

在PB中实现MS-SQL Server中的功能

摘要: 本文详细阐述了在PB中如何实现后台数据库中用户活动及其各对象加锁状态的实时监测, 以及提供了在PB中实现数据备份和恢复的解决方法。本文的实现技术对于在PB中实现DBMS中的另外一些系统功能是值得借鉴的。

关键词: PowerBuilder SQL Server 监测

Implementing of MS-SQL Server functions in PB

段江南 蔡兆麟 (华中科技大学主校区动力楼 314 室 430074)

1 引言

PowerBuilder (简称PB) 作为一个强大的基于客户/服务器 (Client/Server) 体系结构应用系统的开发工具而得到广泛的使用。开发者利用PB可以快速构建界面友好的应用程序。作为一个专业的数据库开发工具, PowerBuilder几乎与当今所有的主流数据库管理系统 (DBMS) 都有高性能的专用接口。本文详细介绍如何在PB开发的前端应用程序中实现对后台数据库 (MS-SQL Server 7.0) 操作活动的实时监测以及备份恢复数据的这些通常只有数据库管理系统才有的功能。

在一般的网络数据库应用系统中, 都会牵涉到多用户联机操作问题, 因此系统管理员对后台数据库的监控是必要的, 它有助于保证整个系统安全稳定运行。另外在应用数据库系统中, 数据备份和恢复也是一种不可缺少的重要功能。当然在数据库的DBMS环境下, 我们可以直接调用数据库本身提供的图

形化用户界面或命令来完成这些功能。例如在SQL Server下, 利用SQL SERVER控制台中的Current Activity功能, 监测用户应用程序操作数据库的各个Process的状态以及当前数据库中各个对象的加锁状况等等, 至于数据库的备份和恢复这些功能是每个数据库管理系统 (DBMS) 的标准功能。但是在实际应用中, 许多应用程序系统的使用人员还不能在DBMS用户界面直接操作来完成这些功能, 所以有必要在PB应用程序中编程实现这些功能, 给用户一个友好和易于理解的统一界面。笔者在开发长沙鼓风机厂管理信息系统 (CG-MRPII) 的维护子系统中, 实现了这些技术。

2 主要思路及其关键技术

实际上我们在SQL Server的控制台中所做的很多操作都可以利用SQL Server所提供的系统存储过程来实现。例如在MS SQL Server的查询分析器中, 我们可以利用sp-attach-db这

个系统存储过程将已有的数据库文件指定一个数据库名后挂接到Server中。同样我们可以利用SQL SERVER中的sp-who和sp-lock两个系统存储过程来分别实现监测SERVER中登录用户的活动和数据库中的各对象加锁的状态。在PB中建立数据源 (Data Source) 分别为这两个系统存储过程的数据窗口, 这样数据窗口就能检索出应用程序所连接的数据库服务器中当前所有活动的信息。

至于实现对数据的备份和恢复功能时, 我们可以利用BCP这个实用程序。BCP实用程序用来在一个已有的数据库表和一个用户可以指定的格式的操作系统文件之间传送数据。具体格式如下 (该命令格式很复杂, 下面只介绍本文中用到的格式):

(1) 数据备份。BCP 数据库名..表名 out 备份文件路径 -c -S 数据库服务器名 -U 用户名 -P 口令

(2) 数据恢复。BCP 数据库名..表名 in

Implementing of MS-SQL Server functions in PB

备份文件路径 -c -S 数据库服务器名 -U 用户名 -P 口令

例如: 要备份名为 xs-ht 表中的数据, 该表存在于 DBSERVER 数据库服务器中的 cg-new 数据库中, 且数据库用户名为 sa 的密码为 sa, 则命令为:

```
BCP cg-new..xs-ht out c:\data\databkup.dat -c -S DBSERVER -U sa -P sa
```

这样就可以把数据库服务器上该表的数据远程备份到执行该命令的计算机中。所以我们可以先从数据库中的系统表——sysobject 中检索出该数据库中所有的表名, 然后用程序按上述格式生成一个 DOS 下的批处理命令文件, 再在 PB 程序中运用 RUN() 函数来运行这个批处理命令, 就可以使用户在 PB 程序中对数据库中的数据进行备份和恢复操作。

3 实现技术的步骤和程序

3.1 数据库当前活动的监测

(1) 首先建立数据窗口, 在建立新数据窗口时选数据源 (Data Source) 为: Stored Procedure, 数据窗口风格为: Grid, 确定后, 在随后界面中将 System Procedure 选项选中, 然后从下面的列表中选择 sp-who 这项 (见图 1), 确定后将生成一个新数据窗口, 将表

头根据各个字段的实际意义改为中文即可。重复上述步骤, 选 sp-lock 系统存储过程。数据窗口中各字段意义在此不详细说明。从图 2 中可以看出运用 sp-who 系统存储过程可以监测各用户进程的实时状态。

(2) 后面就是一般的数据窗口检索命令的编程了, 即在图 2 中的刷新按钮下编写数据窗口检索的脚本即可。

另外从图 2 可知, 运用该方法已经实现了库锁监控、登录信息、用户管理、用户权限功能。

3.2 数据的备份和恢复

(1) 首先建立一个数据查询条件, SELECT sysobjects.name FROM sysobjects WHERE sysobjects.type = 'U' 的数据窗口, 用来列出所连接数据库中所有表的清单, 以便为后面在程序中根据 BCP 命令的格式动态生成批处理文件提供表名清单。

(2) 数据的备份, 建立一个如图 3 一样的用户界面, 在备份按钮中的脚本如下。

```
// 变量声明 long ll-Rows, ll-index  
string ls-batfile, ls-name  
string ls-path  
integer li-FileNum  
string ls-taskfile  
datastore ids-table
```

```
transaction ids-backup
```

```
// 判断用户输入参数是否完整
```

```
ls-path=""
```

```
ls-path=sle-path-b.text
```

```
if ls-path="" then
```

```
messagebox ("错误", "请选择备份数据存放目录!", stopsign!)
```

```
return
```

```
end if
```

```
// 生成备份时连接数据库的事务对象
```

```
ids-backup=create transaction
```

```
// 设置事务对象参数
```

```
ids-backup.DBMS = "MSS Microsoft SQL Server 6.x"
```

```
ids-backup.Database = sle-db-b.text
```

```
ids-backup.LogPass = sle-pwd-b.text
```

```
ids-backup.ServerName = sle-server-b.text
```

```
ids-backup.LogId = sle-user-b.text
```

```
// 用自定义的事务对象尝试连接数据库, 判断用户参数是否正确, 已保证后面的
```

```
// 批处理命令的正确性以及用来检索该数据库中的所有表。
```

```
connect using ids-backup;
```

```
if ids-backup.sqlcode<>0 then
```

```
messagebox ("错误", "数据库连接参数错误, 不能连接指定的数据库!", stopsign!)
```

```
return
```

```
end if
```

```
// 播放备份过程中的动画
```

```
uo-1.of-playavi ("bmp\download.avi")
```

```
w-backup.height=1148
```

```
// 从数据存储中检索该数据库中的所有表名清单
```

```
ids-table=create datastore
```

```
ids-table.dataobject=" d-table-list"
```

```
ids-table.settransobject (ids-backup)
```

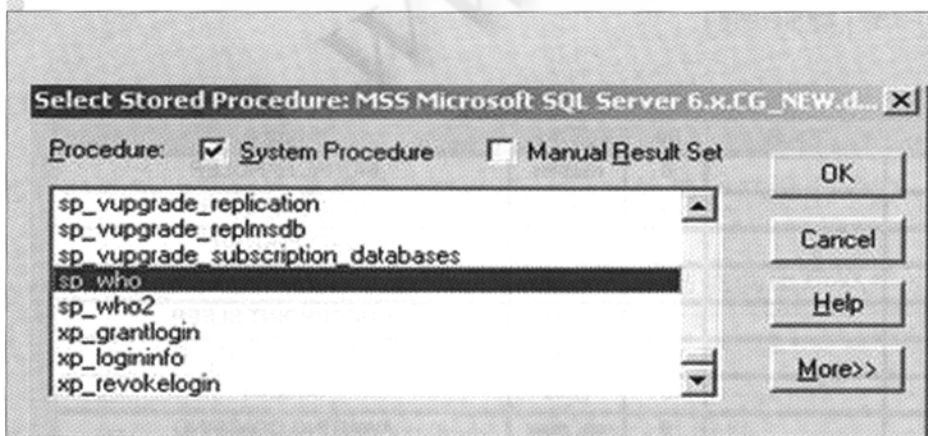


图 1

Practical Experience

```

ll-Rows=ids-table.retrieve()
// 根据表名清单生成数据备份批处理命令的内
容
ls-batfile=""
for ll-index=1 to ll-Rows
ls-name=ids-table.object.name[ll-index]
ls-batfile=ls-batfile+" BCP" +sle-db-b.
text+".." +ls-name+ &
" out "+ls-path+" \" +ls-name+ ".out" +
-c -S" +sle-server-b.text+" -U" + &
sle-user-b.text+" -P" +sle-pwd-b.text+"
rn"
next
//////////
destroy ids-table
// 判断用户是否选择了批处理命令存放的目录
ls-taskfile=""
ls-taskfile=sle-task-b.text
if ls-taskfile="" then
messagebox ("错误!", "请选择任务存放文
件!", ,stopsign!)
return
end if
// 将批处理命令内容存入用户选择的批处理文

```

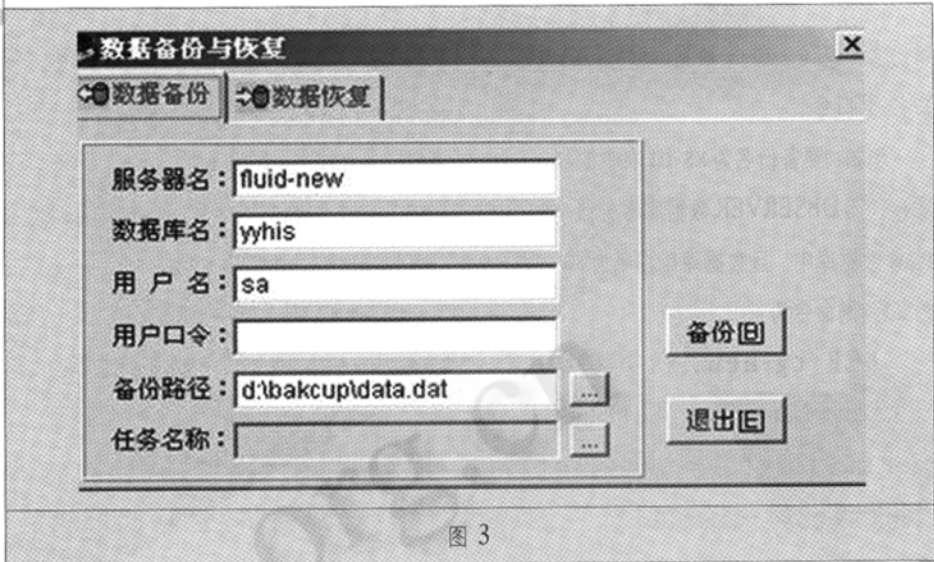


图 3

```

件中
li-FileNum = FileOpen (ls-taskfile,
LineMode!, Write!,
LockWrite!, Append!)
FileWrite (li-FileNum, ls-batfile)
FileClose ( li-filenum )
// 运行批处理命令 run (ls-taskfile,
Minimized!)

```

(3) 数据的恢复。数据恢复的界面如图 3, 此时选中数据恢复 TAB 页, 备份按钮变为恢复按钮, 该按钮下的脚本和上述备份按钮下的脚本差不多, 只是在生成批处理命令时将参数 out 改为 in 就可以了。限于篇幅, 在这里就不说明了。

4 总结

本文中的功能实现技术是笔者在开发完长沙鼓风机厂管理信息系统 (CG-MRP II) 的维护子系统后, 根据经验总结出来的。在 PB6.5+Ms SQL Server 7.0 环境下运行良好, 且程序稳定和具有通用性特点, 这些功能可以不加修改的移植到另外任何一个开发模式为 P B 6 . 5 + M S SQL Server 7.0 的数据库应用系统中。实践证明, 本文所提供的新功能具有很高的执行效率。

参考文献

- 1 Microsoft, Microsoft SQL SERVER 管理员指南, 科学出版社, 1997.

实时监控 | 库锁监控 | 登录信息 | 用户管理 | 用户权限

刷新(F5) | 终止(C) | 退出(E)

进程ID	状态	登录名称	主机名	SPID	数据库名	操作命令
1	sleeping	sa		0	master	SIGNAL HANDLER
2	background	sa		0	yyhis	LOCK MONITOR
3	background	sa		0	yyhis	LAZY WRITER
4	sleeping	sa		0	yyhis	LOG WRITER
5	sleeping	sa		0	yyhis	CHECKPOINT SLEEP
6	background	sa		0	yyhis	AWAITING COMMAND
7	runnable	sa		0	yyhis	SELECT
8	sleeping	sa		0	cg_new	AWAITING COMMAND
9	sleeping	sa		0	cg_new	AWAITING COMMAND

图 2