

钟 菊

(鞍山科技大学高等职业技术学院 114044)

摘要: 本文介绍了 Web 应用程序开发中的分页方法——基于 ADO 的分页方法, 同时又提出了一种效率更高的分页方法——基于存储过程的分页方法。

关键词: Web 应用程序 分页 ADO 存储过程

1 引言

Web 应用程序是一种分布式的应用程序, 要由服务器端的 Web 服务器和客户端的浏览器相互配合来完成功能, 所以其结构又被称为 B / S (Browser/Server) 结构, Web 应用程序一般都要使用后台数据库, 如图 1 所示。

访问数据库并将访问结果组织成页送回给客户端浏览器是 Web 应用程序要实现的一项基本任务, 而在数据库的访问过程中, 经常要返回大量的信息, 如果在单一的页面中显示出来, 既不美观, 使用起来也不方便, 分页法即是解决这种问题的常用手段, 所谓分页法就是将返回信息组织成页, 每一页包含固定数量的记录, 当返回信

息多于一页时, 可以按照用户的要求, 在多页之间反复切换, 本文以采用 ASP+SQL Server 技术开发的应用程序为例, 介绍了基于 ADO 的分页方法, 同时又提出了一种效率更高的分页方法——基于存储过程的分页方法。

2 基于 ADO 的分页方法

该方法利用 Recordset 对象的三个属性实现分页, 其中 AbsolutePage 属性用来设定记录在 Recordset 对象中的位置是在哪一页上 (这里的页只是逻辑上的页), PageSize 属性用来设定或显示 Recordset 对象中每一“页”的记录数, PageCount 属性用于返回 Recordset 中的逻辑页数。

基于 ADO 的分页方法首先按照用户提交的表单数据组织 SQL 语句并访问数据库, 取得一个 Recordset 对象, 然后通过设置 Recordset 对象的 AbsolutePage 和 PageSize 属性实现, 下面以每页显示 5 条记录为例进行说明:

AbsolutePage 属性由表单变量 `scrollaction` 来决定, Scrollaction 的值是用户要显示页的页码, 程序代码如下:

```
<%
If Request.form("scrollaction")= "第一页" then
    显示第一页
    Session("CurrentPage")=1 将 Session 变量 CurrentPage 的值为 1
End If
If Request.form("scrollaction")= "上一页" then
    显示上一页
    Session("CurrentPage")=Session("CurrentPage")-1 将 Session 变量 CurrentPage 的值减 1
End If
If Request.form("scrollaction")= "下一页" then
    显示下一页
    Session("CurrentPage")=Session("CurrentPage")+1 将 Session 变量 CurrentPage 的值加 1
End If
If Session("CurrentPage")="" then  Session 变量 CurrentPage 的值为空
    Session("CurrentPage")=1 将 Session 变量
```

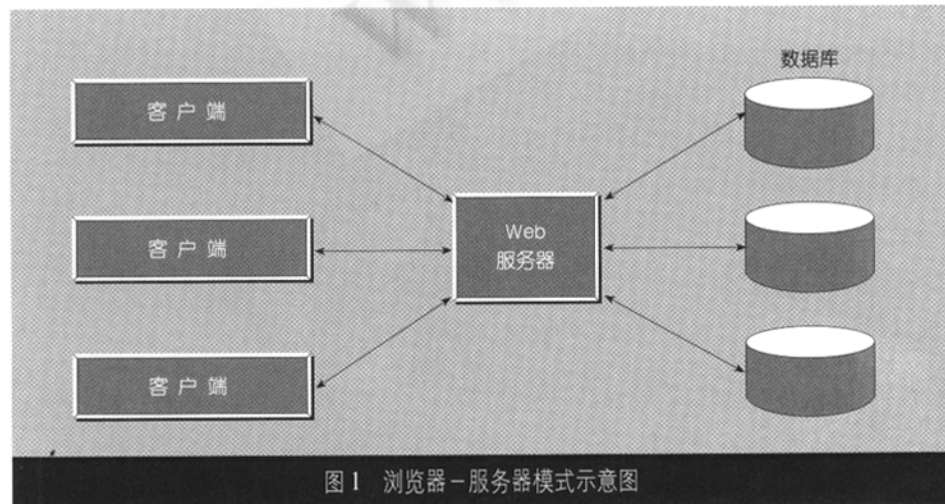


图 1 浏览器-服务器模式示意图

Realization of Paging for Web Application

CurrentPage 的值为 1

End If

Rs.PageSize=5 '设置每页显示的记录数为 5

Rs.AbsolutePage=CurrentPage '设置当前页号

%>

当 AbsolutePage 和 PageSize 确定下来后,

PageCount 的值也就确定下来了。接下来就可

以使用循环控制当前页上记录的显示了。程序

代码为:

```
<table>
```

```
<tr>
```

```
<% '显示 RS 记录集中的所有字段名称
```

```
For i=0 to Rs.Fields.Count-1 %>
```

```
<th> <% =Rs(i).name %> </th>
```

```
<% next %>
```

```
</tr>
```

```
<% '将每页显示的记录数送给变量 rowcount
```

```
rowcount=Rs.PageSize
```

```
'如果当前游标不是记录集的底,且当前页记录
```

```
数不足,继续循环
```

```
do while NOT Rs.EOF And rowcount>0 %>
```

```
<tr>
```

```
<% '显示一条记录
```

```
For i=0 to Rs.Fields.Count-1 %>
```

```
<td> <% =Rs(i) %> </td>
```

```
<% next %>
```

```
</tr>
```

```
<% rowcount=rowcount-1 '记录数减 1
```

```
Rs.MoveNext '游标向前移动
```

```
Loop
```

```
Set Rs=nothing '关闭记录集
```

```
%>
```

```
</table>
```

这种方法简单、方便,但由于其将所有的查

询结果都返回到客户端,所以效率不高。

3 基于存储过程的分页方法

该方法使用了一个存储过程,并将要显示页

的页号和页的尺寸作为存储过程的输入参数,这

样每次仅将查询结果的一个子集返回到客户端,

减少了网络上的信息传输量,所以在 Web 应用程

序中,使用这种方法可以使你的系统效率更高。

具体编程过程如下:

3.1 建立一个新的存储过程

首先,从 Microsoft SQL Server 程序组中启

动 SQL 查询分析器,然后在查询窗口中输入以下
的文本:

```
CREATE PROCEDURE paging_proc /* 创  
建存储过程 */
```

```
@CurrentPage int, /*CurrentPage 为当前页  
号 */
```

```
@PageSize int /*PageSize 为页大小 */
```

```
)
```

```
AS
```

```
DECLARE @StartRecord int, @EndRecord  
int /* 定义变量 */
```

```
BEGIN TRANSACTION GetRecords /*开  
始事务 */
```

```
SET @StartRecord = (((@CurrentPage - 1) *  
@PageSize) + 1) /* 求当前页的第一条记录的记  
录号 */
```

```
SET @EndRecord = (@StartRecord +  
@PageSize - 1) /* 求当前页的最后一条记录的记  
录号 */
```

```
CREATE TABLE #TemporaryTable /* 创建  
临时表 */
```

```
(
```

```
Row int IDENTITY(1,1) PRIMARY KEY, /
```



Realization of Paging for Web Application

IDENTITY 列与主键约束一起使用,从而保证表中各行具有唯一标识--记录号/

```

.....
.....
)
INSERT INTO #TemporaryTable SELECT ...
/* 向临时表中插入数据 */
SELECT * FROM #TemporaryTable
WHERE (Row >= @StartRecord) AND (Row <=
@EndRecord) /* 检索记录号在 StartRecord 和
EndRecord 之间的记录 */

```

DROP TABLE #TemporaryTable /* 删除临时表 */

COMMIT TRANSACTION GetRecords /* 结束事务 */

单击执行查询按钮,建立这个存储过程,该存储过程的名字是 paging_proc。

3.2 在 ASP 网页中调用存储过程

要实现在一个 ASP 网页中调用该存储过程,可以使用命令对象的一个实例。首先把该命令对象的 CommandType 属性设为 adCMDStoreProc。然后,用该命令对象的 CreateParameter() 方法建立两个参数。再用 Append 方法把参数添加到命令对象的 Parameters 集合中。编程代码如下:

```

<%
Set MyConn=Server.CreateObject("ADODB.
Connection")
' 创建一个名字为 MyConn 的 Connection 对象
Set MyCommand=Server.CreateObject
("ADODB.Command")
' 创建一个名字为 MyCommand 的 Command
对象

```

MyConn.Open "DSN=MyData.dsn" ' 打开与数据源的连接

Set MyCommand.ActiveConnection=MyConn ' 设置命令的操作对象

MyCommand.CommandType = adCMDStoreProc ' 设置命令类型为存储过程

MyCommand.CommandText="paging_proc" ' 设置过程名为 paging_proc

Set MyFirstParam=MyCommand. CreateParameter("pageNo",adInteger,adParamInput)

' 创建输入参数 pageNo
MyCommand.Parameters.Append MyFirstParam

' 将参数添加到命令对象的参数集合中
Set MySecondParam=MyCommand.

CreateParameter("pageSize",adInteger,adParamInput)

' 创建输入参数 pageSize
MyCommand.Parameters.Append

MySecondParam
' 将参数添加到命令对象的参数集合中

MyCommand("pageNo")=CurrentPage ' 设置当前页号

MyCommand("pageSize")=5 ' 设置页大小为 5
Set RS=MyCommand.Execute() ' 执行存储

过程,并将结果返回记录集 RS 中
<%

3.3 显示当前页上记录

使用循环控制输出 RS 中的所有记录。编程代码如下:

```

<table>
<tr>
<% ' 显示 RS 记录集中的所有字段名称

```

```

For i=0 to Rs.Fields.Count-1 %>

```

```

<th> <% =Rs(i).name %> </th>

```

```

<% next %>

```

```

</tr>

```

```

<% ' 如果当前游标不是记录集的底,继续循环
do while NOT Rs.EOF %>

```

```

<tr>

```

```

<% ' 显示一条记录

```

```

For i=0 to Rs.Fields.Count-1 %>

```

```

<td> <% =Rs(i) %> </td>

```

```

<% next %>

```

```

</tr>

```

```

<% Rs.MoveNext ' 游标向前移动

```

```

Loop

```

```

Set Rs=nothing ' 关闭记录集

```

```

%>

```

4 结论

综上所述,基于存储过程的分页方法确实优于基于 ADO 的分页方式。在实际应用中,我们也体会到了它的效率之高。读者在 Web 应用程序开发中可以参照使用。 ■

- 1 袁屹、王硕,与 ASP 技术相结合的分布式数据库系统 [J],计算机系统应用,2001.6.
- 2 刘宏峰、陈江波,ASP3.0 网络开发技术大全 [M],人民邮电出版社,2001.4.
- 3 李香敏等,用 ASP 轻松开发 Web 网站 [M],北京希望电子出版社,2000.9.
- 4 袁鹏飞、孙军安,SQL Server2000 数据库系统管理 [M],人民邮电出版社,2001.5.