

.NET 框架数据访问结构

秦鑫 朱绍文 陈绪君 金灿 宁国勤 (武汉华中师范大学电子与计算机研究所 430079)

摘要: 本文介绍了 .NET 框架中的 ADO.NET (SQL Server.Net、OLE DB.Net) 数据访问技术, 并给出了两者执行效率测试的方法及评价。

关键词: 数据访问技术 数据提供者 ADO.NET

1 概述

.NET 框架是微软推出的一种新的计算平台, 它使得分布式 Internet 应用程序的开发变得更加简单。 .NET 框架提供了一个一致的面向对象的编程环境, 并可以按照工业标准生成所有通信, 以确保 .NET 框架的代码和任何其他代码集成。 .NET 框架中有两个主要组件: 公共语言运行库 (CLR) 和 .NET 框架类库。公共语言运行库是基础, 它提供核心服务 (如内存管理、线程管理和远程处理); 类库是一个综合性的面向对象的可重用类型集合, 可以使用它开发如 Web 窗体和 XML Web services 等多种应用程序。

.NET 框架中的 ActiveX 数据对象 (ADO.NET) 是向 .NET 程序员提供的公开数据访问服务的类, 它提供对 Microsoft SQL Server 等数据源以及通过 OLE DB 和 XML 公开的数据源的一致访问, 数据共享使用者应用程序可以使用 ADO.NET 来连接到这些数据源, 并检索、操作和更新数据。

ADO.NET 的设计目标是: 提供断开式数据结构; 与 XML 紧密集成; 组合来自多个、不同数据源的数据的通用数据表示形式以及为与数据库交互而优化的功能。

2 ADO.NET 数据访问技术

ADO.NET 是对 ADO 的继承, 如同 ADO 一样, 是一个用来存取数据和信息的 API。 ADO.NET 包含两个核心组件: DataSet 和 .NET 数据提供者。 DataSet 是 ADO.NET 断开

式结构的核心组件, 其设计目的是为了实现在于任何数据源的数据访问; .NET 数据提供者是一组包括 Connection、Command、DataReader 和 DataAdapter 对象在内的组件, 其设计目的是实现数据操作和对数据的快速、只进、只读访问。 DataSet 和 .NET 数据提供者对象模型分别如图 1 和图 2 所示。

DataSet 是一个或多个 DataTable 对象表示的表的集合, DataTableCollection 对象包含 DataSet 中的所有 DataTable 对象 DataRelation

Collection 对象包含 DataSet 中的各种关系。 .NET 数据提供者的 Connection 对象建立与特定数据源的连接; Command 对象用于对数据源执行命令; DataReader 对象用于对数据源中读取只进且只读的数据流; DataAdapter 对象用于用数据源填充 DataSet 并解析更新。

ADO.NET 包括两个 .NET 数据提供者: SQL Server .NET 数据提供者 (专用于 Microsoft SQL Server 7.0 或更高版本) 和 OLE DB .NET 数据提供者。

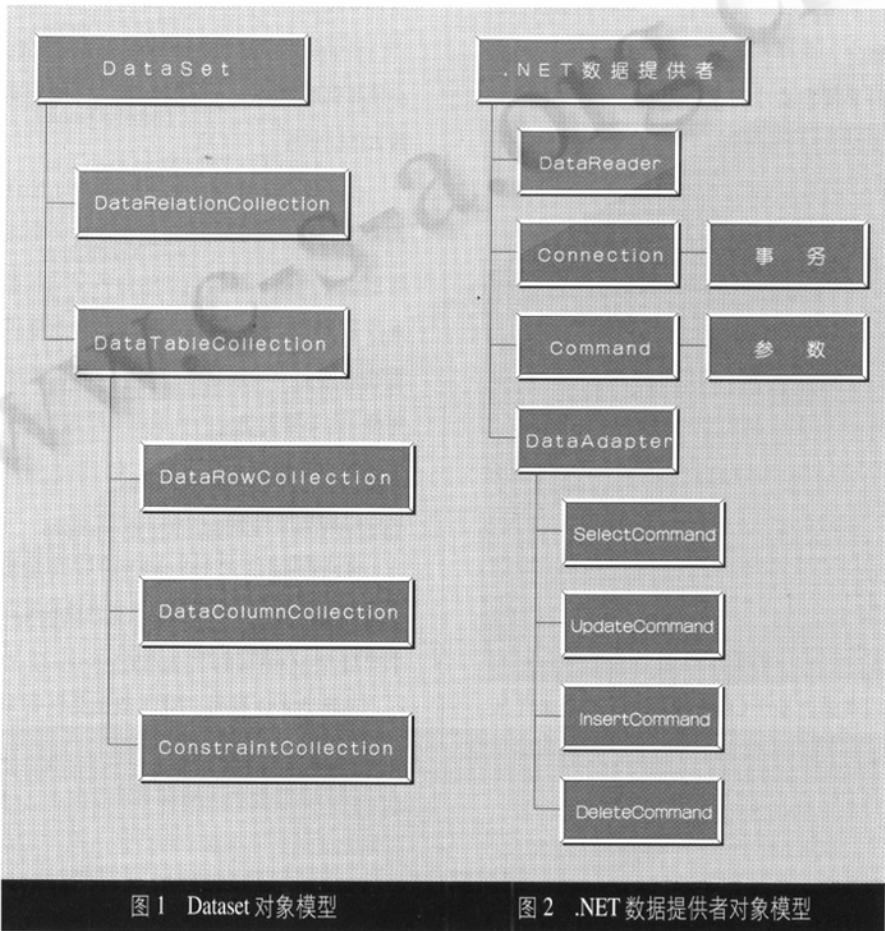


图 1 DataSet 对象模型

图 2 .NET 数据提供者对象模型

2.1 OLE DB.Net 数据提供者

OLE DB .NET 数据提供者通过 COM Interop 来使用本机 OLE DB 启用数据访问, 该数据提供者支持手动和自动事务, OLE DB.NET 数据提供者是一个普通的提供程序, 可以与大多数 OLE DB 提供程序 (如 SQL OLE DB 和 JOLT 等) 一起工作, 它通过 OLE DB 服务组件 (提供连接池和事务服务) 和数据源的 OLE DB 提供程序与 OLE DB 数据源进行通信。

OLE DB.NET 包含的相应的对象分别为: OleDbConnection, OleDbCommand, OleDbDataReader, OleDbDataAdapter, 它们位于命名空间 System.Data.OleDb 中。

使用 OleDbConnection 时首先要指定与数据库连接的连接字符串, 然后调用 Open() 函数打开一个数据库连接; OleDbCommand 对象用来表示和执行一个对数据源执行的命令, 同样在使用 OleDbCommand 对象时要指定命令行字符串, 另外还要指明所用的 OleDbConnection, 此后可以调用 ExecuteNonQuery() 执行无返回行集的命令 (如 Select 和 Update 语句等); 可以用 DataReader 对象来顺序读取返回行的数据。

2.2 SQL Server.Net 数据提供者

SQL Server .NET 数据提供者只提供对 Microsoft SQL Server 7.0 或更高版本的访问, SQL Server .NET 数据提供者使用自己的协议来与 SQL Server 进行通信, 它是经过了优化处理的, 可以直接访问 SQL Server, 而不用添加 OLE DB 或者 ODBC 层, 它完全是在 .NET 框架中实现的, 由于没有通过基于 COM 的 OLE DB 提供程序, 因此 SQL Server.NET 能够提供良好的数据访问性能。

SQL Server.NET 包含的相应的对象分别为: SqlConnection, SqlCommand, SqlDataReader, SqlDataAdapter, 它们位



于命名空间 System.Data.SqlClient 中, 其功能及用法和 OLE DB.NET 数据提供者对应的对象相同, 只是 SqlConnection 和 OleDbConnection 的连接字符串有所区别: SqlConnection 的连接字符串不需要指明 Provider, 而 OleDbConnection 需要指明 "Provider = SqlOleDb".

若要使用 SQL Server .NET 数据提供者, 必须具有对 Microsoft SQL Server 7.0 或更高版本的访问权。

比较 OLE DB.NET 和 SQL Server.NET 两种数据提供者采用的数据访问技术可以看出, 访问数据时所用到的命名空间和类对象是不同的, 最根本的是两种数据提供者在底层上的实现是不一样的, 可以根据需要 (比如需要访问的数据库类型等) 来决定到底使用哪种数据访问技术, 以达到最优的数据访问性能。

图 3 直观的显示 SQL Server.NET 数据提供者和 OLE DB.NET 数据提供者连接数据源的差别。

从图 3 可以看出, .NET 数据提供者用于连接到数据库, 执行命令, 检索结果, 它在设计上是轻量的, .NET 数据提供者在数据源和代码之间创建了一个最小层, 从而在不以牺牲功能为代价的前提下提高了数据访问性能。

3 执行效率评价的程序实现

3.1 测试条件

CPU: CII 600MHz; RAM: 128MB; Hard Disk: 5400RPM;

后台数据库为 MS SQL Server 7.0 的 Pubs, 使用的数据表为 authors, 数据记录数追加为 100000 条。

3.2 SQL Server.NET 数据提供者测试程序

```
Dim myConn1 As SqlClient.SqlConnection
Dim myDA1 As SqlClient.SqlDataAdapter
Dim myComm1 As New SqlClient.
SqlCommand()

Dim time1, time2, time3, time4 As New
DateTime()

Dim diff1, diff2, diff3 As New TimeSpan()
Dim strConn1, strComm1 As String
Dim ds1 As New DataSet()
ds1.Clear()

strConn1 = "data source=localhost; initial
catalog=pubs; uid=sa; pwd=" '定义连接
字符串

strComm1 = "select * from authors"
myConn1 = New SqlClient.SqlConnection
(strConn1)

myDA1 = New SqlClient.SqlDataAdapter
(strComm1, myConn1)

myComm1.CommandText = "update
```

```

authors set state= ' NA ' ' 更新所有记录
state= ' NA '
myComm1.Connection = myConn1
If myConn1.State = ConnectionState.Open
Then ' 如果数据库连接打开, 则将其先关闭
myConn1.Close()
End If
time1 = Now() ' 取此刻时间用于计时, 下同
myConn1.Open() ' 打开数据库连接
time2 = Now()
myDA1.Fill(ds1, " authors ") ' 提取数据
填充数据集
time3 = Now()
myComm1.ExecuteNonQuery() ' 执行更新
命令
time4 = Now()
myConn1.Close() ' 关闭连接
diff1 = time2.Subtract(time1)
diff2 = time3.Subtract(time2)
diff3 = time4.Subtract(time3)
MsgBox(" 数据库连接时间 " + diff1.
ToString + Chr(13) + " 数据提取时间 " + diff2.
ToString + -Chr(13) + " 数据更新时间 " + diff3.
ToString, MsgBoxStyle.OKOnly, " SQL
Server.NET ")

```

3.3 OLE DB.NET 数据提供者测试程序

```

Dim time1, time2, time3, time4 As New
DateTime()
Dim diff1, diff2, diff3 As New TimeSpan()
Dim myConn1 As OleDb.OleDbConnection
Dim myDA1 As OleDb.OleDbDataAdapter
Dim myComm1 As New OleDb.OleDb
Command()

```

```

Dim strConn1, strComm1 As String
Dim ds1 As New DataSet()
ds1.Clear()
strConn1 = " Provider=sqloledb;data
source=localhost;initial catalog=pubs;uid=sa;pwd="
strComm1 = " select *from authors "
myConn1 = New OleDb.OleDbConnection
(strConn1)
myDA1 = New OleDb.OleDbDataAdapter
(strComm1, myConn1)
myComm1.CommandText = " update authors
set state= ' NA ' ' 更新所有记录 state= ' NA '
myComm1.Connection = myConn1
If myConn1.State = ConnectionState.Open Then
myConn1.Close()
End If
time1 = Now()
myConn1.Open()
time2 = Now()
myDA1.Fill(ds1)
time3 = Now()
myComm1.ExecuteNonQuery()
time4 = Now()
myConn1.Close()
myConn1 = Nothing
ds1 = Nothing
diff1 = time2.Subtract(time1)
diff2 = time3.Subtract(time2)
diff3 = time4.Subtract(time3)
MsgBox(" 数据库连接时间 " + diff1.ToString
+ Chr(13) + " 数据提取时间 " + diff2.ToString +
Chr(13) + " 数据更新时间 " + diff3.ToString,
MsgBoxStyle.OKOnly, " OLE DB.NET ")

```

4 测试结果及其分析

4.1 测试结果

见表 1

4.2 结论

由表 1 得出以下结论: 当数据库为 SQL Server 7.0 时, 无论是在数据库连接、数据提取、数据更新方面, SQL Server.Net 提供程序执行效率较 OLE DB .NET 提供程序具有明显的优势。因此当连接微软 SQL Server 数据库时, 建议采用 SQL Server.Net 数据提供者访问数据源, 而不采用 Ole DB.Net 数据提供者。由于 SQL Server.Net 提供程序专为 SQL Server 数据库设计的, 因此当数据库为 Oracle, Sybase 等非微软 SQL Server 的数据库时, 必须采用 Ole DB.Net 数据提供者进行访问。 ■



参考文献

- 1 叶德谦等, 使用 ADO.NET 实现对关系数据库的访问, 微型电脑应用 [J], 2001 年第 17 卷第 8 期。
- 2 施也冲, .NET 组件与 COM 组件的互操作, 现代计算机 [J], 2001.5。
- 3 Evangelos Petroustos, Visual Basic 6 从入门到精通 [M], 电子工业出版社, 1999.1。
- 4 Craig Utley, A Programmer's Introduction to Visual Basic.NET [M]. SAMS Publishing, May 2001。

表 1 测试结果 (单位: 秒)

数据提供者	数据库连接	数据提取	数据更新
OLE DB.NET	1.8226208	10.1546016	5.1574160
SQL Server.NET	1.4320592	6.1288128	3.5050400