

数据库中大文本数据存取方法的探讨

Study on Access Methods of the Image Data of Database

李用江 (湛江市湛江海洋大学计算机科学与技术系 524088)

摘要: 在构建分布式应用系统时, 存取数据库中大文本数据方法的选择是非常重要的。本文从软件开发的实际出发, 提出了几种实用技巧, 实现了对数据库中大容量数据的直接存取。

关键词: VB.NET ADO.NET 数据库 图像 存储过程

1 引言

在数据库中可以存储多媒体数据, 如数字、正文、图形、图像和声音等, 其中数字、字符等称为格式化数据; 文本、图形、图像、声音、视频等称为非格式化数据, 非格式化数据具有数据量大、处理复杂等特点。

传统的数据库系统如SQL Sever 2000等只具有对多媒体数据存储的能力, 并没有提供如声音等多媒体数据的直接处理能力。因此, 在构建分布式应用系统时, 对存取数据库中大文本数据方法的选择是非常重要的。数据库中大文本数据存取的方法通常采用语言法, 即通过VC#.NET、VB.NET、VC++、VJ++、PowerBuilder、ASP、Vbscript、Jscript等程序和脚本语言直接存取这种类型字段中的数据。

本文提出了几种切实可行的解决方案,

即基于.NET的ADO.NET技术, 在VB.NET中直接调用执行SQL Server的存储过程, 实现了对多媒体数据库中的大容量数据的直接存取。

2 MS SQL Server 2000 对多媒体的支持

在SQL Server中提供了一个用C++实现的TEXTCOPY.EXE程序(位于MSSQL安装目录下的BINN目录中), 它的主要功能是对图像或大文本数据进行高效地输入输出。可以在MS-DOS方式下执行“textcopy /?”得到它的描述, 摘要如下:

```
TEXTCOPY [/S [sqlserver]] [/U [login]] [/P [password]] [/D [database]] [/T table] [/C column] [/W "where clause"] [/F file] [/I | /O] [/K chunksize] [/Z] [/? ]
```

参数说明:

./S SQL服务器名称或地址

/U 用户名 (SA)

/P 密码

/D 库名

/T 表名

/C 字段名

/W WHERE参数 "where id = 'XXX'"

/F 导出导入的文件名及路径

/I 导入

/O 导出

/K 数据大小 (1K-4KB, 默认为4KB, 可省略)

使用时, 首先要写一个存储过程, 再执行这个存储过程, 从而实现多媒体数据库中的大容量数据的直接存取。创建存储过程的SQL语句如下:

```
CREATE PROCEDURE sp_textcopy (
    @srvname varchar (30), -服务器名
    @login varchar (30), -用户名
```

```
@password varchar (30),-密码
@dbname varchar (30),-数据库名
@lbnam varchar (30),-表名
@colname varchar (30),-字段名
@filename varchar(30),-WHERE子句
-导出导入的文件名及路径
@whereclause varchar (40),
@direction char(1) -导出或导入
AS
```

```
DECLARE @exec_str varchar (255)
SELECT @exec_str =
'textcopy /S ' + @srvname +
'/U ' + @login +
'/P ' + @password +
'/D ' + @dbname +
'/T ' + @lbnam +
'/C ' + @colname +
'/W "' + @whereclause +
"/F ' + @filename +
'/ ' + @direction
```

EXEC master..xp_cmdshell @exec_str

在查询分析器中，使用这种方法对非格式化数据进行输入输出非常方便，也克服了其他方法在不能处理1MB以上的非格式化数据，或者时间太长的弊病。

3 从文件中读入多媒体数据到数据库中

若要在数据存储区和应用程序之间移动数据，必须首先拥有到该数据存储区的连接。在 ADO.NET 中，可以使用以下两个连接对象之一创建和管理连接：

(1) SqlConnection: 该对象管理与 SQL Server 7.0 版或更高版本的连接。

(2) OleDbConnection 该对象管理与通过 OLE DB 访问的任何数据存储区的连接 (为了通用性，本文采用了这种方法)。

如果要执行更新数据库的命令，编程时要有如下几个步骤：

① 如果窗体或组件上尚没有可用的连接

对象，请进行添加。

② 向窗体或组件添加一个数据命令，并用要执行的 SQL 语句或存储过程配置它。

③ 如果命令采用参数，则配置参数。

④ 如果有参数，则添加代码来设置参数值。

⑤ 添加代码以打开与数据命令关联的连接。

⑥ 调用命令的 ExecuteNonQuery/Method，将结果设置为一个整数。

⑦ 关闭连接。从文件中读入多媒体数据到数据库中的方法有多种，有的使用了存储过程，有的是直接插入或更新。下面给出了三种方法供大家参考。为方便叙述，我们使用了 SQL Server 的示例数据库 pubs 中的 pub_info 表[2]，其中包括三个字段：pub_id, char, 4; logo, image; pr_info, text。在这里主要是对字段 logo 的操纵。

3.1 使用 sp_textcopy 存储过程

这种方法适合对大数据的操作。它是通过执行数据命令实现直接对数据库或其他数据源执行更新命令，而不需要数据集或数据适配器。

首先建立一个存储过程 sp_textcopy，然后调用执行它。

3.2 使用 sp_updateimg 存储过程

这种方法适合对大数据的操作。它是将图片文件转换成流文件，然后指定一个缓冲区，利用 Read 方法从当前流中读取一个字节序列，并移动读写指针的位置，把转换后的字节序列读到缓冲区中。最后，利用 SQL 的 update 语句把缓冲区中的字节序列插到数据库中的 logo 字段，也不需要数据集或数据适配器。

实现过程如下：

(1) 首先，创建一个插入操作的存储过程。创建存储过程的语句如下：

```
create procedure sp_updateimg
@pub_id as char(10), @logo as image
as
```

```
update pub_info set logo=@logo where
pub_id=@pub_id
return
```

(2) 建立一个向数据库输入数据的方法，源程序如下：

```
Public Sub DataInnew()
Dim fileName As String = 'C:\img\temp.jpg' '取文件名
Dim stream As FileStream = New
FileStream(fileName, System.IO.FileMode
Open, System.IO.FileAccess.Read)
'创建一个与所要存储的字符序列等
长的缓冲区，
Dim buffer(stream.Length) As Byte
'读取流文件中的字符序列
stream.Read(buffer, 0, stream.Length)
stream.Close()
Dim cmdresults As Integer = 0
'设置命令的类型
OleDbCommand1.CommandText =
"sp_updateimg"
OleDbCommand1.CommandType =
CommandType.StoredProcedure
'更新pub_id为"0736"的logo字段的数据
Dim stringNo As String = "0736"
'设置参数,参数"logo"为LongVarBinary
OleDbCommand1.Parameters("logo").
Value = Buffer
OleDbCommand1.Parameters
("pub_id").Value = stringNo
OleDbConnection1.Open()
Try
'执行命令
cmdresults = OleDbCommand1.
ExecuteNonQuery()
Catch ex As Exception
MessageBox.Show("更新数据命令执
行失败: " + ex.Message)
Finally
Beep()
```

```

End Try
OleDbConnection1.Close()
MessageBox.Show("更新数据记录
数: " + cmdresults.ToString())
End Sub

```

3.3 使用数组

下面所给的方法不使用存储过程，通过执行数据命令实现直接对数据库或其他数据源执行更新命令。它是将图片文件转换成流文件，然后读到一个数组中，再由这个数组形成一个16进制的大字符串，最后，利用SQL的update语句把大字符串插到数据库中的logo字段，也不需要数据集或数据适配器。

这种方法的缺点是处理较大数据时耗时太长。下面的方法DataIn实现了直接将图像文件“C:\img\in.jpg”更新到数据库中去。

```

Public Sub DataIn()
    '要导入图像数据的文件名
    Dim fileName As String = "c:\img\in.
jpg"
    Dim fileIn As FileStream = New
FileStream(fileName, FileMode.Open)
    Dim bbinaryReader As BinaryReader
= New BinaryReader(fileIn)
    '定义所用数组的长度
    Dim length As Long = fileIn.Length
    Dim pic(length) As Byte
    Dim s As String = "0x"
    Dim i As Long
    'Dim tempByte As Byte
    For i = 0 To length - 1
        pic(i) = bbinaryReader.ReadByte()
    '将8位无符号整数化为两位16进制数
        s = s + IntToString(pic(i) \ 16) +
IntToString(pic(i) Mod 16)
    Next i
    fileIn.Close()
    Dim cmdresults As Integer = 0'返回值
    清零
    '形成SQL语句串

```

```

OleDbCommand1.CommandText =
"UPDATE pub_info SET logo=" + s + " WHERE
pub_id='0736"
    '设置SQL语句为Text型，即实际执行
的SQL语句

```

```

OleDbCommand1.CommandType =
CommandType.Text
    '打开连接
OleDbConnection1.Open()
Try
    '执行命令
    cmdresults = OleDbCommand1.
ExecuteNonQuery()
Catch ex As Exception
    MessageBox.Show("更新数据命令
执行失败: " + ex.Message)
Finally
    Beep()
End Try
OleDbConnection1.Close()'关闭连接
MessageBox.Show("更新数据记录
数: " + cmdresults.ToString())
End Sub

```

为了将8位无符号整数化为两位16进制数，要使用下面这个方法。

```

Public Function IntToString(ByVal Int As
Integer) As String
    Select Case Int
    Case 10
        IntToString = IntToString + "A"
    Case 11
        IntToString = IntToString + "B"
    Case 12
        IntToString = IntToString + "C"
    Case 13
        IntToString = IntToString + "D"
    Case 14
        IntToString = IntToString + "E"
    Case 15

```

```

IntToString = IntToString + "F"
    Case Else
        IntToString = IntToString + Int
    End Select
End Function

```

4 从数据库中读出图像数据到文件中

要处理数据库中的数据就要连接数据库，VB.NET提供了ADO.NET数据适配器，它是ADO.NET托管提供程序的组成部分，用于在数据源和数据集之间交换数据。Visual Studio提供两种主要的数据适配器以供与数据库一起使用：

(1) OleDbDataAdapter 对象适用于由OLE DB提供程序公开的任何数据源。

(2) SqlDataAdapter 对象特定于SQL Server。由于该对象不必通过OLE DB层，所以它比OleDbDataAdapter快。但它只能用于SQL Server 7.0或更高版本。

数据集在断开缓存中存储数据。数据集的结构类似于关系数据库的结构，它公开表、行和列的分层对象模型。

4.1 使用 sp_textcopy 存储过程

这种方法同3.1方法相似，从数据库中向外输出数据，只需将上面的程序中的“direction”修改为“O”即可（即OleDbCommand1.Parameters("direction").Value = 'o'）。

4.2 从数据库中向外输出数据时则使用文件名

这种方法必须要有文件名。首先将数据库中的数据向外输出为一个文件，再使用图像工具显示出来，有时候这种方法也是必要的。

下面这个成员函数通过从数据集中取数据，实现了将数据库中的图像数据显示在pictureBox图像控件中。若是其他类型的数据，如声音，只需修改处理数据的方法即可。其中“position”表示数据库中记录所在

表1 几种方法的综合比较

方法	输入输出	时效性	文件大小	使用ADO.NET的情况
3.1使用sp_textcopy存储过程	可输入输出	速度快,效率高	适用大于1MB	使用SQL存储过程、执行数据命令直接操作数据库
3.2使用sp_updateimg存储过程	可输入	速度快,效率高	适用大于1MB	使用SQL存储过程、执行数据命令直接操作数据库
3.3使用数组	可输入	数据的存入和导出时间 开销很大,速度很慢	适用小于1MB	执行数据命令直接操作数据库
4.2从数据库中向外输出数据时使用文件名	可输出	速度快,效率高	适用大于1MB	使用ADO.NET数据适配器和数据集
4.3从数据库中向外输出数据时不使用文件名	可输出	速度快,效率高	适用大于1MB	使用ADO.NET数据适配器和数据集

的位置, "DataSet11.pub_info(position).logo" 表示数据集dataSet11中表pub_info位于position的记录 logo字段的数组。

```
Private Sub DataOut(ByVal position As Integer)
    '导出图像数据的文件名
    Dim fileName As String = "e:\img\temp.jpg"
    Dim fileOut As FileStream = New FileStream(fileName, FileMode.OpenOrCreate)
    Dim binaryWriter As BinaryWriter = New BinaryWriter(fileOut)
    '将图像数据写入文件, dataSet11.pub_info[position].logo为数据集中的第(position+1)个数据
    binaryWriter.Write(DataSet11.pub_info[position].logo)
    '显示图像数据
    Me.pictureBox1.Image = Image.FromStream(fileOut)
    fileOut.Close()
    MessageBox.Show("导出一个图像数据!")
End Sub
```

4.3 从数据库中向外输出数据时不使用文件名

这种方法适合浏览图片而不建立文件,这是一种常用的方法。

```
Private Sub loadpic(ByVal position As Integer)
    '显示图像数据
    Dim Stream As MemoryStream
    Stream = New System.IO.MemoryStream(DataSet11.pub_info[position].logo, True)
    Stream.Write(DataSet11.pub_info[position].logo, 0, DataSet11.pub_info[position].logo.Length)
    Dim bmp As New Bitmap(Stream)
    Stream.Close()
    Me.pictureBox1.Image = bmp
End Sub
```

5 几种方法的比较

在实际应用中,多媒体数据存放在数据库中,与多媒体数据格式无关,数据保密性好,安全可靠,不易丢失和被修改。上述几种方法都有优点也存在不足,表1就这些方法作综合比较,供大家选择参考。

参考文献

- 1 李满潮, Visual C#.NET 高级编程[M], 清华大学出版社, 2002.271-320.
- 2 [美] Jeffrey R · Shapiro. 周之、黄玫译, SQL Server 2000 参考大全[M], 清华大学出版社, 2002. 452-470.
- 3 Microsoft Visual Studio .NET/Framework SDK 的帮助文档[CP/OL].
- 4 萨师宣、王珊, 数据库系统概论[M], 高等教育出版社, 2002.373.
- 5 吴立钊、王耀南, 基于Java语言的图像数据库存取技术[J], 计算机应用, 2002, 22(1):45-46.
- 6 孙林, VB.NET 多媒体编程[M], 清华希望电子出版社, 2002.60-117.