

# 基于.NET的 Oracle BLOB 数据高效存取方法

李 良, 柴 毅, 王道斌

(重庆大学自动化学院, 重庆 400044)

**摘要:** 为保证实际项目中海量数据存储的管理效率, 在 ASP.NET 平台上使用 C# 编程, 通过 ADO.NET 访问 Oracle 后台数据库, 采用内存流读取数据, 启用通用自定义 HTTP 请求处理程序同步处理 HTTP Web 请求, 实现了对 BLOB 类型数据的高效存储和读取。该方法同样适用于其他形式的 BLOB 大型数据。

**关键词:** .NET 平台; BLOB 数据; Oracle 数据库

## High-efficient Accessing Method of Oracle BLOB Data Based on .NET

LI Liang, CHAI Yi, WANG Dao-bin

(College of Automation, Chongqing University, Chongqing 400044)

**【Abstract】** In order to meet the practical projects' needs of massive data management efficiency, this paper programs with C# on ASP.NET platform and uses memorystream and the custom HTTP request-processing program to access the BLOB data high-efficiently in Oracle database. The method is suitable for other types of BLOB data, and it is of practical significance.

**【Key words】** .NET platform; BLOB data; Oracle database

### 1 概述

自从微软推出了新型体系结构 ASP.NET 以后, .NET 彻底改变了过去 Windows 应用程序的开发模式。随着技术的发展, 越来越多的程序员采用 .NET 开发 Web 应用程序<sup>[1]</sup>。ASP.NET 带来了全新的 Web 设计概念, 将软件设计和 Web 融为一体, 提供了可视化的开发环境和丰富的服务器控件, .NET 框架还提供了数千个类, 以便开发功能强大的 Web 应用程序。

在实际项目中, Oracle 数据库系统经常需要存储海量图片、文本、Word、Excel、声音、视频、动画等 BLOB(Binary Large Object) 类型的大型数据<sup>[2]</sup>, 因此, BLOB 类型数据的高效存储和读取成为数据库技术中要解决的一个重点问题。

当前流行的解决方案是在存储 BLOB 数据时, 采用存储过程、触发器等驻留内存编译的方法提高访问效率; 读取数据时, 直接读取或调用读取函数来处理。但这样读取 BLOB 数据时就不能发挥其高效的性能, 代码量也会极大地增加。本文采用的方法是: 在存储时, 采用过程或 SQL 变量和文件流等方法; 在读取时, 采用自定义 HTTP 请求处理程序同步处理 Web 请求的方法, 可以随时根据请求要素调用自定义 HTTP 处理函数。这种方法只须通过 Handler.ashx 一般处理类简单地发出 Web 请求, 即可对 BLOB 数据进行任何所需的处理, 从而提高数据的访问效率并减少程序的代码量。

### 2 数据库设计与实现

#### 2.1 Oracle 数据库建表

项目使用 Oracle 作为后台数据库系统。在 Image 数据库 System 方案中的 ImgSpace 表空间建立 ImgTable 表来存储 BLOB 类型数据(图片、文档等), 其结构如表 1 所示, 其中, Product\_id 字段是图片所对应的产品信息表中的产品 ID, 其值通过相应页面的文本框 productid.Text 获得。

表 1 ImgTable 表结构

字段名	数据类型	字段说明	是否空值
Img_id	VarChar2(30)	图片编号(自动)	Not null(主键)
Product_id	VarChar2(30)	关联产品编号	Not null(外键)
Img_name	VarChar2(50)	图片名称	Not null
Img_data	BLOB	图片数据	Not null

Oracle 建表 SQL 源程序如下:

```
CREATE TABLE "SYSTEM"."IMGTABLE" ("IMG_ID"
VARCHAR2(30 byte) NOT NULL, "PRODUCT_ID" VARCHAR2
(30 byte) NOT NULL, "IMG_NAME" VARCHAR2(50 byte) NOT
NULL, "IMG_DATA" BLOB NOT NULL, CONSTRAINT
"SYS_C003143" PRIMARY KEY("IMG_ID") //主键约束
USING INDEX)
TABLESPACE "IMGSPACE"
//在 ImgSpace 表空间建立 ImageTable 表
```

#### 2.2 Oracle 自动编号

由于 Oracle 没有提供类似 SQL Server 的自动编号功能, 因此一般使用 Oracle 中的序列和触发器编程实现自动编号功能。这样对于每个表, 都要编写一段相应的触发器程序, 开发效率会变得很低。也可以使用 GUID 自定义的自动编号格式。GUID 是一个 128 bit 整数(16 Byte), 可用于所有需要唯一标识符的计算机和网络, 且重复的可能性非常小, 所以, 在项目的 Image 数据库中使用 Guid.NewGuid() 方法为字段值自动编号, 生成全局唯一标识符, 从而降低繁琐程度, 提高软件开发效率。

#### 2.3 数据库连接配置

**作者简介:** 李 良(1982 - ), 男, 工程师、硕士研究生, 主研方向: 控制理论与控制工程, Web 应用开发, 路由配置, 无线传感器网络; 柴 毅, 教授、博士生导师; 王道斌, 硕士研究生

**收稿日期:** 2007-11-30 **E-mail:** asimay\_y@126.com

为了与后台数据库相连，需要在Web应用程序的配置文件web.config中配置数据库连接字符串<sup>[3]</sup>：

```
<appSettings>
<add key="ConnectionString"
value="Data Source=Image;User ID=system;
password=manager"/>
//定义 Oracle 数据库连接字符串(数据库名、用户名和密码)
</appSettings>
```

由于 Web 页面获取连接字符串时需要使用的类 ConfigurationSettings 被包含在 System.Configuration 命名空间中，因此在页面相应代码隐藏文件中，需要引入该命名空间。页面获取连接字符串的程序代码如下：

```
Using System.Configuration; //引入命名空间
Private string ORACLECONNSTR=ConfigurationSettings.
AppSettings["ConnectionString"]. ToString(); //获取连接字符串
OracleClient 是微软为 Oracle 特别定制的数据提供程序，
用于在托管空间中访问 Oracle 数据源。但默认情况下，vs2005
并不加载这个组件资源，因此，需要把 System.Data.
OracleClient.dll 文件拷贝到项目解决方案的\Bin 文件夹下，
然后引入此命名空间：Using System.Data.OracleClient。
```

### 3 Web 应用设计与实现

#### 3.1 存储 BLOB 数据的 Web 应用设计

存储 BLOB 类型的图片或文档数据时，可以通过微软提供的上传控件 FileUpload 的属性 PostedFile.ContentLength 获取上传文件字节大小，PostedFile.FileName 获取上传文件文件名，PostedFile.ContentType 获取上传文件类型。然后使用 Stream 流对象读取上传文件数据到一个字节数组中，再将该字节数组数据存储到 Oracle 数据库中。具体步骤如下：

(1)在 ASPX 页面布局：1 个 FileUpload 控件，命名为 FileUp；1 个 button 按钮，命名为 BtnSave(保存)；1 个 Label 控件，命名为 FileLabel，用来显示上传文件的类型、大小和提示信息等。

(2)在 BtnSave\_Click()事件中加入如下代码，保存 BLOB 数据到 Oracle 数据库：

```
if(FileUp.HasFile)//判断是否有文件上传，有文件上传时执行
{String strSQL = "insert into system.ImgTable(Img_id,Product_id,
Img_name,Img_data) values(:ID,:PID,:Name,:Photo)";
OracleConnection cn = new OracleConnection
(ORACLECONNSTR);
OracleCommand cmd = cn.CreateCommand();
//建立 connection 和 command 对象，使用 ADO.NET 访问数据库
cmd.CommandText = strSQL.ToString();
cmd.Parameters.Add(":ID",OracleType.VarChar,30).Value =
Guid.NewGuid().ToString();
//自动编号
cmd.Parameters.Add(":PID",OracleType.VarChar,30).Value =
productid.Text.ToString();
cmd.Parameters.Add(":Name",OracleType.VarChar,50).Value =
FileUp.FileName;
//加入数据库参数
int intLen = FileUp.PostedFile.ContentLength;
//获取上传文件的字节大小
byte[] pic = new byte[intLen];
FileUp.PostedFile.InputStream.Read(pic, 0, intLen); //用 Stream
//流读取数据内容到 pic 数组
cmd.Parameters.Add(":Photo", OracleType.Blob).Value = pic;
```

```
//存储 BLOB 数据到 Oracle
try {cn.Open(); cmd.ExecuteNonQuery();
//执行 SQL 操作
FileLabel.Text="提示：文件"+FileUp.FileName+"上传成功，类型
为："+FileUp.PostedFile.ContentType+"
大小为："+FileUp.PostedFile.ContentLength+"B";}
//显示上传文件的相关信息
catch (Exception ex) {throw new Exception(ex.Message,ex);}
//捕获数据库异常
finally {cn.Close();}}
```

#### 3.2 读取 BLOB 数据的 Web 应用设计

在数据库中，图片的保存方法与文本数据的保存方法不同。文本数据可以直接在数据库中查看，而图片数据是以二进制方式保存，查看时必须使用 Stream 流的 Read 方法读出图片数据，同时需要引入输入输出接口命名空间 System.IO。为实现高效地访问数据，还需要编写自定义 HTTP 处理程序创建进程，并返回动态创建的内存流数据，处理流程如图 1 所示。在项目中，使用 Datalist 控件绑定要即时显示的图片。

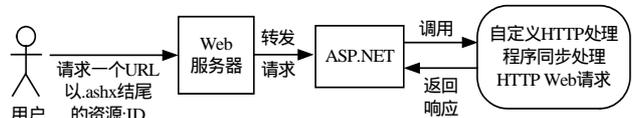


图 1 自定义 HTTP 请求处理程序处理流程

(1)在需要显示图片的 ASPX 页面上布局一个 Datalist 控件，在<ItemTemplate>模板列中添加一个 img 控件，以通用自定义 HTTP 请求处理程序 Handler.ashx 的继承类 Handler 封装 HTTP 请求信息，然后将图片 ID 绑定到 img 控件：

```

```

利用自定义 HTTP 处理程序类 Handler 处理 HTTP Web 请求。

(2)由于内存流 MemoryStream 可有效降低应用程序对临时缓冲区和临时文件的需要，且与其他流相比，执行 I/O 操作的速度和效率高得多，因此可以利用 Handler 类的同步处理 HTTP Web 请求的函数 ProcessRequest，将图片数据用内存流 MemoryStream 读取显示出来。主要实现代码如下：

```
public void ProcessRequest (HttpContext context)
{context.Response.ContentType = "image/jpeg";
context.Response.Cache.SetCacheability
(HttpCacheability.Public);
context.Response.BufferOutput = false;
//响应设置
Int32 id = -1;//初始化
Stream stream=null;//定义 stream 流对象
if(context.Request.QueryString["Img_id"]!=null && context.
Request.QueryString["Img_id"]!="") //利用自定义 HTTP 程序处理类
//Handler 对请求的 QueryString 进行判断
{id=Convert.ToInt32(context.Request.QueryString["Img_id"]);
OracleConnection cn=new OracleConnection
(ORACLECONNSTR);
Oracle cmd = cn.CreateCommand();
//定义 Connection,Command 对象，用 ADO.NET 访问数据库
cmd.CommandText = "select Img_data from system.ImgTable
where Img_id=' "+id+"'";
//检索 Oracle 数据库 BLOB 对象
try {cn.Open(); //打开数据库连接
object result = cmd.ExecuteScalar();
//返回 SQL 查询结果
```

(下转第 68 页)