecipadamente quantos nomes de arquivo virão após a chave *-R* (já que não especificamos um nível ao restaurar), o *nbackup* considera que todos os argumentos após o *-R* são nomes de arquivos de backup. Esse é o motivo porque nenhum outro parâmetro (*-U* ou *-P*) pode vir após a lista de nomes de arquivo;

• Não existe nenhum limite formal quanto ao número de níveis de backup, porém, na prática, raramente fará sentido ultrapassar os níveis 3 ou 4.

Links não encadeados

O que aconteceria se acidentalmente omitirmos um arquivo ou especificarmos uma série de arquivos que não pertence a um conjunto específico? Imaginemos que acidentalmente especifiquemos *inventory_2-Mar-2007.nbk* em vez de *inventory_3-Mar-2007.nbk* no exemplo anterior.

Ambos são arquivos de backup de nível 1, portanto, em ambos os casos, obteremos séries corretas de nível "0, 1, 2". No entanto, o nosso arquivo de nível 2 é incremental com relação ao backup de nível 1 de 3 de Março e não para o do dia 2 de Março.

Tal erro nunca nos conduzirá a um banco de dados incorretamente restaurado. Cada arquivo de backup tem o seu ID único. Além disso, cada arquivo backup de nível 1 ou superior, contém o ID do backup que foi baseado.

Ao acionar a restauração, o *nbackup* verifica esses IDs e caso a cadeia esteja interrompida em algum lugar, a operação é abortada e recebemos uma mensagem de erro.

Uma aplicação prática

Um esquema de backup incremental baseado em *nbackup* poderia ter os seguintes passos:

• Cada mês, um backup completo (de nível 0) é construído;
• Cada semana, um backup de nível 1;
• Diariamente, um backup de nível 2;
• De hora em hora, um backup de nível 3.

Caso todos os backups tenham sido conservados, poderemos restaurar o banco de dados para o seu estado original, em qualquer hora do passado. Para cada ação de restauração, um máximo de quatro arquivos de backup será utilizado.

Naturalmente, planejaremos de tal maneira que os backups mais grandes, que consomem muito tempo, sejam construídos em períodos fora das horas de pico. Nesse caso, os backups de níveis 0 e 1, podem ser construídos nos fins de semana e os de nível 2 à noite.

Se não quisermos guardar os backups para sempre, poderemos acrescentar um horário para sua eliminação:

• Backups de Nível 3 são eliminados depois de 8 dias;
• Backups de Nível 2 depois de um mês;
• Backups de Nível 1 depois de seis meses;
• Backups completos depois de dois anos, porém, o primeiro de cada ano deverá ser guardado.

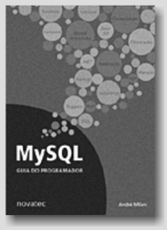
Naturalmente, isso é apenas um exemplo. O procedimento mais adequado a ser adotado, dependerá da aplicação, do tamanho do banco de dados, da sua atividade etc.

Vamos em frente?

Nesse ponto, já sabemos tudo o que é necessário para realizar e restaurar backups completos e/ou incrementais com o *nbackup*. Caso quisermos utilizar ferramentas da nossa própria escolha para realizar o backup do banco de dados Firebird, devemos continuar a leitura. Se esse não for o caso, então boa sorte ao trabalhar com o *nbackup*!

Desconto Especial!

Lançamento



MySQL
Guia do Programador
Novatec
R$ 74,00

UML Essential
3ª ed.
Bookman
R$ 61,00

Dominando Netbeans
Construa aplicativos Java
tanto em Desktop, como para Web
Ciência Moderna
R$ 76,00

Dominando Firebird
Uma referência para
Desenvolvedores de Bancos de Dados
Ciência Moderna
R$ 179,00

MySQL 5 Interativo
Guia Básico de Orientação
e Desenvolvimento
Érica
R$ 89,00

COMPRE JÁ e garanta um DESCONTO ESPECIAL!
Utilize o código de cupom LDP02 quando solicitado.*
* Válido até 30/04/2007

Frete Grátis
Para todo o Brasil!

Até 6x sem juros no
Visa, Mastercard e Hipercard.*
*Parcela mínima de R$ 20,00

Outras facilidades de pagamento:

Tel: (11) 4062-5152

livrosdeprogramacao.com.br

Bloqueio e desbloqueio

Caso preferirmos utilizar ferramentas de backup da nossa própria escolha ou apenas realizar uma simples cópia de arquivo, devemos utilizar o modo de bloqueio/desbloqueio do *nbackup's*. Bloqueio aqui quer dizer que o arquivo do banco de dados principal é congelado temporariamente, isso não quer dizer que não poderá haver nenhuma modificação no banco de dados.

Tal como no modo de backup, as modificações são dirigidas para um arquivo delta temporário e após o desbloqueio, o arquivo delta será fundido com o arquivo principal. Não existe nenhuma interface gráfica; o *nbackup* deve ser acionado a partir da linha de comando ou a partir de um arquivo de lote ou de uma aplicação.

Bloqueando o banco de dados e realizando o backup

Uma sessão típica onde faríamos o backup, teria a seguinte sequência de passos:

1. Bloqueamos o banco de dados com a chave *-L* (lock - bloqueio):

```
nbackup [-U <user> -P <password>] -L <database>
```

2. Agora copiamos/realizamos o backup/efetuamos a compactação (ZIP) do arquivo do banco de dados, escolhendo uma ferramenta (uma simples cópia de arquivo também é possível).

3. Desbloqueamos o banco de dados com a chave *-N* (desbloqueio):

```
nbackup [-U <user> -P <password>] -N <database>
```

O último comando fará também com que qualquer modificação que tenha sido gravada no arquivo delta, seja fundida com o arquivo principal. O backup assim construído, contém os dados do modo como estavam no momento em que o banco de dados foi bloqueado, independentemente de quanto tempo esse estado de bloqueio durou e independentemente de quanto tempo aguardamos antes de realizar o backup real.

> **Obs.:** *Tudo o que foi dito com relação à realização de backup/restore, também se aplica às chaves de bloqueio/desbloqueio: não devemos utilizá-las com bancos de dados multi-arquivo. Enquanto não houver mudanças nesse sentido, não devemos deixar em absoluto o* ***nbackup*** *solto em bancos de dados multi-arquivo.*

Restauração de um backup construído após o "nbackup -L"

Uma cópia de um banco de dados bloqueado é um banco de dados bloqueado também, portanto, não podemos apenas copiá-lo e começar a utilizá-lo. Caso o banco de dados original for perdido ou for danificado e precisarmos restaurar a cópia construída (ou caso desejarmos instalar a cópia em outra máquina), devemos proceder como a seguir:

1. Copiar/restaurar/abrir o arquivo ZIP do banco de dados realizado com as correspondentes ferramentas.

2. Agora, desbloquear o banco de dados, não com a chave *-N*, porém, com a chave *-F* (*fixup*):

```
nbackup -F <database>
```

Com que finalidade existem duas chave de desbloqueio *-N* e *-F*?

- *-N* se certifica que quaisquer modificações realizadas desde o bloqueio mediante a chave *–L*, sejam fundidas com o arquivo do banco de dados principal;
- Depois que o banco de dados retorna ao modo leitura/escrita normal e o arquivo temporário tenha sido eliminado;
- *-F* apenas altera o *flag* de estado do banco de dados auto-restaurado para normal;

Portanto, utilizamos:

- *-N* depois de termos construído uma cópia/backup (para reverter o *-L* anteriormente acionado);
- *-F* depois de ter restaurado o backup.

> **Obs.:** *ter chamado a chave* **-F** *de* **Fixup** *(consertar), não foi uma escolha das mais felizes. No fim das contas, não conserta nada, somente desbloqueia o banco de dados. O* **flag -N** *(desbloquear), por outro lado, não só realiza o desbloqueio, como também um conserto (***fixup***). No entanto, teremos que conviver com isso.*

Por baixo dos panos

Essa seção não acrescenta nenhum conhecimento indispensável, porém, fornece algumas informações extras para nos aprofundarmos na compreensão das várias chaves. O *nbackup -L* faz o seguinte:

Conecta ao banco de dados, inicia uma transação, chama ALTER DATABASE BEGIN BACKUP, efetua o *commit* da transação e desconecta do banco de dados. O *nbackup -N* segue os mesmos passos, porém, utilizando END BACKUP. O *nbackup -F* funciona da seguinte maneira:

O arquivo do banco de dados restaurado é aberto, dentro do arquivo, o *flag* de estado é modificado de bloqueado (*nbak_state_stalled*) para normal (*nbak_state_normal*) e após o arquivo é fechado novamente.

> **Obs.:** *O nbackup -F funciona em um nível de arquivo puro e pode, portanto, ser executado sem o servidor Firebird rodando. Quaisquer parâmetros -U ou -P acrescentados ao comando serão completamente ignorados.*

Conclusão

Vimos neste artigo como funciona o utilitário nbackup para criação de backups, inclusive incrementais, no Firebird 2.0. Vimos as várias opções disponíveis na linha de comando, além de detalhes internos de funcionamento. ■

Clube83.indb 34 04.04.07 17:38:37