

跨平台异构

利用DBI实现

数据源的存取

华南师范大学计算与信息中心 潘战生

本文介绍了Perl语言的数据库接口DBI模块的体系结构以及DBI应用程序存取数据库的一般方法,并讨论了DBI跨平台存取数据源的技术。

引言

DBI (DataBase Interface) 是Perl语言 (<http://www.perl.com>) 的数据存取应用程序接口 (Application Programming Interface, 简称API)。DBI定义了一套函数、变量以及一系列规则,提供了与实际使用的数据库无关的数据库接口。由于Perl语言是一种在Web编程中被广泛使用的语言,因此,DBI无疑是开发基于Web的数据库应用系统的强有力工具。

Perl语言首先是在UNIX操作系统平台上实现的。由于Perl系统软件在网上可以免费下载,世界各地的程序员又为它开发出了用于各种目的模块(Module),因而使Perl几乎无所不能。现时有的UNIX类操作系统甚至内置了Perl语言系统(如各类Linux)。不但如此,Perl及各种模块也走进了Windows平台,而且它们与UNIX平台做到了很好的兼容,DBI模块也不例外。在Activestate的网站(<http://www.activestate.com>),人们可以免费下载最新的Perl系统、DBI以及各种数据库的驱动模块。用户的DBI应用程序可不加修改地在UNIX及Windows(9x/NT)平台运行,真正实现了与平台无关。

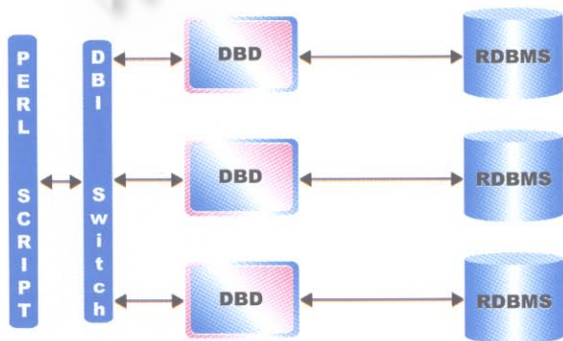


图1 DBI存取数据库的模式

DBI的体系结构

DBI既指一种接口规范,又是一个软件模块,DBI应用程序存取数据库数据的模式见图1所示。用户用Perl语言的DBI接口对数据库进行存取时,DBI并不直接与具体的数据库系统打交道,它通过一组称为数据库驱动程序(Database Driver, 简称DBD)的模块实现与各自的数据库系统的交互。每个数据库系统都有自己的DBD模块。由此DBI就象一个转换开关,将用户的DBI语句交由指定的数据库驱动程序DBD来完成。世界各地的许多自愿人员已开发出了目前流行的各种数据库产品的DBD模块,下表为UNIX平台的DBD模块:

DBD:Adabase	DBD:ODBC
DBD:Altera	DBD:Oracle
DBD:CSV	DBD:Pg
DBD:DB2	DBD:Solid
DBD:Fulcrum	DBD:SYBASE
DBD:Illustra	DBD:Xbase
DBD:Informix	DBD:pNET
DBD:Ingres	DBD:Msq1-Mysql-Modules
DBD:Interbase	

DBI应用程序的框架结构

DBI应用程序存取数据库的一般步骤如下:

1. 装入DBI驱动程序
2. 与数据库进行连接,此举会自动装入所需的DBD模块
3. 打开包含SQL语句的光标(Cursor)
4. 取SQL语句操作结果
5. 关闭光标
6. 关闭数据库连接
7. 程序退出

存取不同数据库的DBI程序代码只是在步骤2处,因要与不同的数据库进行连接(自动装入指定的数据库驱动模块)而稍有差别外,其余代码完全相同。只要程序中与

操作。也许读者会问, ODBC不是也可以做到与数据库无关吗? 不错, 但 DBI 所带的数据库驱动模块 (DBD) 要比 ODBC 多得多, 这由上表可以看出。事实上通过 DBD:ODBC 模块, 任何 ODBC 能访问的数据库 DBI 也能访问。以下给出一个 DBI 存取 ODBC 系统数据源的简单例子。

```
#!/usr/bin/perl

# 装入 DBI 模块
use DBI;

# 与数据库连接, 此处装入 DBD:ODBC 模块
$dbh = DBI->connect("dbi:ODBC:shijuan", "", "");

if( !defined $dbh){
    die $DBI::errstr;
}

# 准备并打开光标
$sql= "select content,submitdate from webtest. dbo.
homework where usr= 'kenny'";
$stmt=$dbh->prepare($sql) or die "Cannt execute
statement:$DBI::errstr";
$rc=$stmt->execute or die "Cann't execute statement:
$DBI::errstr";

# 取查询结果
while(($content,$submitdate)=$stmt->fetchrow()){
    处理查询结果 .....
}

# 关闭光标, 关闭数据库连接
$stmt->finish;
$dbh->disconnect;
```

用 DBI 实现跨平台数据源的存取

这里所谓的跨平台存取数据源, 是指某种平台 (例如 UNIX) 下运行的 DBI 应用程序, 存取位于另一种平台 (例如 Windows NT) 上的数据库。由于应用程序与数据库不是在同一台机器上, 故应用程序及数据库两端都需要数据库网络组件或第三方软件的支持。例如 Sybase SQL Server、Microsoft SQL Server 的 Net library、ORACLE 的 SQL*Net 等。以下以 Linux 操作系统下用 Perl 语言 DBI 应用程序通过 DBD:Sybase 模块, 访问位于 Windows NT 系统下的 Microsoft SQL Server 6.5 数据为例, 说明跨平台数据源的存取方法。这里, Linux 上的 Perl 语言 DBI

程序为客户端, 其网络通信依靠 Sybase SQL Server 11 的 Net library, 因此必须在 Linux 上安装 Sybase SQL Server 11 (实际上只需 Net library 组件即可, 但 Sybase SQL Server 11 for Linux 在安装时未提供只安装其某个组件的选择); Windows NT (IP 地址假设为: 202.116.32.1) 上的 Microsoft SQL Server 6.5 作为服务器端 (数据库服务的 TCP 监听端口假设为 1433), 网络通信依靠其 Net library (安装数据库时已带)。为此, 在 Linux 的 Sybase SQL Server 客户配置文件 /opt/interfaces 中, 加入对于数据库服务器的描述如下:

```
scnunt
query tcp ether 202.116.32.1 1433
master tcp ether 202.116.32.1 1433
```

DBI 脚本中与数据库连接的代码片段见下, 其余代码请参考本文第三部分。

```
.....
BEGIN{
    $ENV{SYBASE}='/opt/sybase';
}
# 通过 DBD:Sybase 与数据源进行连接
$dbh = DBI->connect("dbi:Sybase:server=scnunt','sa','");
if( !defined $dbh){
    die $DBI::errstr;
}
.....
```

与此类似, 经过适当配置也可以在 Windows NT 上安装 Microsoft SQL Server 6.5 的客户组件 (在 setup 时指定只安装客户应用), 通过 DBD:ODBC 模块访问位于 Linux 上的 Sybase SQL Server 数据。■

更 正

由于工作失误, 本刊 2000 年第 4 期目录标题: “基于应用网关的合型防火墙的设计与实现” 应改为“基于应用网关的混合型防火墙的设计与实现”, 特此更正, 并向文章作者及读者致歉。

自 2000 年 3 月份改版以来, 本刊陆续收到许多热心读者来函, 或鼓励、或肯定、或提出中肯意见与建议。编辑部感到: 我们的工作距离读者们的期望还有不小的差距, 我们会更加努力, 避免失误, 力求用更高水平的内容回报读者朋友的真诚与厚爱。

《计算机系统应用》编辑部