

文章编号: 1007-130X(2001)06-0004-03

用 PHP实现网上留言或网上论坛的三种典型算法^{*}

Three Typical Algorithms to Implement the Web Message Board or Forum Using PHP

邓春燕, 周治国, 李文印, 玄光哲

DENG Chun-yan, ZHOU Zhi-guo, LI Wen-yin, XUAN Guang-zhe

(吉林大学朝阳校区网络中心, 吉林 长春 130026)

(Network Center, Jilin University Chaoyang Subsection, Changchun 130026, China)

摘 要: 本文主要介绍 PHP的概念以及网上留言版或网上论坛的基本实现过程, 结合 PHP着重介绍了三种实现留言版的算法, 并对其优缺点进行评述。

Abstract The article briefly introduces the basic conception of PHP and the basic implementation of the Web Message Board or Forum. It emphasizes three algorithms of how to implement the Web Message Board or Forum using PHP, and gives comments on the merits and shortcomings of these algorithms.

关键词: 算法; 互连网; PHP语言; 网上留言版; 网上论坛

Key words algorithm; interconnected network; PHP language; Web message board; Web Forum
中图分类号: TP393.092 **文献标识码:** A

1 引言

PHP^[1]的全名是一个递归的缩写名称, “PHP Hypertext Preprocessor”。PHP是一种 HTML内嵌式语言(类似 IIS上的 ASP), 而其独特的语法融合了 C Java Perl以及 PHP式的新语法, 可以比 CGI或 Perl更快速地执行动态网页。

PHP是 1994年由 Rasmus Lerdorf 开始计划开发, 1995年以 Personal Home Page Tools (PHP Tools) 开始对外发表第一个版本。早期它只提供了访客留言本、访客计数器等简单功能。1995年的第二版 PHP加入了 mSQL 的支持, 1997年推出了第三版。

PHP3跟 Apache服务器的紧密结合, 不断更新的功能, 对所有主流与非主流数据库的支持及高速的执行效率, 使得 PHP在程序设计方面有着

更好的应用。它的源代码完全公开, 在 Open Source意识抬头的今天, 它更是这方面的中流砥柱。

PHP的第四代 Zend核心引擎使得整个脚本程序的核心大幅改动, 让程序的执行速度满足更快的要求。最佳化之后的效率, 已较传统 CGI或 ASP等程序有更大的提高。

2 留言版的基本思路

随着因特网的快速发展, 以 Web方式进行交互式信息交换越来越成为需要, 一种基于浏览器输入, 在 Web服务器上运行的程序方法即通用网关接口 CGI^[2]便是在这种情况下产生的。于是, 人们利用 C Perl等语言来编写 CGI程序以实现这种交互式的信息交流, 其中在因特网上最为常见

* 收稿日期: 2001-01-14; 修订日期: 2001-05-28

作者简介: 邓春燕(1977-), 女, 硕士生, 研究方向为网络技术。

通讯地址: 130026吉林省长春市吉林大学朝阳校区网络中心; Tel (0431) 8504068; E-mail: zhouzhiguo@jlu.edu.cn

Address: Network Centre, Jilin University Chaoyang Subsection, Changchun, Jilin 130026, P. R. China

的便是网上留言版或网上论坛

实现网上留言版的大体过程是:

(1) 用户利用 Web 服务器提供的表单提交自己的留言信息;

(2) Web 服务器把用户提交的合法信息保存到某个文件或数据库中, 然后通过 Web 页面把所有的留言信息显示出来 (一般按时间排序);

(3) 其他用户可以回复任何一篇文章 (过程同 (1)、(2))。

过程中的关键是如何显示所有用户的留言信息, 特别是当用户的留言信息非常多的时候。因为它涉及到网页的速度、分层显示以及网页美观, 所以留言版的关键在于如何快速显示多条留言信息。本文借助 PHP 来实现留言版, 使用了三种算法。下面将描述这三种算法的具体思路和优缺点。

3 用 PHP 实现留言版的典型算法

用户的留言信息一般可以存放在文件或数据库中, 本文采用的是存放在数据库 (Sybase 数据库)^[3,4] 中, 表名假定为 guestbook, 大致字段为: Serial (序列号)、Subject (标题)、Body (内容)、LengthSize (长度)、ChildCount (回复数)、HitCount (点击数)、CreateDate (提交时间)、Author (作者) 和 ParentSerial (父序列号)。

3.1 最简单的显示算法

它按照时间递减 (或者说按照序列号的降序) 的顺序逐条显示留言信息, 也就是最新的留言信息排在最上面。这种方式体现不出层次感, 留言之间的关系也不明确。它的具体实现如下 (数据库连接程序略掉):

```
<? php
$ result= sybase. query(" select* from guestbook
order by Serial DESC", $ dbh); (1)
while($ row= sybase. fetch. array($ result))
//循环显示留言信息
{
echo " < u> < li> < i> < a href= \" getmessagebyid? Serial= ".
$ row[" Serial" ]. " \> ".
$ row[" Subject" ]. " < /a> <font color= # 003399> ".
$ row[" LengthSize" ]. " < /font>
字节". $ row[" Author" ]. " ". $ row[" CreateDate" ]. "
< br> < /i> < /u> \n"; (2)
}
?>
```

由于留言信息过多, 这时候应该分页显示, 此

时只需把语句 (1) 换成 \$ result= sybase- query (" select* from guestbook where Serial <= ". \$ Serial, \$ dbh), 并在语句 (2) 处加上一条语句 \$ Serial= \$ row [" Serial"], 基本上就可以实现分页显示的效果了 (注: 还要有换页的按钮或超级链接)。

3.2 递归算法

首先, 定义了一个递归函数, 利用它来显示留言的回复信息, 具体程序如下:

```
<? php //函数的参数含义: $ pSerial 为父序列号, & $ total 保留, $ dbh 为数据库的连接代号
function ShowChildList($ pSerial, & $ total, $ dbh)
{
$ result= sybase. query(" select* from guestbook where
ParentSerial= ". $ pSerial, $ dbh);
if(sybase. num. rows($ result)> 0)
{ //以分层的形式显示留言信息和它的回复信息,
//按照时间的降序
echo " < u> \n";
while($ row= sybase- fetch- array($ result))
{
echo " < li> < a style= \" font-size: 10pt; \" href= \"
getmessagebyid. php? id= ". $ row [ Serial ]. " \
"> ". stripSlashes( $ row [ Subject ]. " < /a> ( ".
$ row [ LengthSize ]. " 字节". $ row [ HitCount ].
" 人次)". $ row [ Author ]. " ( ". $ row [ Create-
Date ]. " ) \n";
if($ row [ ChildCount ] > 0) //若某个留言的回复信息
//个数大于零, 则调用递归函数
ShowChildList($ row [ Serial ], $ total, $ dbh);
}
echo " < /u> \n";
sybase- free. result($ result); //释放结果集
}
}
```

```
? >
<? php //显示父留言的程序 (含分页功能)
if(! isset($ Serial))
//若没有分页, 则显示第一页
$ result= sybase. query(" select* from guestbook where
ParentSerial= 0 order by Serial DESC", $ dbh);
else //分页
$ result= sybase. query(" select* from guestbook where
ParentSerial= 0 and Serial <= ". $ Serial
" order by Serial DESC", $ dbh);
if(sybase. num. rows($ result) < 1)
echo " 目前还没有任何留言";
else
{
echo " < u> \n";
$ len= 0;
While($ row= sybase- fetch- array($ result))
{
echo " < li> < a style= \" font-size: 10pt; \" href= \"
getmessagebyid. php? id= ". $ row [ Serial ]. " \
```

```

“>”.stripSlashes($row['Subject']).“</a>
($row['LengthSize']).“字节”.$row['Hit-
Count’].“人次”).$row['Author’].“($row
['CreateDate’].“)\n”;
if($row['ChildCount']>0)
ShowChildList($row['Serial'],$total,$dbh);
$len=$len+1;
// $len是已显示的留言数目
if($len>=MAXSHOW)break;
// MAXSHOW是每页最多显示的留言个数
}
echo“</u>\n”;
?>

```

3.3 索引算法

索引算法 (本文所取的名称) 在存储留言时做了重大改进,使得显示留言的速度提高了很多,其思路是: 一个留言版在正常情况下, 用户看的时间远远多于用户填写留言的时间。所以, 把每页的留言信息都建成一个已索引好的文件, 放在 Web服务器上, 用户浏览时, Web服务器便打开这个已经索引好的文件, 这样显示速度大大提高了。对于如何建立这个索引文件, 本文采取用户提交或回复留言时利用程序重新建立第一页的索引文件 (若用户的留言是新的留言或第一页上某留言的回复信息) 或某页的索引文件 (若用户的留言是某页上某留言的回复信息) 的办法, 而无需改动其它索引文件。这样, 用户向 Web提交留言信息的速度就有所下降, 但用户浏览留言的速度提高了, 相比较而言, 总体速度有了很大提高。

```

function ShowChildList($pSerial) //显示回复信息的函数
{
    $Depth=$Depth+1;
    $result=sybase_query(“select* from guestbook where
        ParentSerial=”. $pSerial.” order by Serial
        DESC”, $dbh);
    if(sybase_num_rows($result)>0)
    {
        while($row=sybase_fetch_array($result))
        {
            fputs($fp,$Depth.“ ”.$row[“Serial”].“ ”.$row
                [‘Author’].“ ”.$row[‘Subject’].“ ”.$row
                [‘ChildCount’].“ ”.$row[‘HitCount’].“ ”.$row
                [‘LengthSize’].“ ”.$row[‘CreateDate’].“ ”.
                $row[‘ParentSerial’].“\n”);
            if($row[‘ChildCount’]>0)
            ShowChildList($row[‘Serial’]);
        }
        sybase-free-result($result);
    }
    $Depth=$Depth-1;
}

```

```

function ReIndexMessage()
//重新建立索引文件的函数
{
    $Depth=0;
    $fp=fopen(“guestbook.idx”,“w”,RDWR|LOCK,70);
    //打开索引文件
    $result=sybase_query(“select* from guestbook where Par-
        entSerial= -1 order by Serial DESC”, $dbh); if(sybase-
        num_rows($result)>0)
    {
        while($row=sybase_fetch_array($result))
        {
            fputs($fp,$Depth.“ ”.$row[“Serial”].“ ”.$row
                [‘Author’].“ ”.$row[‘Subject’].“ ”.$row
                [‘ChildCount’].“ ”.$row[‘HitCount’].“ ”.$row
                [‘LengthSize’].“ ”.$row[‘CreateDate’].“ ”.$row
                [‘ParentSerial’].“\n”);
            if($row[‘ChildCount’]>0)
            ShowChildList($row[‘Serial’]);
        }
        fclose($fp);
    }
}

```

由于分页显示的程序和父留言信息的显示程序较长, 此处从略。

4 结束语

采用什么样的算法, 不光取决于速度, 还得看算法占用的空间大小。以上三种算法都经过试验 (详见 <http://202.98.1.60/index.php> 和 <http://202.98.1.60/~cust4956/index.php>), 相对来说算法所占的存储空间很小, 而且执行速度非常快, 在因特网中基本上不会给用户带来额外的等待时间。

参考文献:

- [1] 网胜工作室. PHP4.0程序设计 [M].北京: 北京希望电子出版社, 2000.
- [2] Dwight J, Erwin M, Niles R. CGI开发使用手册 [M].金帆工作室,译.北京: 机械工业出版社, 1998.
- [3] 陶浦洲,李强. Sybase数据库技术大全 [M].北京: 科学出版社, 1997.
- [4] Du Bois P. MySQL网络数据库指南 [M].钟鸣,田晓涛,译.北京: 机械工业出版社, 2000.