

基于 OCCI 技术存取数据库大对象的方法及实现

全 杰 (中国电子科学研究院 系统集成部 北京 100041)

摘 要: 如何高效快捷地存取 LOB 类型数据是数据库应用程序开发的一个技术难题。对此, 提出基于 OCCI 技术存取 Oracle 数据库 LOB 类型数据的一种方法, 举例阐述了其技术实现过程, 同时给出在 VC6.0 环境中已通过调试的源代码。该方法具有简单快捷、容易处理的优点。

关键词: OCCI; LOB; Oracle; VC; 数据库

Method and Realization of Accessing Large Object in Database Based on OCCI-Technology

QUAN Jie (China Academy of Electronics and Information Technology, Beijing 100041, China)

Abstract: In the development of database application, it is a technologic difficulty to access large object (LOB) data quickly and efficiently. In this paper, a simple and easy method for storing and getting large object data in Oracle database based on OCCI is presented. The process of implementing in technology is also expatiated by some examples. The codes have been debugged and provided under Visual C++ 6.0 environment.

Keywords: OCCI; LOB; Oracle; VC; database

在现代信息系统开发中, 经常需要使用 Oracle 这样的大型关系数据库管理系统来统一管理一些二进制数据, 如文件、图片等。当将整个二进制文件作为单一数据存入 Oracle 数据库时应使用 LOB 类型。LOB 类型数据的存取操作在 API 调用上和其他简单类型不同, 很多缺少开发经验的程序员在处理 LOB 类型数据时往往遇到一些技术问题, 如何高效快捷地存取 LOB 类型数据是当前数据库应用程序开发所面临的一个技术难题。

文献[1, 2]介绍了使用 DBMS_LOB 包存取 Oracle 大对象的一种方法, 文献[3, 4]介绍了在 VC 中利用 Pro* C/C++、OO4O、OCI 技术存取 Oracle 的 BLOB 字段的三种方法。这些方法各有优点, 但技术实现较复杂。本文以二进制大对象(BLOB)为例说明如何在 VC6.0 环境中使用 OCCI 技术实现对 LOB 字段的存取操作。

1 LOB简介

传统的关系模型在处理简单的结构化数据方面非常成功, 一般通过 INSERT 语句就能将简单的结构化

数据直接存储到简单的数据库表中, 而通过 SELECT 语句能直接访问数据库表的简单的结构化数据。但这种模型在处理复杂的非结构化数据时不适用。

为了解决数据库中复杂的非结构化数据存取难题, Oracle 提出了 LOB(Large Object, 大对象), 图像、音频、视频、文件等都是常见的非结构化数据, 这种数据最大特点是数据大小不确定性, 长度是变化的, 要想和普通的简单类型字段一样在数据库中进行存储和访问, 必须使用 LOB。

LOB 是 Oracle 用来存储大对象数据的一种复杂的数据类型。共有四种类型的 LOB: BLOB、CLOB、NCLOB 和 BFILE, 其中 BLOB——内部无结构的二进制大对象, 适用于存储非文本的字节流数据(如程序、图像、音频), 最大长度为 4GB; CLOB 即内部字符大对象, 最大长度为 4GB; NCLOB——内部定长多字节字符大对象, 最大长度为 4GB; BFILE——外部二进制文件, 大小由操作系统决定。

在 Oracle 中, 存储在 LOB 中数据称为 LOB 的值。如用 SELECT 语句对某一 LOB 字段进行查询, 则返回的不是 LOB 的值, 而是 LOB 字段的定位器, 必须通

收稿时间:2009-10-22;收到修改稿时间:2009-12-26

过定位器才能对 LOB 的值进行存取操作。

2 OCCI技术

Oracle C++ 调用接口(Oracle C++ Call Interfaces, 简称 OCCI)是在 C++ 中访问 Oracle 数据库的一组应用程序接口(API)^[5]。OCCI 比 OCI 具有更多的优势,主要表现在下列方面:

(1) 封装了许多复杂的实现细节,提供了一致的调用接口,代码简洁,程序易于开发、理解。

(2) 引入面向对象的设计和编程技术、方法,使得编写的程序更健壮、可靠、可扩展、易复用和易维护。

OCCI 由一系列 Environment、Connection、Statement 和 ResultSet 类和方法构成的。

Environment 类用来创建、终止 OCCI 环境。创建 Environment 的实例是建立数据库连接的基础,程序入口处通过调用 Environment 类的静态方法 createEnvironment 来创建 Environment 类的实例,程序出口处调用 terminateEnvironment 方法来销毁实例。

Connection 类是创建数据库连接的基础类。首先创建一个 Environment 的实例,其次调用这个实例的静态方法 createConnection 建立和数据库的连接,最后调用 Environment 实例的 terminateConnection 方法关闭数据库连接。

Statement 类用来在已建立数据库连接的基础上执行 DDL/DML 语句、SELECT 查询、PL/SQL 程序块和存取程序输出结果。一旦创建了 Statement 类的实例,就可以调用 Statement 类中的 execute、executeUpdate、executeArrayUpdate、executeQuery 方法执行 SQL 命令。Statement 对象用于绑定和存取各种数据类型的值的接口一致。数据类型间的转换由 OCCI 自动处理。

ResultSet 类用来暂时存放 SELECT 语句查询操作获得的结果,提供了一套 get 方法对符合 SQL 条件的行中数据进行访问。

一般来说,在 VC++ 中使用 OCCI 开发数据库应用程序基本步骤如下:

(1) 利用 Environment 类和 Connection 方法创建 OCCI 环境,建立数据库连接。

(2) 创建 Statement 句柄,利用 execute-

Update、executeQuery 等方法执行 SQL 命令。

(3) 利用 terminateConnection、terminateEnvironment 方法关闭数据库连接,销毁 OCCI 环境。

OCCI 为处理 LOB 提供专门的类和方法,如 Blob::getBlob()、Blob::open()、Blob::close()、Blob::getStream()、Blob::closeStream()等。

3 OCCI存取BLOB字段数据的实现

3.1 二进制数据写入 BLOB 字段的过程

假设在 Oracle 9i 数据库中已创建一个表,结构如下:

```
CREATE TABLE test_lob(
ANNEXNAME VARCHAR2(40),
ANNEXFILE BLOB
);
```

BLOB 类型不同于字符、数字或者时间类型,不能用 INSERT 语句直接将值插入到数据库,但可以通过程序调用 OCCI 特定的接口函数来完成。主要过程如下:

(1) 使用 INSERT 语句向表插入一条新记录,其中 BLOB 字段须设置为 EMPTY_BLOB()。

(2) 使用 SELECT 条件语句并用 FOR UPDATE 子句锁定行,检索得到结果集对象。

(3) 从查询结果集对象中获取 BLOB 字段定位器并赋给 BLOB 对象,打开一个 BLOB 对象并设置读写属性。

(4) 取得一个流化对象,把二进制数据读入内存缓冲区并写入流化对象。因内存缓冲区最大容量为 32767 字节,因此在写入大量二进制数据时必须循环写入。

(5) 关闭 BLOB 对象。

下面举例说明如何通过流化写操作将二进制数据文件写入 Oracle 数据库表中的 BLOB 字段。

```
#define ChunkSize 10240
class COcciDB
{
Environment * env; //维护 OCCI 环境类指针
Connection * conn; //维护与数据库连接类指针
Statement * stmt; //执行 SQL 语句类指针
public:
COcciDB(string username,string password,string
```

```

dblink)
{
//创建 OCCI 环境并连接数据库
env = Environment::createEnvironment
(Environment::DEFAULT);
conn = env->createConnection
(username,password,dblink);
stmt= conn->createStatement();
}
virtual ~COcciDB()
{
//断开与数据库连接和终止环境
env->terminateConnection(conn);
Environment::terminateEnvironment(env);
}
void store_file_dbblob(string Filename)
{
/*1、使用 INSERT 语句向表插入一条新记录，其中
BLOB 字段须设置为 EMPTY_BLOB()。*/
string sqlinsert = "insert into
test_blob(annexname,annexfile)
values(:v1 ,EMPTY_BLOB())";
stmt->setSQL(sqlinsert);
stmt->setString(1, Filename);//为变量 V1 绑定参
数
stmt->executeUpdate();
conn->commit();
/*2、使用 SELECT 条件语句并用 FOR UPDATE 子句锁
定行，检索得到结果集对象。*/
string sqlslect = "SELECT ANNEXFILE FROM
test_blob where ANNEXNAME = :v1 FOR UPDATE ";
stmt->setSQL(sqlslect);
stmt->setString(1, Filename);//为变量 V1 绑定参
数
ResultSet* rset = stmt->executeQuery();
while (rset->next ())
{
/*3、从查询结果集对象中获取 BLOB 字段定位器
并赋给 BLOB 对象，打开一个 BLOB 对象并设置读写
属性。*/
Blob blob = rset->getBlob(1);

```

```

blob.open (OCCI_LOB_READWRITE);
WriteFileToBlob (blob, Filename);
blob.close ();
}
stmt->closeResultSet (rset);//关闭结果集合
}
//将二进制数据写入到 BLOB 类型字段中
void WriteFileToBlob(Blob &blob,string Filename)
{
Stream *strm=blob.getStream();
string strPathFileName = "C:\\\" + Filename;
CFile f(strPathFileName.c_str(),CFile::modeRead);
char bVal[ChunkSize+1];
memset(bVal,0,ChunkSize+1);
UINT ulsRead=0;
for(int i=0;;i++)
{
ulsRead=f.Read(bVal,ChunkSize);
if(ulsRead==0)
break;
strm->writeBuffer(bVal,ulsRead);
if(ulsRead<ChunkSize)
break;
}
char* c=(char*)"";
strm->writeLastBuffer(c,0);
blob.closeStream(strm);
f.Close();
}
}

```

3.2 读取 BLOB 字段数据的过程

与上述二进制数据写入数据库过程类似，从数据库中读取 BLOB 字段数据不能通过 SELECT 语句直接将二进制数据读取显示，但可以通过程序调用 OCCI 特定的函数来完成。主要过程如下：

(1) 使用 SELECT 条件语句检索得到 BLOB 类型字段的定位器且用 FOR UPDATE 子句锁定行，否则不能更新 BLOB。

(2) 从查询结果集对象中获取 BLOB 类型字段对象数据并赋给一个 BLOB 对象，打开 BLOB 对象并设置只读属性。

(3) 获取 BLOB 对象的流化值并赋给一个流化对象, 从流化对象中读取流化数据到内存缓冲区并写入文件。因内存缓冲区最大容量为 32767 字节, 因此在读取 BLOB 数据时需要循环读取。

(4) 关闭 BLOB 对象。

下面举例说明如何通过流化读操作把存储在 Oracle 数据库表中的 BLOB 字段数据读取出来并生成文件。

```
class COcciDB
{
//声明、构造函数内容和前面代码一样, 故省略
void get_dbblob_file(string Filename)
{
string sqlslect = "select ANNEXFILE from
test_blob where ANNEXNAME = :v1";
stmt->setSQL(sqlslect);
stmt->setString(1, Filename);//为变量 V1 绑定参
数
ResultSet* rset = stmt->executeQuery();
while (rset->next ())
{
//遍历结果集并作保存处理
Blob blob = rset->getBlob (1);
blob.open (OCCI_LOB_READONLY);
//从数据库读出二进制对象到文件
ReadBlobToFile(blob,Filename);
blob.close();
}
stmt->closeResultSet (rset);//关闭结果集合
}
void ReadBlobToFile(Blob &blob,string Filename)
{
string strPathFileName = "C:\\\" + Filename;
CFile f;
f.Open(strPathFileName.c_str(),CFile::modeWrite|CFil
e::modeCreate);
//创建流对象, 取得 BLOB 字段数据
Stream *instream = blob.getStream ();
//建立内存缓冲区
char bVal[ChunkSize+1];
memset(bVal,0,ChunkSize+1);
```

```
long llsWrite=0;
long size = blob.length();
//ChunkSize=10240, 计算写入文件次数
int n;

n=(size%ChunkSize)==0?(size/ChunkSize):(size/Chu
nkSize+1);
//循环写入文件
for (int i=0;i<n;i++)
{
//将数据包写入文件
llsWrite = instream->readBuffer (bVal,
ChunkSize);
f.Write(bVal,llsWrite);
}
//关闭流和文件
blob.closeStream (instream);
f.Close();
}
```

4 结语

本文讨论了在 VC 中利用 OCCI 技术实现 Oracle 中 BLOB 字段存取操作的一种方法, 具有简单快捷、容易处理等优点。实现的源代码在 Oracle 9.i、VC6.0 环境中已通过调试, 在实际开发中只需稍加扩展即可使用。

参考文献

- 1 马传松, 邓静, 李振坤. 在 Oracle 数据库中如何对多媒体数据对象进行处理. 广东自动化与信息工程, 2003,(3):40 - 42.
- 2 王连备. Oracle 中 LOB 字段的使用和维护. 电脑编程技巧与维护, 2002,(9):35 - 36.
- 3 董金梅, 原有进, 郑建华. 访问 Oracle 大对象数据的技术及实现. 计算机应用研究, 2005,(1):177 - 178.
- 4 卢晓云. 基于 VC 的 Oracle LOB 数据库访问技术. 计算机与网络, 2003,(3):129 - 132.
- 5 Oracle Corporation. Oracle9i Application Developer's Guide-Large Objects(LOB). Oracle Documents.