

# 基于 Visual FoxPro 环境 ACCESS 数据库操作的实现

王永国, 张士江, 谢 倩

(安徽大学 数学科学学院, 安徽 合肥 230039)

**摘 要:**在 Visual FoxPro 中使用 ACCESS 数据库,能够克服其自身存在的数据库、数据表与索引文件分离、数据表安全性不高等不利因素,较好地发挥二者在数据库管理与编程方面的优势。鉴于这方面的资料较少,系统地阐述 Visual FoxPro 环境下 ACCESS 数据库的操作方法,实现在 Visual FoxPro 中 ACCESS 数据库的建立、压缩与修复;数据表的建立、结构与记录的维护;Visual FoxPro 数据库与 ACCESS 数据库的相互转换、Visual FoxPro 中通用字段与 ACCESS 中 OLE 字段的数据转换等方法是很多用户的客观需求。用 Visual FoxPro 编程方式全面操纵 ACCESS 数据库具有重要的理论与应用价值。

**关键词:**VFP; ACCESS; 数据库; 数据表; 操作; ADO; ODBC; 连接

**中图分类号:**TP312

**文献标识码:**A

**文章编号:**1673-629X(2011)01-0095-05

## Realization of ACCESS Database Operations in Visual FoxPro Environment

WANG Yong-guo, ZHANG Shi-jiang, XIE Qian

(School of Mathematical Sciences, Anhui University, Hefei 230039, China)

**Abstract:** Using the ACCESS database in Visual FoxPro, it can overcome the disadvantages of the separation of the database, data table and index file and the not high security of the data table and so on. It makes better use of database managements and programming in ACCESS and Visual FoxPro respectively. As the less information in this aspect, there are users' objective demands that the ACCESS database operation methods are systematically demonstrated in the Visual FoxPro environment and they are implemented about the ACCESS database creation, the database compression and the database recovery in Visual FoxPro, the creation of data table, the maintenance about structures and records of data table, the mutual conversion of the databases between Visual FoxPro and ACCESS, the conversion of the general type in Visual FoxPro to the OLE type in ACCESS and vice versa. It is no doubt valuable in the database theory and the application.

**Key words:** VFP; ACCESS; database; database table; operation; ADO; ODBC; connection

## 0 引 言

Visual FoxPro(以下简称 VFP)是微软公司开发的基于局域网环境的中、小型数据库系统,支持“结构化查询语言”(SQL)命令和函数。由于其函数丰富、灵活方便、问世较早,在国内一段时期广为流行。时至今日,仍有大批高校将其作为数据库入门语言开设,不少网络前台软件也选用 VFP 开发。但近年来,随着微软公司宣传与开发战略的调整,加之 VFP 本身存在数据表与数据库分离、概念不够严谨及数据表安全性不高等原因,目前正面临微软桌面办公系统 ACCESS 软件

的挑战,原因一方面要归功于 ACCESS 数据库集表、查询、窗体、报表、模块等各种对象于一体,文件单一,具有较高的安全性<sup>[1,2]</sup>;另一方面它采用 Visual BASIC 的编程语言,与 Office 中的 VBA、网站脚本编程语言 VBScript 有着直接的联系,应用较为广泛。如果能用 VFP 编程方式全面操纵 ACCESS 数据库,无疑具有重要的理论与应用价值。

目前,利用 VFP 实现对 ACCESS 数据库控制的软件产品虽不乏先例,但一般没有公开其核心技术,已知的一些方法一般通过建立 ODBC 数据源的方法实现与 ACCESS 数据库的连接并操控其中的数据表<sup>[3-5]</sup>,不仅实现起来麻烦,而且没有提及如何在 ACCESS 中建立数据库(表)、压缩修复维护数据库、输入数据记录等,更没有介绍 VFP 中通用型字段与 ACCESS 中 OLE 字段的相互处理办法,从而为具体实现带来了很

收稿日期:2010-04-26;修回日期:2010-07-12

基金项目:安徽省 2008 年度教学研究项目(01010169)

作者简介:王永国(1965-),男,副教授,硕士生导师,研究方向为计算数学、图像处理、数据库应用。

多困难。我们通过查阅资料并经过在实践中摸索,全面解决了上述问题,从而为在 VFP 中灵活操纵 ACCESS 数据库,提供了实现的方法与技术保障。该方案已成功应用在我们的《VC 多媒体课程教学包的开发与应用》及《猪用选种选料测定系统》等多个项目和软件中,收到了很好的效果。

## 1 主要问题

在一般的数据库应用程序开发中,可能的操作主要有数据库(表)的建立与维护、数据库(表)的操作,但对数据库而言,还有库(表)的压缩与修复问题,否则,当对数据库(表)进行增、删、改等操作时,将会形成数据垃圾,造成存储空间冗余,ACCESS 也是如此。因此,必须实现压缩修复操作。通常对 VFP 中的库(表)可以用 PACK [MEMO|DBF|DATABASE|表名] 实现,在 ACCESS 中也有相应的 CompactDatabase/RepairDatabase 方法实现,但在 VFP 中如何对 ACCESS 数据库库(表)进行压缩与修复?

由于 VFP 与 ACCESS 数据表的数据类型存在差异,在操作时如何实现相应类型的无缝转换?特别是 VFP 中强大而极具特色的通用型(G)字段与 ACCESS 的 OLE 字段如何实现自然对接<sup>[6]</sup>?

另外,由于 VFP 数据库中仅含有数据表的相关信息,其数据表是独立于数据库之外;而 ACCESS 数据表并不独立,是深藏于数据库中的。有时可能已知数据库的名字,但却不知数据库中数据表及其名称信息<sup>[1]</sup>,甚至还要实现 VFP 数据库(表)与 ACCESS 数据库(表)之间的相互转换等等。

基于上述问题,下面给出具体的解决方法:

## 2 实现方法

### 2.1 在 VFP 中创建 ACCESS 数据库

先来简单认识一下 ADO 和 ADOX。在 Visual BASIC 中,使用 ADO 开发数据库应用程序时,要引用对象库“Microsoft ActiveX Data Objects 2.5 Library”(简称 ADO),它是 Visual BASIC 数据库最核心的对象群,也是 VB 数据库开发人员经常所引用的对象库。如果要在程序运行过程中创建数据库和表,还要引用对象库“Microsoft ADO Ext. 2.1 For DDL and Security”,简称为 ADOX<sup>[7-9]</sup>,其库文件名是:Msadox.dll。ADOX 是对 ADO 对象和编程模型的扩展,它将 ADO 扩展为包括创建、修改和删除模式对象。ADOX 中的 Catalog 对象包含了描述数据源模式目录的集合。ADOX 的常用方法有:Append、Create、Delete 等。如需进一步了解 ADOX 的信息,可参阅 Microsoft 发布的有关 ADOX 说明的 Web 页。

了解了 ADOX 后就可以利用它来创建 ACCESS 数据库了。现假设要在 D 盘根目录下建立一个名为 ahdx.mdb 的 ACCESS 数据库,其打开密码是:1234,则建立代码如下:

```
Vcatalog = CREATEOBJECT('ADOX.Catalog')
```

```
Vcatalog.CREATE(' Provider = Microsoft. Jet. OLEDB. 4. 0; Data Source = D:\ahdx.mdb; Jet OLEDB: Database Password = 1234')
```

&& 建立 ACCESS 数据库文件

```
Vcatalog = null && 释放并关闭对象
```

值得一提的是,该方法无需创建 ODBC 数据源,也无需使用机器安装 Office 中的 ACCESS 软件,只要 SYTEM 目录中含有 Msadox.dll 文件即可。

注意:1)为了解决不同版本的 ADO 引用不同(如建立 ACCESS 2007 数据库时,引擎用 OLEDB. 12. 0),可以发布你基于的 Msadox.dll。

2)D 盘根目录下的 ahdx.mdb 数据库不能存在,否则不能成功建立。

3)该方法也可用来创建、删除、操作数据表,但不如 2.2 中的方法简单。有关具体实现,将另撰文介绍。

### 2.2 在 VFP 中创建 ACCESS 数据表及其操作

在 VFP 中创建 ACCESS 数据库中的数据表,首先要解决与 ACCESS 数据库的连接问题,其次就是要解决 VFP 下如何表述 ACCESS 数据库表字段类型问题;最后才是怎样用 SQL 语句来实现。

(1)连接与断开。

为了实现与 ACCESS 数据库的连接,先对 Microsoft 数据访问组件(MDAC)略作介绍。

ODBC 是开放式数据库互连(Open Database Connectivity)的简称,是 Microsoft 引进的一种早期数据库接口技术,常用于和异构数据库连接。

OLEDB(Object Linking and Embedding, Database)是一个基于 COM(Component Object Model)的数据存储对象,能提供对所有类型的数据操作,位于 ODBC 层与应用程序之间。

ADO 对 OLEDB 的操作进行了封装,更易使用,它被设计用来继承微软早期的数据访问对象层,包括 RDO(Remote Data Objects)和 DAO(Data Access Objects)。也就是说 ADO 是位于 OLEDB 之上的“应用程序”,你的 ADO 调用先被送到 OLEDB,然后再交由 ODBC 处理。

鉴于使用 ADO 可能会在不同应用系统中产生 ADO 的版本问题,且在 VFP 中实现时代码量大、书写麻烦,而建立 ODBC 数据源又较为麻烦(方法可参见文献[4]),下面给出无需配 ODBC 数据源,使用 VFP 的 SQLSTRINGCONNECT 函数<sup>[10]</sup>和字符串的方式来连接 MDB 数据库。

```
nhandle = SQLSTRINGCONNECT (" Driver = { Microsoft Access Driver (* . mdb) }; dbq = D: \ahdx. mdb; pwd = 1234")
```

其中 dbq 指定连接的数据库, pwd 指定库密码。如果成功连接到数据源上, nhandle 是一个正的、非零的语句句柄, 此时就可以对该库进行相关的表操作了; 否则返回 -1。这里连接数据库操作类似于 VFP 中的打开数据库。

通常操作完毕, 一般用 SQLDISCONNECT(<连接句柄>) 终止与指定数据源的连接。

## (2) 数据类型。

当在 VFP 中对 ACCESS 数据表进行诸如建表操作时, 必须搞清楚此类型如何在 VFP 中表示, 经过查阅相关资料与实践探索, 总结如表 1 所示。

表 1 ACCESS 数据类型在 VFP 中的表示

数据类型	表示方法
自动编号	Autoincrement (起始值, 增量)
文本或字符	TEXT, VarChar, Char
字节	Byte
整型	Short, Long
长整型	Int, Integer
单精度	Real
双精度	Float, Double, Single
日期/时间	Date, Datetime
货币	Money, Currency
逻辑	Bit, YesNo
备注	Memo
OLE	OLEObject

## (3) ACCESS 数据表的操作。

由于 VFP 支持“结构化查询语言”(SQL) 命令, 因此, 在 VFP 中对 ACCESS 数据表进行诸如建表、修改表结构、删除表等操作, 可以通过 SQLEXEC<sup>[10]</sup> 函数发送一条 SQL 语句到数据源来实现。语法格式如下:

SQLEXEC (数据源的语句句柄, SQL 语句)

如果 SQLEXEC 执行成功返回 1; 如果发生连接级错误返回 -1。

示例代码段 (假定数据库连接成功, 即 nhandle 为一个正的、非零值, 下同):

### \* 新建表

```
=sqlxec (nhandle, "CREATE TABLE XSXX (xh TEXT (3), xm TEXT (8), xb TEXT (2), nl integer, csrq datetime, zp memo, hf bit, bz memo)")
```

### \* 修改表结构 (删除表操作略)

```
=sqlxec (nhandle, "Alter TABLE XSXX DROP xh") && 删字段
=sqlxec (nhandle, "Alter TABLE XSXX Alter Column xh integer Primary Key") && 改属性
```

### \* 查询记录 (修改记录与删除记录类似)

```
IF SQLEXEC (nhandle, "Select * from XSXX Where xh = '002'", "CTemp") > 0
```

Browse

ENDIF

### \* 插入记录<sup>[11, 12]</sup>

在 VFP 中向 ACCESS 数据表中插入单条记录比较容易, 可直接在 SQLEXEC 中使用 INSERT 语句来实现, 不过当插入值为变量时, 需在变量名前加?, 如下例所示:

```
m. xh = "002"
```

```
m. xm = "张三"
```

```
m. xb = "男"
```

```
m. nl = 17
```

```
m. csrq = Date ()
```

```
m. zp = STRCONV (FILETOSTR ([d: \w. jpg]), 13) && 将 d: \w. jpg 转变成字符串存入 OLE 字段
```

```
Ac = SQLEXEC (nhandle, "insert into xsxx (xh, xm, xb, nl, csrq, bz, zp) values (? m. xh, ? m. xm, ? m. xb, ? 17, ? Date (), ? filetostr ([d: \acc. prg]), ? m. zp)")
```

```
IF Ac < 0
```

```
MESSAGEBOX ("插入数据失败!", 16, '错误信息')
```

ENDIF

但当插入批量数据时, 虽然 VFP9.0 的 INSERT 语句也支持 SELECT 子句, 不过用 SQLEXEC 函数无法实现。

故改用下列方法:

```
SET MULTILOCKS ON
```

```
Ac = SQLEXEC (nhandle, "select * from xsxx", "xgb") && 将 xsxx 表内容写入临时表 xgb
```

```
IF Ac > 0
```

```
select xgb
```

```
= cursorsetprop ("tables", "xsxx", "xgb") && 设置临时表 xgb 的更新目标表 xsxx
```

```
= cursorsetprop ("keyfieldlist", "xh", "xgb") && 设置临时表的更新关键字 xh
```

```
= cursorsetprop ("updatablefieldlist", "xh, xm, xb", "xgb") && 设置临时表的更新字段
```

```
= cursorsetprop ("updatenamelist", "xh xsxx. xh, xm xsxx. xm, xb xsxx. xb", "xgb")
```

```
&&& 用 xgb 中的相应字段更新 xsxx 表
```

```
= cursorsetprop ("sendupdates", .t., "xgb") && 设置 xgb 为可更新
```

```
= cursorsetprop ("wheretype", 2) && 设置更新条件
```

```
= cursorsetprop ("buffering", 5, "xgb") && 设置表缓冲
```

```
append from d: \aaa && 假设批量增加的记录
```

```
IF tableupdate (1, .t., "xgb") && 更新表 xsxx 中的内容
```

```
MESSAGEBOX ("数据插入更新成功!", 32, '系统提示')
```

ENDIF

ENDIF

```
SQLDISCONNECT (nhandle) && 断开连接
```

### 2.3 VFP 与 ACCESS 数据库(表)的相互转化

前面已经解决了 VFP 操作 ACCESS 数据库(表)的问题,也就意味着解决了 VFP 数据库向 ACCESS 转换问题。要在 VFP 中解决 ACCESS 数据库转化为 VFP 数据库,必须首先获知 ACCESS 数据库中的表名<sup>[1,13]</sup>及其内容。方法是先利用 VFP 中 SQLTABLES 函数<sup>[3]</sup>将数据源的表名存储到 VFP 的临时表中以获取 MDB 库中表的信息,再将其中的信息转存为 VFP 库中的信息。转化后两表之间字段类型的对应关系如表 2 所示。

表 2 ACCESS 表转换为 VFP 表数据类型对应关系

ACCESS 数据表	VFP 数据表
自动编号 Autoincrement(起始值,增量)	长整形 integer
文本 text	字符型 varchar, char, memo
长整形 integer	长整形 integer
字节型 byte	长整形 integer
整形 short long	长整形 integer
单精度型 real	双精度型 double
双精度型 float double single	双精度型 double
日期/时间 date	日期型 datetime
是/否 yesno	逻辑型 logical
OLE 对象 oleobject	通用型 general
备注 memo	备注 memo
货币 money	货币型 currency
超链接/查阅向导	通用型

示例代码段(假定数据库连接成功,即 nhandle 为一个正的、非零值):

```
nResult=SQLTABLES(nhandle, 'Table', 'ResTable') && 列出
MDB 表信息到 ResTable 表
```

```
IF nResult > 0 CREATE DATABASEahnew && 创建一个名为
ahnew 的 DBC 文件
```

```
SELECT ResTable && 选择 ResTable 并遍历游标
SCAN
```

```
* 生成一个获取 ACCESS 表内容的 SQL SELECT 语句字符串
cSQLCommand = "SELECT * FROM [ "+ALLTRIM( ResTable. ta-
ble_name)+ " ]"
```

```
* 将 ACCESS 表内容通过 SQLEXEC() 函数送到临时表 newdata
中
```

```
nGetData = SQLEXEC(nhandle, cSQLCommand, 'newdata')
```

```
IF nGetData > 0
```

```
* 将表名中的空格用下划线替换,以便做 VFP 文件名
```

```
cNewName = STRTRAN( ALLTRIM( ResTable. table_name ),
" ", "_")
```

```
SELECT newdata
```

```
COPY TO (cNewName) DATABASEahnew && 将临时表保存到数
据库中
```

```
ENDIF
```

```
ENDSCAN
```

```
ENDIF
```

这里,值得注意的是 ACCESS 中的表名在 VFP 中变成文件名,操作时需注意将此转换成符合 Windows 命名规则的名称。另外,当文本型数据的字符数大于 254 时,在 VFP 中将成为备注型。

### 2.4 VFP 中 ACCESS 数据库的压缩与修复

用户在对 ACCESS 数据库进行诸如增、删、改等操作时,常常会有大量临时数据释放,但关闭时 ACCESS 并不回收,久而久之,数据库体积越来越大,因此需要压缩数据库。在没有介绍如何压缩之前,再来认识一下 ACCESS 提供的三种对象模型及其分工:

1) Microsoft ActiveX 数据对象(ADO)包含了在给定的数据源中创建、维护和删除记录所需的对象。

2) Microsoft ADO Ext. for DDL and Security (ADOX)除了提供管理安全性所需的对象之外,还提供了新建数据库及其所含对象所需的“数据定义语言(DDL)”对象。

3) Microsoft Jet and Replication Objects 2.5 Library (JRO),因为 ADO 对象是为了与 Microsoft Jet 数据库以及其他许多数据库一起使用而设计的,所以 Jet 所特有的功能被分解到 JRO 库中。Jet 复制对象(JRO)仅与 ACCESS 数据库一起使用,基本上用于数据库压缩和复制管理<sup>[7]</sup>。

基于上述理论,可以使用 JRO 中 JetEngine 对象的 CompactDatabase 方法来实现对 ACCESS 数据库的压缩任务。

\* 示例代码片段

```
O_je=CreateObject('JRO. JetEngine')
```

\* 待压缩修复的文件

```
strS = " Provider=Microsoft. Jet. OLEDB. 4. 0. ;;Data Source=D:
\ ahdx. mdb;Jet OLEDB:Database Password=1234"
```

\* 压缩修复后存放的文件,若 Database Password=后为空,则压缩修复后取消密码

```
strT = " Provider=Microsoft. Jet. OLEDB. 4. 0. ;;Data Source=D:
\ test. mdb;Jet OLEDB:Database Password="
```

```
O_je. CompactDatabase(strS, strT)
```

```
O_je = null
```

```
ERASE "D:\ahdx. mdb"&& 删除压缩修复前的文件
```

```
RENAME "d:\test. mdb" TO "D:\ ahdx. mdb" && 恢复为压
缩修复文件
```

注意:

1) 压缩后存放的数据库文件不能存在,否则,压缩失败。

2) 也可使用 DAO 的 DBEngine. 36 对象的 CompactDatabase 方法压缩数据库。

### 3 结束语

文中比较全面、系统地阐述了如何在 VFP 环境下操作 ACCESS 数据库(表)问题,从而为扩大 VFP 在数据库领域的应用与安全提供了一种有益的尝试。在探索过程中,要处理好以下问题:

1) 访问数据库的方式很多,在对 ACCESS 数据库(表)处理的不同阶段,各有利弊,可根据操作情况灵活选择。

2) 两者数据表字段的类型在表述上的差异,特别要注意处理好 OLEObject 字段类型与 General 字段类型的转化问题,否则,有可能导致转化后的数据无法正确读出。处理时建议使用 VFP 提供的 STRCONV(<表达式>,<转换类型>)函数,并转换<表达式>中的单字节字符到 base64 二进制形式。

3) VFP 从 7.0 开始 SQL 语句有较大变化,需注意操作上的差别。文中的案例在 VFP 8/9 下测试通过。

#### 参考文献:

- [1] 王永国. 基于 VB 的 ACCESS 无纸化考试系统[J]. 微计算机应用, 2007, 28(1): 108-112.
- [2] VFP 完全控制大、中、小型数据库编程接口[基本介绍][DB/OL]. [2010-06-25]. <http://www.duote.com/soft/9836.html>.
- [3] 汪学祥. 在 Visual FoxPro 中访问非 DBF 格式数据文件[J]. 山西煤炭管理干部学院学报, 2002(4): 74-75.
- [4] 周松, 秦嘉杭, 周晓飞. 在 VFP 中用远程视图实现数据

的安全性[J]. 计算机与现代化, 2002(10): 52-54.

- [5] Basoz C. Linking a DB from Access 2003 to a ForPro application [DB/OL]. 2006-07-19. <http://social.msdn.microsoft.com/Forums/en-AU/visualfoxprogeneral/thread/bebecf90-90be-4df4-aacd-2a6da5e0af53>.
- [6] apple\_8180. 通过 VFP, 向 SQL Server 或 Mdb 等中保存、还原图片[DB/OL]. [2010-07-09]. <http://faq.csdn.net/read/217758>. HTML.
- [7] Hank 动态创建、压缩 Access 数据库(\*.MDB)[DB/OL]. 2000-12-29[2010-02-10]. <http://dev.csdn.net/htmls/2/2773.html>
- [8] my\_love\_xuevfp. 中怎么创建 access 数据库文件[DB/OL]. [2008-11-19]. <http://wenda.tianya.cn/wenda/thread?tid=6e23da0fe01eae2d&clk=wtptcs&exp=1>.
- [9] 李秉璋. 基于 ADOX 接口动态创建数据库方法探索[J]. 计算机应用与软件, 2006, 23(5): 41-42.
- [10] 王苏. Microsoft Visual FoxPro 9.0 中文手册[DB/OL]. 2007-11-30[2009-03-20]. [http://download.csdn.net/source/8405\\*5](http://download.csdn.net/source/8405*5).
- [11] Strahl R. Using Unicode in Visual FoxPro Web and Desktop Applications [DB/OL]. [2005-01-20]. <http://www.programmersheaven.com/2/using-unicode-in-visual-foxpro>.
- [12] wuno2001. VFP6 如何读取及写入资料到 Assess MDB 资料库[DB/OL]. [2008-11]. <http://vfp.sunyear.com.tw/viewtopic.php?t=4290>. [13] Article ID: 260405. How to import table structures and data from Microsoft Access to Visual FoxPro[DB/OL]. [2007-02-12]. <http://support.microsoft.com/kb/260405/en-us/>.

(上接第 94 页)

展,多线程不再是并发执行,而是通过多核的同时运行而达到真正的并行。因而多线程模型在并行模型中的作用越来越大。文中给出了 CMP 集群的新的编程方法 TBB+MPI 并实验验证了性能优势,充分利用了多核计算节点的计算能力,在实际编程过程中,一个主要难点就是 TBB 线程任务分配和 MPI 数据同步,这也是以后要解决的主要问题。

#### 参考文献:

- [1] Grama A, Gupta A. 并行计算机导论[M]. 北京:机械工业出版社, 2005.
- [2] 胡晨霞, 王晓蔚. 基于多核集群系统的并行编程模型的研究[J]. 计算机技术与发展, 2008, 18(4): 70-73.
- [3] Quinn M J. MPI 与 OpenMP 并行程序设计(C语言版)[M]. 陈文光, 武永未, 译. 北京:清华大学出版社, 2001: 212-245.
- [4] Utrera C, Corbalan J, Labarta J. Implementing malleability on MPI jobs[C]//Proc. Parallel Architectures and Compila-

tion Techniques. [s.l.]: [s.n.], 2004: 215 - 224.

- [5] 蔡佳佳, 李名世, 郑锋. 多核微机基于 OpenMP 的并行计算[J]. 计算机技术与发展, 2007, 17(10): 87-91.
- [6] Malyshekin V. Parallel computing technologies[C]// 8th International Conference on PaCT. [s.l.]: [s.n.], 2005.
- [7] 郭静, 祝永志, 王延玲. 基于 MPI 的动态负载均衡算法的研究[J]. 计算机技术与发展, 2009, 19(5): 150-153.
- [8] 单莹, 吴建平, 王正华. 基于 SMP 集群的多层次并行编程模型与并行优化技术[J]. 计算机应用研究, 2006(10): 254-260.
- [9] Foster I. Designing and building parallel programs[M]. 北京:机械工业出版社, 2002.
- [10] Reinders J. Intel Threading Building Blocks 编程指南[M]. 北京:机械工业出版社, 2009.
- [11] Akhter S, Robert J. 多核程序设计技术——通过软件多线程提升性能[M]. 北京:电子工业出版社, 2007.
- [12] 刘维峰, 卢伟, 许海燕. 基于局域网和 MPI 的 PC 集群计算环境[J]. 计算机工程与设计, 2005, 26(5): 1327-1329.