

文章编号:1004-9045(2008)04-0363-04

暴雨基础数据库数据的批量录入与性能优化

叶金桃, 陈波

(中国气象局武汉暴雨研究所, 武汉 430074)

摘要:针对暴雨基础数据库,详细描述了在 VC++ 中利用 ADO 接口对 SQL Server 2000 数据库进行编程的步骤和方法,同时结合暴雨基础数据库的实际应用情况,给出了气象数据批量录入的程序。此外,还利用 SQL Server 2000 中的存储功能编程实现了暴雨过程累积降水量的实时自动计算,并利用数据库中的表建立索引等方法实现数据库的性能优化,从而提高了整个系统的效率。

关键词:ADO;VC++;SQL Server;数据库优化

中图分类号:TP319 **文献标识码:**B

1 引言

我国夏季频繁遭受暴雨洪涝灾害袭击,尤其是江淮流域梅雨期暴雨过程较多,因灾造成的损失十分严重,因此对暴雨的研究一直是气象科研预报工作的重点和难点^[1-3],暴雨数据库的建立无论对科研和业务预报均具有重要意义。

随着新的观测手段不断出现,气象资料的种类越来越多,数据量日趋增大,要将这些资料存储于数据库中,光靠手工操作是绝对不行的,需要一种快速的手段将其录入到数据库中。随着计算机技术的发展,数据库技术的应用也越来越广泛。VC++ 6.0 为数据库理论提供了功能强大的开发工具,提供了较快的访问数据库技术,并且具有良好的用户图形界面。本文在刘桂枝等^[4]对暴雨数据库研究的基础上,建立了实时暴雨数据库,阐述了在 Visual C++ 6.0 中利用 ADO 接口来访问 SQL Server 2000 数据库的技术实现快速录入气象数据的具体方法。

2 ADO 与暴雨基础数据库简介

2.1 ADO 概述

ADO(ActiveX Data Objects)是 Microsoft 数据库应用程序开发的新接口,是建立在 OLE DB 之上的高层数据库访问技术,可以编写应用程序以通过 OLE DB 提供者访问和操作数据库服务器中的数据。

OLE DB 是一组“组件对象模型”(COM)接口,是新的数据库低层接口,它封装了 ODBC 的功能,并以统

一的方式访问存储在不同信息源中的数据。但是,OLE DB 应用程序编程接口的目的是为各种应用程序提供最佳的功能,它并不符合简单化的要求。所需要的 API 应该是一座连接应用程序和 OLE DB 的桥梁,这就是 ActiveX Data Objects (ADO)^[5]。

2.2 暴雨基础数据库简介

暴雨基础数据库来源于科技部社会公益研究专项《基于 GIS 的暴雨诱发地质灾害分析预警系统》的子系统。数据库主要收录气象监测资料和降水预报产品,包括地面站资料(全省各地逐日逐时降雨量、各地最新雨情、累积雨量)、卫星资料(风云 2 卫星云图实时链接)、天气雷达资料(实时雷达拼图)和分县雨量客观、主观预报资料。由于系统需要从数据库中读取到最新的信息,因此数据库必须做成一个实时的数据库,需要将每天实时的资料都存放到数据库中,供预警系统调用。本文利用 VC 编程和 ADO 连接数据库实现数据记录的自动更新。

3 在 VC 中使用 ADO 实现暴雨数据的批量录入实例

3.1 生成应用程序框架并初始化 OLE/COM 库环境

创建一个标准的 MFC AppWizard 应用程序 hourlyrain。

3.2 引入 ADO 库文件

使用 ADO 前必须在工程的 stdafx.h 头文件里用直接引入符号 #import 引入 ADO 库文件,以使编译器能正确编译。代码如下所示:

收稿日期:2007-10-15;定稿日期:2008-06-25

基金项目:科技部社会公益研究专项“基于 GIS 的暴雨诱发地质灾害分析预警系统”(2004DIB3J117)资助

作者简介:叶金桃,女,1980 年生,工程师,主要从事暴雨应用研究。E-mail:yejintao@whhr.com.cn

```
#import "c:\program files\common files\system\ado\msado15.dll"no_namespaces rename("EOF" adoEOF)
```

这行语句声明在工程中使用 ADO, 但不使用 ADO 的名字空间, 并且为了避免常数冲突, 将常数 EOF 改名为 adoEOF。现在不需添加另外的头文件, 就可以使用 ADO 接口了。

3.3 初始化 OLE/COM 库环境

必须注意的是, ADO 库是一组 COM 动态库, 这意味着应用程序在调用 ADO 前, 必须初始化 OLE/COM 库环境。在 MFC 应用程序里, 一个比较好的方法是在应用程序主类的 InitInstance 成员函数里初始化 OLE/COM 库环境。

```
if(! AfxOleInit())
{
    AfxMessageBox("OLE 初始化出错! ");
    return false;
}
```

3.4 使用 _ConnectionPtr 接口连接数据库

要求在 SQL Server 2000 中新建一个数据库名为“MyDatabase”的数据库, 用户名为 MyUserID, 密码 MyPassWord, ServerName 为本地服务器名。

在工程头文件 hourlyrain.h 中定义一个指向 Connection 对象的指针:

```
_ConnectionPtr m_pConnection;
```

在 InitInstance 成员函数里添加如下代码:

```
CString strSRC = "Driver=SQL Server; Server=ServerName; Database= MyDatabase;UID= MyUserID;PWD= MyPassWord ";
//初始化 connection 指针, 创建一个 connection 对象的实例
if ( FAILED ( m_pConnection . CreateInstance ( " ADODB . Connection" )))
{
    AfxMessageBox("Create Instance failed");
    return 0;
}
//连接数据库
if(FAILED(m_pConnection->Open(_bstr_t(strSRC), "", "", -1)))
{
    AfxMessageBox("Can not open Database! ");
    m_pConnection.Release();
    return 0;
}
```

可以通过 VC 程序附带的 MSDN 查看 Open 函数的原型。

```
HRESULT Recordset15::Open (const _variant_t & Source,
const _variant_t&ActiveConnection,enum Cursor -TypeEnum Cursor
Type, enum LockTypeEnum LockType, long Options)
```

3.5 使用 _RecordsetPtr 接口打开数据库表

为了取得结果记录集, 定义一个指向 Recordset 对象的指针。在工程头文件 hourlyrain.h 中定义一个指向 Recordset 对象的指针:

```
_RecordsetPtr m_pRecordset;
```

在 InitInstance 成员函数里添加如下代码:

```
CString strSQL="select * from dbo.mytable";
// mytable 为新建的数据库表名
// 初始化 recordset 指针
if(FAILED(m_pRecordset.CreateInstance("ADODB.Recordset")))
{
    AfxMessageBox("Create Instance failed");
    return;
}
//取得记录集
if(FAILED(m_pRecordset->Open(_variant_t(strSQL), _variant_t
(strSRC), adOpenDynamic, adLockOptimistic, adCmdText)))
{
    AfxMessageBox("Open table failed! ");
    m_pRecordset.Release();
    return;
}
```

3.6 数据库表数据的录入

利用 ADO 技术实现了对数据库的连接和打开记录集后, 即可以利用 VC 的 MFC 函数读取气象数据文件, 并进行批量地录入到表中了, 同时也可以对数据库表进行编辑, 添加, 修改, 删除等相关操作。下面以表 real_rain24_data (全省各地最新雨情) 中增加一条新记录为例:

SQL Server 中建立数据库表 real_rain24_data:

```
CREATE TABLE [dbo].[real_rain24_data] (
    [date_dat] [bigint] NOT NULL, /* 资料日期, 形式为
    yyyyymmdd */
    [time_dat] [smallint] NOT NULL, /* 资料时次, 北京时间
    05 时和 08 时 */
    [no_57249] [decimal](7, 1) NULL, /* 区站号为 57249 的站
    点的资料 */
    [no_57251] [decimal](7, 1) NULL,
    ..... /* 湖北省其余的气象站点资料 */
    [no_58501] [decimal](7, 1) NULL /* 区站号为 58501 的站
    点的资料 */
) ON [PRIMARY]
GO
```

VC 中添加一条记录的程序代码:

```
void CHourlyrainApp::OnRealHourlyRain()
{
    int date_dat=20070307, station=57249, time_dat=8;
    float rain=0.5;
    CString no_station;
    no_station.Format("no_%d", station);
    try
    {
        m_pRecordset->AddNew(); //添加一条新记录
        //往数据库字段中添加数据 m_pRecordset->PutCollect
        (_variant_t((long)0), _variant_t((long)date_dat)); m_pRecord-
        set->PutCollect(_variant_t((long)1), _variant_t((long)time_dat));
        m_pRecordset->PutCollect (_variant_t(no_station), _variant_t
        ((float)rain));
        m_pRecordset->Update(); //更新数据库
    }
    //实际应用中, date_dat, station, rain 均为变量, 其值从气象
```

数据文件中获取,time_dat 为实况雨量的时次,此处设定为 08 时。

//如果要进行实时数据的录入,则往数据库中增加数据时,还得考虑记录是否会重复,需要设置主键之类的。

```
catch(_com_error e)
{
    AfxMessageBox(e.ErrorMessage());
    return;
}
m_pRecordset->Close(); //关闭 m_pRecordset 指针对象
m_pRecordset.Release(); //释放 m_pRecordset 指针对象
}
```

代码中使用了 try 和 catch 来处理 COM 异常,如不处理,ADO 的异常有可能使程序崩溃,所以一定要记得捕捉 _com_error 异常。

3.7 关闭并释放 Connection 对象

在析构函数中关闭 Connection 对象并将其释放,代码如下:

```
if(m_pConnection! =NULL)
{
    m_pConnection ->Close();
    m_pConnection.Release();
}
```

4 暴雨数据库的优化处理

4.1 建立索引

由于暴雨基础数据库中的一些表中的数据是取用的实时数据,随着时间的推移,数据量与日俱增,如果数据存放的比较零乱,查询起来就非常困难,需要对其进行优化处理。建立索引是加快查询速度的有效手段,可以根据需要,在基本表上建立一个或多个索引,加快查找优化。在良好的数据库设计基础上,能有效地使用索引是 SQL SERVER 取得高性能的基础。

对于地面基础数据库表 micaps_surf_data,在对其建立索引的时候发现之前录入的部分历史数据中存在不少冗余数据记录,因此对其建立索引之前必须去掉存在的冗余数据,程序如下。

```
delete a from micaps_surf_data a
inner join
(select V04001,V04002,V04003,V04004,V01000
from micaps_surf_data
group by V04001,V04002,V04003,V04004,V01000
having count(*) > 1 )b
on (a.V04001=b.V04001 and a.V04002=b.V04002 and a.
V04003=b.V04003 and a.V04004=b.V04004 and a.V01000=
b.V01000 )
and
autoID not in
(select min(autoID)
from micaps_surf_data
group by V04001,V04002,V04003,V04004,V01000
having count(*)>1)
```

//表中字段 V04001,V04002,V04003,V04004,V01000 分别代表年,月,日,时,区站号

删除冗余数据记录后,对表中的字段 V04001,V04002,V04003,V04004,V01000 分别建立索引,优化查询功能。SQL Server 将根据索引的指示,直接定位到需要查询的数据记录,从而加快 SQL Server 的数据检索操作^[7]。

4.2 利用存储过程计算累积降水量

由于项目中要求提供 1、2、3、7、10、14 天的累积降水数据,因此对湖北省 08 时的逐日降雨量进行累加,所采用的方法是对表 real_rain24_data 中的数据读取进行累加计算,利用 SQL Server 中的触发器和游标功能编程写一个存储过程 (procedure), 名称为 cumulate_rain, 通过配置触发器使此存储过程定时自动运行,对降水累加计算。

SQL Server 中建立累积雨量数据库表:

```
CREATE TABLE [dbo].[cumulate_rain_data] (
[date_dat] [bigint] NOT NULL, /* 资料日期 */
[station] [bigint] NOT NULL, /* 区站号 */
[day_1_rain] [decimal](7, 1) NULL, /* 1 天的雨量累积量 */
[day_2_rain] [decimal](7, 1) NULL, /* 2 天的雨量累积量 */
[day_3_rain] [decimal](7, 1) NULL, /* 3 天的雨量累积量 */
[day_7_rain] [decimal](7, 1) NULL, /* 7 天的雨量累积量 */
[day_10_rain] [decimal](7, 1) NULL, /* 10 天的雨量累积量 */
[day_14_rain] [decimal](7, 1) NULL /* 14 天的雨量累积量 */
) ON [PRIMARY]
GO
```

对应的存储过程 cumulate_rain 的代码如下(以计算 2、3 天的累积雨量为例):

```
CREATE PROCEDURE dbo.cumulate_rain AS
declare @day1 bigint, @day2 bigint,@day3 bigint
select @day1=convert(char(8),getdate(),112)
select @day2=convert(char(8),dateadd(day,-1,getdate()),112)
select @day3=convert(char(8),dateadd(day,-2,getdate()),112)
insert into cumulate_rain_data ( date_dat,station)
(select @day1, station from para_surf_sta )
declare @no_sta bigint
declare @cno_sta varchar(8),@str varchar(5000)

declare sta_cursor cursor //定义游标
for select station from para_surf_sta
open sta_cursor //打开游标

fetch next from sta_cursor into @no_sta
while @@fetch_status=0
begin
set @cno_sta='no_'+str(@no_sta,5)
set @str='
update cumulate_rain_data
set day_2_rain=(
select isnull(sum('+@cno_sta+') ,0)
from real_rain24_data where time_dat=8 and (date_dat='+str(@day1,8)+'
or date_dat='+str(@day2,8)+'),
day_3_rain=(
```

```

select isnull(sum('+@cno_sta+',0)
from real_rain24_data where time_dat=8 and (date_dat='+str
(@day1,8)+' or date_dat='+str (@day2,8)+' or date_dat='+str
(@day3,8)'+')
where date_dat='+str(@day1,8)+' and station='+str(@no_sta,5)
EXEC(@str)
fetch next from sta_cursor into @no_sta
end

close sta_cursor //关闭游标
deallocate sta_cursor //释放游标

```

5 结论与讨论

该文介绍了暴雨基础数据库数据录入与性能优化的一些方法和技巧。上述在 VC 中使用 ADO 实现数据的批量录入实例的方法和程序步骤均在 VC 6.0 下测试通过。除了上面介绍的 ADO 的一些基本功能外, ADO 还可以利用 Command 对象直接执行 SQL 命令取得记录集。因此, ADO 接口技术在 VC++ 中编程技术的灵活运用, 是实现气象数据的批量录入最快捷有效、安全可靠的方法。

同时, 利用 SQL Server 2000 编程功能实现一些数据库表数据的处理也是很有效的。随着数据库表内数据的不断增加, 对其性能进行优化, 提高 SQL Server

数据处理的效率, 以便预警系统的查询使用, 也是非常必要的。除本文介绍的部分优化功能外, 数据库优化还有很多方法, 包括数据库服务器内存的分配、sql 语句的优化等等^[8], 这些有待于今后逐一应用到系统中。经上述优化处理, 数据库的查询检索功能得到了明显的提高, 因而提高了整个系统的效率。

参考文献:

- [1] 陈忠明, 闵文彬, 崔春光. 暴雨中尺度涡旋系统发生发展的诊断[J]. 暴雨灾害, 2007(1): 29-34.
- [2] 王 丽, 金 琪, 柯怡明. 三种数值预报产品短期强降水预报定量误差评估[J]. 暴雨灾害, 2007(4): 301-305.1
- [3] 李银娥, 谌 伟, 张萍萍. 一次华北冷涡暴雨过程的诊断分析[J]. 暴雨灾害, 2007(4): 306-310.2
- [4] 刘桂枝, 王志斌, 陈波. 暴雨基础数据库的建立与维护[G]//武汉区域气象中心. 暴雨·灾害, 北京: 气象出版社, 2004(1): 91-100.
- [5] 求是科技. SQL Server 2000 数据库管理与开发技术大全[M]. 北京: 人民邮电出版社, 2004.
- [6] 启明工作室. Visual C++ SQL Server 数据库应用系统开发与实例[M]. 北京: 人民邮电出版社, 2004.
- [7] 萨师焯, 王珊. 数据库系统概论[M]. 北京: 高等教育出版社, 2000.
- [8] 吴京慧. 数据库优化技术分析研究 [J]. 计算机与现代化, 2004(12): 90-92.

The Data Batch Input and Capability Optimize of the Heavy Rain Basic Database

YE Jin-tao, CHEN Bo

(Wuhan Institute of Heavy Rain, CMA, Wuhan 430074)

Abstract: For the basic database of the heavy rain, the programme steps and methods to SQL Server 2000 database are described in detail by using the interface of ADO in the VC++. Meanwhile, according the actual application status of the heavy rain database, the program of batch meteorology data input are provided in this article. Moreover, some methods to optimize the capability of database, such as making use of the function of SQL server procedure to realize the real time auto-calculate of cumulative rainfall, and establishing index by utilizing the table of database to improve the efficiency of the whole system.

Key words: ADO; VC++; SQL server; Database optimized