

用数据管道优化 Adaptive Server Anywhere 数据库 Ways to Optimize Adaptive Server Anywhere Database by Using Data Pipeline

黄成荣

Huang Chengrong

(南宁师范高等专科学校数学与计算机科学系 龙州 532400)

(Dept. of Math. & Comp. Sci., Nanning Teacher's College, Longzhou, 532400)

摘要 介绍 PowerBuilder 的数据管道工具及其对 Adaptive Server Anywhere 数据库的优化方法。通过重新组织 Adaptive Server Anywhere 数据库内部的数据,来达到清除数据碎片、减少其体积的目的,从而优化了 Adaptive Server Anywhere 数据库,提高了数据库的性能和运行效率。

关键词 PowerBuilder 数据管道 数据库 优化

中图分类号 TP311.11

Abstract The function of the Data Pipeline of PowerBuilder, its extended attributes and optimization way are introduced. The circulating speed of the Adaptive Server Anywhere Database are optimized by using the pipeline to reorganize the data inside, clear the data fragments and reduce the spaces they took.

Key words PowerBuilder, Data Pipeline, database, optimization

PowerBuilder 应用程序所使用的后台数据库大多是 PowerBuilder 自带的 Adaptive Server Anywhere 数据库。Adaptive Server Anywhere 数据库文件有一个特点:其体积只增加不减少,而且在删除记录后,体积反而增大。在使用过程中,由于数据的增删,数据库文件的体积会越来越大,数据库内部产生较多数据碎片,从而降低系统的读写速度,影响了应用程序的性能。针对这种情况,PowerBuilder 提供了 3 种操作和管理数据库的工具:数据库画笔、查询、数据管道。本文讨论数据管道实现对数据库的优化的技术。

1 数据管道

数据管道是 PowerBuilder 操作和管理数据库的工具,它向用户提供了在数据库内部、数据库之间、甚至数据库管理系统之间以用户表的形式进行表结构和数据的传输,是一种快速复制数据的简便途径。它有 2 种使用方法,一种是在数据管道画笔中设计数据库管道并执行,

另一种是建立数据管道对象并保存起来，在程序运行期间使用，PowerBuilder 将按照已经定义好的数据管道向目的数据库灌入数据。

数据管道虽然可以复制源数据库中列的编辑风格、显示格式和有效性规则以及表的主键，但是，表的索引、外键是不能被复制的。另外，数据管道将数据灌入到目的数据库后，均以表的形式存在。也就是说，源数据库中的表和视图，通过数据管道灌入到目的数据库后都是表，不存在视图，除非用户亲自建立。

PowerBuilder 提供了 5 种管道操作方式：(1) Create-Add Table；(2) Replace-Drop/Add Table；(3) Refresh-Delete/Insert Rows；(4) Append-Insert Rows；(5) Update-update/Insert Rows，其中第 (3)、(4)、(5) 种操作方式要求目的数据库中的目的表必须存在，如果目的表不存在，则数据管道操作失败。

2 数据管道对 Adaptive Server Anywhere 数据库的优化

数据管道优化 Adaptive Server Anywhere 数据库的基本思路是：通过重新组织数据库内部的数据，以达到清除数据碎片、减小体积、提高性能的目的。首先将源数据库的所有表的数据全部传输到一个新的、没有数据碎片的目的数据库中，然后再将目的数据库文件替换成源数据库文件。为安全起见，在运行数据管道对象前先将源数据库文件备份好。下面以 PowerBuilder 8.0 为例说明优化 Adaptive Server Anywhere 7.0 数据库的具体方法。

2.1 创建目的数据库

用 PowerBuilder 的数据库画笔新建 1 个数据库时，默认生成 2 个文件：.db 文件和 .log 文件。其中 .db 文件是用来存放数据库信息，包括表的结构、视图、数据等；而 .log 文件是数据库的日志文件，用来记录用户每一次对数据库有影响的操作，例如创建或删除表、视图、触发器等对象，插入、删除、修改表中的数据等。使用数据管道从一个数据库向另一个数据库灌入数据时，需要连接 2 个数据库，为了确保 2 个数据库互换后也连接成功，在 Create Adaptive Server Anywhere Database 对话框中，不选“Use Transaction Log”复选框，以便不建立数据库的 .log 文件，将数据库命名为 destination.db，这个数据库就是目的数据库。源数据库中含有的表，均在目的数据库中建立相同结构的表；源数据库中含有的视图，也要在目的数据库中建立相同结构的视图；源数据库中含有的表的索引、外键，在目的数据库中建立相同的表的索引、外键。这样目的数据库与源数据库有相同的表、视图、主键、索引、外键，唯一不同的是目的数据库中的每一个表和视图不含有任何记录，把这个目的数据库作一个名为 destinationbak.db 的备份，将它和 destination.db 放在源数据库所在目录上。

2.2 创建数据管道对象

为了优化源数据库，必须将源数据库中的每个表的记录灌入到目的数据库中。为此，要为每个表建立相应的数据管道对象。对于含有外键的表、其外键所连接的表必须先于该表灌入到目的数据库中。在 Data Pipeline 工作区的 Option 下拉列表框中，选择 Refresh-Delete/Insert Rows 选项，将每个表所对应的数据管道对象依次命名为 p_1 、 p_2 、 p_3 、 \dots 、 p_n (n 为源数据库的表的数量)。

2.3 创建用户对象

在应用程序中使用数据管道对象时，需要使用用户对象画笔通过继承 PowerBuilder 中系统的数管道对象 Pipeline，定制出适合用户特殊需要的非可视标准类用户对象，然后再对该

的事件编写事件处理程序。创建非可视标准类用户对象的步骤如下：

(1) 单击 PowerBar1 上的 New 按钮，在弹出的对话框中单击 PB Object 选卡，并且选择该选卡中的 Standard Class 图标。

(2) 单击 OK 按钮，在打开的 Select Standard Class Type 对话框中选择 Pipeline 图标。

(3) 单击 OK 按钮，打开用户对象画笔。

(4) 为该用户对象定义 3 个单行编辑框实例变量，分别用来显示数据管道的读、写、出错行数：

```
SingleLineEdit sle_read, sle_write, sle_error
```

(5) 给用户对象的单行编辑框文本赋值。系统的数据管道对象 Pipeline 有个 pipemeter 事件，该事件在数据管道每处理完一块数据后触发一次，数据块的大小由数据管道对象的 commit 决定。在 pipemeter 事件中添加如下脚本：

```
sle_read.text=string (RowsRead)
sle_write.text=string (RowsWritten)
sle_error.text=string (RowsInError)
```

其中，RowsRead、RowsWritten、RowsInError 是系统数据管道对象 Pipeline 的属性。这 3 个属性都是长整型，第 1 个表示数据管道在执行过程中在源数据库表中读取了多少条记录；第 2 个表示管道在执行过程中在目的数据库表中写入了多少条记录；第 3 个表示管道在执行过程中发生错误的记录数。

(6) 保存该用户对象，命名为“u_pipeline”。

2.4 创建窗口和添加控件

表 1 添加控件列表

(1) 创建一个窗口对象，命名为“w_pipeline”。

(2) 在该窗口上面添加控件如表 1 所示。

| 控件名 | Name | Text | 用途 |
|----------------|-----------|------|---------------|
| SingleLineEdit | sle_read | 0 | 显示读取的数据行数 |
| | sle_write | 0 | 显示写入的数据行数 |
| | sle_error | 0 | 显示出错的数据行数 |
| CommandButton | cb_start | 启动 | 启动数据管道优化数据库 |
| | cb_cancel | 中止 | 中止数据管道对某个表的操作 |
| | cb_reset | 取消 | 恢复优化前源数据库的状态 |
| Datawindow | cb_close | 退出 | 退出程序 |
| | dw_error | | 显示数据管道错误 |

2.5 编写脚本

(1) 为 w_pipeline 窗口声明实例变量

```
transaction tr_source
transaction tr_destination
u_pipeline i_upipeline
```

(2) 在应用程序的“Global External Functions...”中声明两个 API 函数：

```
FUNCTION ulong CopyFileA (ref string lpExistingFileName, ref string lpNewFileName,
ulong bFailIfExists) Library " Kernel32.dll"
```

```
FUNCTION ulong SetFileAttributcsA (ref string lpFileName, ulong dwFileAttributes)
Library " Kernel32.dll"
```

(3) 为 w_pipeline 窗口定义函数

定义 2 个窗口函数：wf_connectdb (string ls_profile, ref transaction tr_trans) return integer 和 wf_disconnectdb (ref transaction tr_trans) return none 分别用于实现与数据库的连接与断开。

(4) 在 w_pipeline 窗口的 open 事件中添加脚本: disconnect。

(5) 在 w_pipeline 窗口的 close 事件中添加脚本清除已声明的实例变量, 并连接数据库。

(6) 为 cd_start 按钮的 clicked 事件编写脚本, 按顺序完成下列功能: 1) 备份源数据库; 2) 将目的数据库的备份替换目的数据库; 3) 连接目的数据库和源数据库; 4) 用数据管道函数 start () 向目的数据库灌入数据; 5) 断开与目的数据库和源数据库的连接; 6) 用目的数据库替换源数据库。

(7) 用数据管道函数 cancel () 为 cb_cancel 按钮的 clicked 事件编写脚本以中止管道操作。

(8) 为 cb_reset 按钮的 clicked 事件编写脚本, 将源数据库的备份替换源数据库。

(9) 在 cb_close 按钮的 clicked 事件中添加脚本: close (w_pipeline)。

2.6 数据库的优化

把 w_pipeline 窗口添加到应用程序中, 在使用时打开该窗口, 单击“启动”按钮运行数据管道对象, 即可实现对 Adaptive Server Anywhere 数据库的优化。表 2 所示是 1 个数据库优化前后的体积及打开 2 个数据窗口控件所关联的数据窗口对象检索数据所需时间的比较。可以看出, 利用数据管道对数据库的优化作用明显。

表 2 数据库优化前后性能比较

| | 数据库 文件体积 (KB) | 打开含 66510 个记录的数据 窗口所用时间 (s) | 打开含 2239 个 记录的数据窗 口所用时间 (s) |
|-----|---------------------|--------------------------------------|--------------------------------------|
| 优化前 | 27427 | 33 | 2 |
| 优化后 | 14260 | 8 | <1 |

3 结束语

PowerBuilder 的数据管道是操作和管理数据库的优秀工具, 用数据管道对 Adaptive Server

Anywhere 数据库内部数据重新组织, 清除数据库文件内部的数据碎片, 减少其体积, 从而实现对数据库的优化。数据管道优化后的数据库文件体积变小, 对数据库的读写速度明显提高, 从而提高数据库运行的效率。

参考文献

- 1 张长富, 李 匀. PowerBuilder 8 参考手册. 北京: 北京希望电子出版社, 2002. 120~145.
- 2 丁 斌. PowerBuilder 6.5/7.0 与多媒体程序设计. 北京: 北京航空航天大学出版社, 2000. 82~87.
- 3 刘宏岩, 何 军等. PowerBuilder 原理与应用指南. 北京: 电子工业出版社, 1999. 661~676.

(责任编辑: 黎贞崇)