

## 瘦客户端的 Oracle 数据库连接技术研究

贺致智, 王 晖

(国防科技大学 信息系统与管理学院, 湖南 长沙 410073)

**摘要:**为了解决客户端/服务器方式下的基于 Oracle 数据库的信息系统中客户端体积过大的问题,文中总结了 3 种不用在客户端安装 Oracle 客户端的数据库访问方法:基于 JNI 的数据库调用、基于 Web Service 的数据库调用和基于 OCI 的数据库调用。并对这几种方法进行了一些比较,得出了这几种方法的一些优点和缺点。

**关键词:**Java 本地接口;Oracle;Web service;Oracle 调用接口

**中图分类号:**TP311.138

**文献标识码:**A

**文章编号:**1673-629X(2006)07-0007-03

## Oracle Database Connection Research with Thin Client

HE Zhi-zhi, WANG Hui

(School of Info. System &amp; Management, National University of Defence Techn., Changsha 410073, China)

**Abstract:** Summarize three means using thin client to access Oracle database. The first is that C accesses Oracle database through JNI and JDBC. The second is that accessing database uses Web service through JDBC. The last is using OCI to access Oracle database. At the end of this paper, compares the three ways and acquires the advantages and disadvantages of these methods.

**Key words:** JNI; Oracle; Web service; OCI

## 0 引言

通过对一般的 C/S 结构系统的研究,发现当使用的数据库系统为非微软的数据库系统时,为了能够使客户端与数据服务器进行通信,通常需要在客户端安装相应数据库系统的客户端。然后通过设置 ODBC 数据源来进行数据库的远程访问。这种用法的好处在于屏蔽了所有的数据库的底层结构,在更换数据库系统之后,可以较好地移植系统。但是对于非微软的数据库系统来说,需要在安装数据库系统的客户端之后才能建立起 ODBC 的数据源,当需要在更多的机器上安装这种应用程序的时候,就需要在每台机器上安装数据库系统的客户端,这对于系统的更新是极不利的。

因此,在这里引入了瘦客户端的概念,即在客户端不需要安装庞大的数据库系统的客户端。与 ODBC 方式相比,这里同样实现了对数据库的查询、增加、删除和修改的功能,客户端程序的体积相当小,并且具有一定的可移植性。

## 1 几种技术的介绍

## 1.1 JNI

JNI 是 Java 的本地编程接口,它允许在 Java 虚拟机中运行的 Java 代码与使用另外的编程语言编写的应用程序

和库进行交互。C 与 Java 能够通过 JNI 进行互相调用。这是利用 JDBC 访问 Oracle 数据库的前提。图 1<sup>[1]</sup>描述了 JNI 与 C/C++ 以及 Java 的关系。

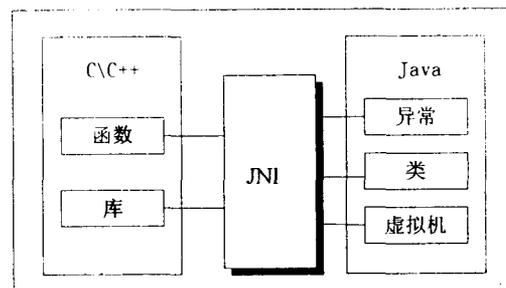


图 1 C, JNI 和 Java 的关系

另外有许多数据库都开发了 Java 版的数据库驱动,并且 Java 具有很好的平台可移植性,这种应用可以在数据库更换之后也能很快地应用在新的数据库上。

## 1.2 Web Service

Web Service 是一个可以通过网络发现并唤醒的可以执行分散的任务或者异族人的软件模块。程序开发者可以开发客户程序,通过远程过程调用或者消息服务来唤醒一系列的 Web Service 来提供某些或者大多数应用逻辑。

Web Service 可以用任何编程语言编程实现,可以运行在任何平台上,它的客户端也可以用任何编程语言编程实现,也可以运行在任何平台上,因而在 Windows 下用 Delphi 编写并运行的客户 Web Service 程序,可以调用 Linux 下用 Java 编写的并运行的 Web Service。

收稿日期:2005-11-07

作者简介:贺致智(1981-),男,湖南常德人,硕士研究生,研究方向为多媒体技术;王 晖,副教授,研究方向为多媒体技术。

Web Service 采用的是面向服务的架构,这种架构的体系结构如图 2<sup>[2]</sup>所示。这种架构有 3 个明确的角色任务:提供者、请求者和代理。它们之间是通过如图 2 所示的相互作用联系在一起。

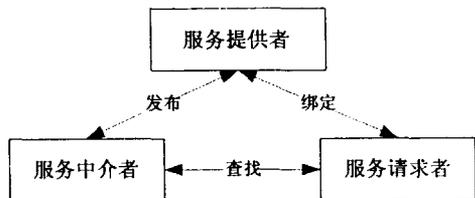


图 2 面向服务的架构

### 1.3 Oracle OCI

OCI(Oracle Call Interface)是 Oracle 提供的 API,用以在第三方语言的过程或函数中访问 Oracle 数据库服务器,它具有这样的优点:

- 1)高效性:编写出来的代码具有很高的效率;
- 2)灵活性:几乎可以控制数据库访问的所有方面;
- 3)支持用户自定义的数据结构;
- 4)对 LOB 提供事务级的支持。

Oracle 的所有操作都是在 OCI 的基础上进行的。所有的客户端程序都需要通过 OCI 来和服务器进行通信。如图 3<sup>[3]</sup>所示,它说明了 OCI 在 Oracle 数据库中所处的位置。

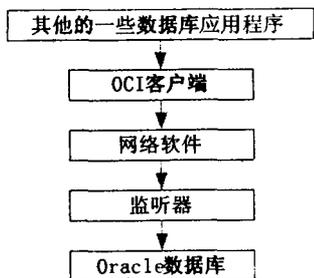


图 3 OCI 客户端所处的位置

## 2 实现方法

上面讨论了几种可以绕过安装 Oracle 客户端的客户端程序设计方法,下面分析它们的实现过程。

### 2.1 通过 JNI 调用 JDBC

利用 Java 的本地应用程序接口,首先利用 Java 编写一个通用的数据库访问的程序,这个程序包含对数据库的查询、增加、修改和删除这 3 项功能。

以查询为例,这里不包含对 LOB 的操作,首先是用 Java 编写提供查询功能的 Java 代码并编译成 Java 字节码。

```

public class DBControl{
public static String[][] Search(String sql)
{
try{
Class.forName("oracle.jdbc.driver.OracleDriver");//找到 Oracle 提供的 Java 版的驱动
Connection conn = DriverManager.getConnection(.....);//

```

取得连接

```

Statement st = conn.createStatement(.....);//创建 Statement
ResultSet rs = st.executeQuery(sql);//执行某个查询
ResultSetMetaData rsm = rs.getMetaData();
//将查询结果转换成 String 类型的二维数组,然后作为方法的返回值返回
rs.close();
st.close();
conn.close();
return ret;
}
catch(Exception e){.....}
}

```

然后,使用 C 编写调用 Java 字节码的程序。

```

void main()
{
JavaVM *jvm;
JNIEnv *env;
JDK1_1InitArgs vm_args;
vm_args.version = JNI_VERSION_1_4;
jint rej = JNI_GetDefaultJavaVMInitArgs(&vm_args);
rej = JNI_CreateJavaVM(&jvm, (void **) &env, &vm_args);
jclass cls = env->FindClass("DBControl");//查找刚刚创建的类
jmethodID
mid = env->GetStaticMethodID(cls, "Search", "(Ljava/lang/String;)[Ljava/lang/String;");//得到 DBControl 类中的 Search 方法
char *sql = "select * from testable";
jobject tt = env->CallStaticObjectMethod(cls, mid, WindowsToString(env, sql));//调用 Search 方法并得到返回的结果,最后利用 env->GetObjectArrayElement()方法得到最后的结果
jvm->DestroyJavaVM();
}

```

同样,将 DBControl 类中 Search 方法进行相应的修改就可以得到相应的增删改的方法,使用同样的方法就可以使用 C 来调用 Java 来实现对数据库的操作。

这种方法的好处在于它绕开了 Oracle 的客户端的安装,利用 Oracle 的 Java 驱动来进行数据库的访问,它需要利用到 Java 虚拟机,因此这种方式的客户端的大小与虚拟机的大小有关。去掉一些不需要的类文件,最终可以得到一个比较小的 Java 虚拟机。

### 2.2 Web Service

利用 Web Service 进行客户端的开发的实质也是利用 Java 的 Oracle 驱动来进行的,它利用 Web Service 将 Java 的 Oracle 驱动进行封装,然后通过其他的编程语言来调用 Web 服务来实现。

首先,建立 Java 的 Oracle 数据库连接的类。

```

public class ssDatabaseAccess {

```

```

private static Connection con = null;
private static String DBdriver = "oracle.jdbc.driver.OracleDriver";
private static String DBurl = "jdbc:oracle:thin:@10.129.10.71:1521:ORCL";
private static String DBuser = "kduser";
private static String DBpassword = "kduser";
public ssDatabaseAccess() {}
public int initDB(String sdriver, String surl, String suser, String spassword) {……} //初始化数据库
public int Close() {……} //关闭数据库
public String[][] execQuery(String sSQL) {……} //将查询的结果按照二维数组返回
public int execUpdate(String sSQL) {……} //执行没有返回值的 SQL 语句
}

```

然后利用集成了 Web Service 的开发环境生成一个 Web Service 的应用。

接下来利用 Delphi 来实现客户端,使用 Delphi 是因为它集成了 Web Service。将已经建立起来的 Web Service 导入到 Delphi 中,就可以得到一些相应的类和文件,直接就可以拿来使用。

这种方式的好处在于,在客户端可以不需要任何附加的东西。但是在服务器端需要有 Web 服务器作为 Web Service 的载体。

### 2.3 OCI

以 OCI 作为 Oracle 客户端开发的应用程序接口的文章已经很多,这里不再赘述,可以参看文献[3~5]。

以这种方式开发的 Oracle 客户端开发可以完成所有

Oracle 客户端的全部功能,具有非常好的性能,并且其客户端体积也大大减小。但是,这种方式使得开发的过程相当复杂,且代码维护困难。

### 3 结束语

通过以上的分析,可以得到如表 1 所示的结论。

	代码 维护性	编程 复杂性	可移植性	服务器 端要求	代码效率
JNI	好	简单	更换数据库之后只需要将数据库驱动改变	没有要求	与 Java 本身的效率有关
OCI	差	复杂	只适用于 Oracle	没有要求	高,直接通过网络访问数据库
Web service	好	简单	更换数据库之后只需要将数据库驱动改变	要求有 Web 服务器	与 Java 本身的效率有关

从表中可见,当程序需要较高的效率的时候,使用 OCI 比较好;当程序没有效率要求的时候,用其它两种方式比较简单,也提高了开发效率。

### 参考文献:

- [1] Sun Microsystems. JNI specification[S]. 2003.
- [2] 吕 曦,王化文. Web Service 的架构与协议[J]. 计算机应用, 2002, 22(12): 62-65.
- [3] 曾志聪,姚国祥. 基于 OCI 技术的 Oracle 数据库连接[J]. 微机发展, 2004, 14(8): 11-13.
- [4] 李乾富,黄书强. VC6.0 访问 Oracle LOB 的方法[J]. 微型机与应用, 2003(2): 7-9.
- [5] 汪林林,马 锐. 用 OCI 开发 ORACLE 数据库的方法[J]. 计算机应用, 2003, 23(12): 46-57.

(上接第 6 页)

### 4 算法分析

用 Excel 的规划求解工具线性规划问题,简单易行,很容易掌握。其规律及技巧可归纳为:在实际的求解过程中,只需确定目标函数单元格及“可变单元格”区域位置两处单元格位置,然后正确地输入约束条件和确定所求的目标是最大还是最小即可求得正确结果。

### 5 结束语

利用 Excel 提供的规划求解法可以解运筹学中的许多问题,譬如线性规划、运输问题、指派问题、网络最优化问题、目标规划、机器分配问题、人事安排等。使用 Excel 的“规划求解”工具可以很方便地解决此类问题,为人们的决策活动制定最优方案。

### 参考文献:

- [1] Balakrishnan V K. Network optimization[M]. London: Chapman Hall, 1995.

- [2] 高 尚. 用 Excel 求解网络规划问题[J]. 计算机与信息技术, 2000(12): 73-76.
- [3] 平 澄. 用 Excel 解方程和得出数学模型的最优化解[J]. 电脑开发与应用, 2002(9): 23-25.
- [4] Ahuja R K. Network Flows: Theory, Algorithms, and Applications[M]. Englewood Cliffs, New Jersey: Prentice Hall, 1993.
- [5] 谢金星,邢文训. 网络优化[M]. 北京:清华大学出版社, 2003.
- [6] 唐焕文. 实用最优化方法[M]. 大连:大连理工大学出版社, 2002.
- [7] Fylstra D, Lasdon L. Design and Use of the Microsoft Excel Solver[J]. Interfaces, 1998, 28(5): 29-55.
- [8] Waren A D. The status of nonlinear programming software: an update[J]. Operations Research Archives, 1987, 35: 489-503.
- [9] 顾运筠. Excel 规划求解的两类应用[J]. 计算机应用与软件, 2005, 22(1): 137-139.