

Part1

- 1.在可以用file_get_contents替代file、fopen、feof、fgets等系列方法的情况下，尽量用file_get_contents，hosted by 因为他的[效率](#)高得多！但是要注意file_get_contents在打开一个URL文件时候的PHP版本问题；
- 2.尽可能的少进行文件操作，虽然PHP的文件操作效率也不低的；
- 3.[优化](#)Select SQL语句，在可能的情况下尽量少的进行Insert、Update操作；
- 4.尽可能的使用PHP内部函数（但是我却为了找个PHP里面不存在的函数，浪费了本可以写出一个自定义函数的时间，经验问题啊！）；
- 5.循环内部不要声明变量，尤其是大变量：对象(这好像不只是PHP里面要注意的问题吧？)；
- 6.多维数组尽量不要循环嵌套赋值；
- 7.在可以用PHP内部字符串操作函数的情况下，不要用正则表达式；
- 8.foreach效率更高，尽量用foreach代替while和for循环；
- 9.用单引号替代双引号引用字符串；
- 10.“用i+=1代替i=i+1。符合c/c++的习惯，效率还高”；
- 11.对global变量，应该用完就unset()掉；

Part2

静态调用的成员一定要定义成 static (PHP5 ONLY)

贴士：PHP 5 引入了静态成员的概念，作用和 PHP 4 的函数内部静态变量一致，但前者是作为类的成员来使用。静态变量和 Ruby 的类变量(class variable)差不多，所有类的实例共享同一个静态变量。

QUOTE:

```
// PHP CODE Highliting for CU by dz902
```

```
<?php
class foo {
    function bar() {
        echo 'foobar';
    }
}

$foo = new foo;

// instance way

$foo->bar();

// static way

foo::bar();
?>
```

静态地调用非 static 成员，效率会比静态地调用 static 成员慢 50-60%。主要是因为前者会产生 E_STRICT 警告，内部也需要做转换。

使用类常量 (PHP5 ONLY)

贴士：PHP 5 新功能，类似于 C++ 的 const。

长话短说

- 2005-2008 Lineks.cn, All rights reserved . Powered Sablog-X,
- 武汉实习中
 - 中国足协去死,垃圾!
 - 意大利居然输了?
 - 再一次听到了黄健翔的解说久违了
 - 哎!中国队出现无望了啊!
 - 升级了,本站PR到 2 了,哈哈,:)

Sablog-X1.6皮肤下载

- [Lin.X Style Sablog-X 1.6皮肤](#)
- [灰色系 Sablog-X 1.6皮肤](#)

随机文章

- [jQuery选择器总结](#)
- [《明朝那些事儿》- 不看要后悔](#)
- [php效率优化](#)
- [网络“X客”大集合：博客、维客、奇客、播客、闪客、摩客……](#)
- [日历代码](#)

分类

- [记录生活](#) [17]
- [休闲娱乐](#) [4]
- [网文摘要](#) [8]
- [PHP&jQuery](#) [16]
- [单片机&MCU](#) [4]
- [经验技巧](#) [8]
- [网页设计](#) [8]

标签 »

[php](#) ⁷[ie](#) ⁴[网页](#) ⁴[firefox](#) [足球](#) [单片机](#) [优化](#) [ff](#) [轮换](#) [图片](#) [黄健翔](#) [语录](#) [第4季](#) [越狱](#) [seo](#) [ajax](#) [菜单](#) [jquery](#) [美剧](#) [插件](#) [sablog](#) [皮肤](#) [大学](#) [css](#) [效率](#) [麻城](#) [配色](#) [设计](#) [效果](#) [人生](#) [爱情](#) [前途](#) [工作](#) [毕业](#) [版本](#) [共存](#) [framework](#) [zend](#) [框架](#) [执行时间](#)

存档

- [2008年07月](#) (3)
- [2008年06月](#) (8)
- [2008年05月](#) (8)
- [2008年04月](#) (8)
- [2008年03月](#) (8)
- [2008年02月](#) (5)
- [2007年12月](#) (7)
- [2007年11月](#) (6)
- [2007年10月](#) (8)
- [2007年07月](#) (4)

使用类常量的好处是：

- 编译时解析，没有额外开销
- 杂凑表更小，所以内部查找更快
- 类常量仅存在于特定「命名空间」，所以杂凑名更短
- 代码更干净，使除错更方便

(暂时)不要使用 `require/include_once`

`require/include_once` 每次被调用的时候都会打开目标文件！

- 如果用绝对路径的话，PHP 5.2/6.0 不存在这个问题
- 新版的 APC 缓存系统已经解决这个问题

文件 I/O 增加 => 效率降低

如果需要，可以自行检查文件是否已被 `require/include`。

不要调用毫无意义的函数

有对应的的常量的时候，不要使用函数。

QUOTE:

```
// PHP CODE Highliting for CU by dz902
```

```
<?php
php_uname('s') == PHP_OS;
php_version() == PHP_VERSION;
php_sapi_name() == PHP_SAPI;
?>
```

虽然使用不多，但是效率提升大概在 3500% 左右。

最快的 Win32 检查

QUOTE:

```
// PHP CODE Highliting for CU by dz902
```

```
<?php
$is_win = DIRECTORY_SEPARATOR == '\\';
?>
```

- 不用函数
- Win98/NT/2000/XP/Vista/Longhorn/Shorthorn/Whistler...通用
- 一直可用

时间问题 (PHP>5.1.0 ONLY)

你如何在你的软件中得知现在的时间？简单，「`time()` `time()` again, you ask me...」。

不过总归会调用函数，慢。

现在好了，用 `$_SERVER['REQUEST_TIME']`，不用调用函数，又省了。

加速 PCRE

- 对于不用保存的结果，不用 `()`，一律用 `?`

这样 PHP 不用为符合的内容分配内存，省。效率提升 15% 左右。

最新评论

- [Lin.x](#): `margin /**/:11px...` (07-23)
- [呵呵](#): FOR ie5的加个`/**/`啥意... (07-23)
- [Sn.LUF](#): 呵呵，哪儿有2.0啊？我也想要个... (07-17)
- [danny](#): 你好强悍啊，我也喜欢玩这个游戏 (07-15)
- [danny](#): PDF文件里的文字不是可以直接复... (07-15)
- [danny](#): 你在说什么陀螺儿话撒？我要找陀螺... (07-15)
- [danny](#): 我也支持意大利队撒! (07-15)
- [quojian的百奥谷](#): 哈哈，有感悟篇啦，谢谢。 (07-11)
- [Lin.x](#): 学习框架是大趋势,好多公司看重这... (07-11)
- [Jesse](#): 还没用过Fremework呢。 (07-10)
- [Lin.x](#): 不好意思,我的程序也是别人给的,... (07-10)
- [fandt](#): Hi,Lin.x能否发一份SAB... (07-10)
- [向奕](#): www.sablog.cn向奕... (07-08)
- [Lin.x](#): 我的幸福已接近尾声了，马上就要开始... (07-02)
- [郭健的百奥谷](#): 还在上大学么？好幸福~~ (07-02)

友情链接

- [Jesse's Blog](#)
- [捡球小子的草窝](#)
- [荷月听雨](#)
- [浅浅蓝's Blog](#)
- [Oh-Cool](#)
- [Kevin Liu's Blog](#)
- [I am HJin](#)
- [Fanbin's Blog](#)
- [华夏太子's Blog](#)
- [向奕's blog](#)
- [Sablog Theme Download](#)
- [Yzsh's Happy Life](#)
- [百奥谷](#)
- [Mr.B 网络日志](#)
- [耳门网络](#)

管理登录

- [Log in](#)
- [Reg](#)

- 能不用正则，就不用正则，在分析的时候仔细阅读手册「字符串函数」部分。有没有你漏掉的好用的函数？

例如：

```
strpbrk()
strncasecmp()
strpos()/strrpos()/stripos()/strripos()
```

加速 strstr

如果需要转换的全是单个字符的时候，用字符串而不是数组来做 strstr：

QUOTE:

// PHP CODE Highlighting for CU by dZ902

```
<?php
$addr = strstr($addr, "abcd", "efgh"); // good
$addr = strstr($addr, array('a' => 'e',
                             // ...
                           )); // bad
?>
```

效率提升：10 倍。

不要做无谓的替换

即使没有替换，str_replace 也会为其参数分配内存。很慢！解决办法：

- 用 strpos 先查找(非常快)，看是否需要替换，如果需要，再替换

效率：

- 如果需要替换：效率几乎相等，差别在 0.1% 左右。
- 如果不需要替换：用 strpos 快 200%。

邪恶的 @ 操作符

不要滥用 @ 操作符。虽然 @ 看上去很简单，但是实际上后台有很多操作。用 @ 比起不用 @，效率差距：3 倍。

特别不要在循环中使用 @，在 5 次循环的测试中，即使是先用 error_reporting(0) 关掉错误，在循环完成后打开，都比用 @ 快。

善用 strncmp

当需要对比「前 n 个字符」是否一样的时候，用 strncmp/strncasecmp，而不是 substr/strtolower，更不是 PCRE，更千万别提 ereg。strncmp/strncasecmp 效率最高(虽然高得不多)。

慎用 substr_compare (PHP5 ONLY)

按照上面的道理，substr_compare 应该比先 substr 再比较快咯。答案是否定的，除非：

- 无视大小写的比较
- 比较较大的字符串

不要用常量代替字符串

为什么：

- 需要查询杂凑表两次
- 需要把常量名转换为小写(进行第二次查询的时候)
- 生成 E_NOTICE 警告
- 会建立临时字符串

效率差别：700%。

不要把 count/strlen/sizeof 放到 for 循环的条件语句中

贴士：我的个人做法

QUOTE:

```
// PHP CODE Highliting for CU by dZ902
```

```
<?php
for ($i = 0, $max = count($array); $i < $max; ++$i);
?>
```

效率提升相对于：

- count 50%
- strlen 75%

短的代码不一定快

QUOTE:

```
// PHP CODE Highliting for CU by dZ902
```

```
<?php
// longest
if ($a == $b) {
    $str .= $a;
} else {
    $str .= $b;
}

// longer
if ($a == $b) {
    $str .= $a;
}
$str .= $b;

// short
$str .= ($a == $b ? $a : $b);
?>
```

你觉得哪个快？

效率比较：

- longest: 4.27
- longer: 4.43
- short: 4.76

不可思议？再来一个：

QUOTE:

```
// PHP CODE Highliting for CU by dZ902
```

```
<?php
// original
```

```

$d = dir('.');
while (($entry = $d->read()) !== false) {
    if ($entry == '.' || $entry == '..') {
        continue;
    }
}

// versus
glob('.*');

// versus (include . and ..)
scandir('.');
?>

```

哪个快？

效率比较：

- original: 3.37
- glob: 6.28
- scandir: 3.42
- original without OO: 3.14
- SPL (PHP5): 3.95

画外音：从此也可以看出来 PHP5 的面向对象效率提高了很多，效率已经和纯函数差得不太多了。

提高 PHP 文件访问效率

需要包含其他 PHP 文件的时候，使用完整路径，或者容易转换的相对路径。

QUOTE:

```

// PHP CODE Highliting for CU by dz902

<?php

include 'file.php'; // bad approach

incldue './file.php'; // good

include '/path/to/file.php'; // ideal

?>

```

物尽其用

PHP 有很多扩展和函数可用，在实现一个功能的之前，应该看看 PHP 是否有了这个功能？是否有更简单的实现？

QUOTE:

```

// PHP CODE Highliting for CU by dz902

<?php
$filename = "./somepic.gif";
$handle = fopen($filename, "rb");
$contents = fread($handle, filesize($filename));
fclose($handle);

// vs. much simpler

file_get_contents('./somepic.gif');
?>

```

关于引用的技巧

引用可以：

- 简化对复杂结构数据的访问
- 优化内存使用

QUOTE:

```
// PHP CODE Highliting for CU by dz902

<?php
$a['b']['c'] = array();

// slow 2 extra hash lookups per access
for ($i = 0; $i < 5; ++$i)
    $a['b']['c'][$i] = $i;

// much faster reference based approach
$ref =& $a['b']['c'];
for ($i = 0; $i < 5; ++$i)
    $ref[$i] = $i;

?>
```

QUOTE:

```
// PHP CODE Highliting for CU by dz902

<?php
$a = 'large string';

// memory intensive approach
function a($str)
{
    return $str.'something';
}

// more efficient solution
function a(&$str)
{
    $str .= 'something';
}

?>
```

这篇文章发表2007, November 18, 9:23 PM , 并被分类于[PHP&jQuery](#) , 已被 409 人阅读, 1条评论。本文采用[创作共享](#) 版权协议, 要求署名、非商业和保持一致。本站欢迎任何非商业应用的转载, 但须注明出自"[Lin.X's Blog](#)", 保留原始链接, 此外还必须标注原文标题和链接。


您可能还对这些日志感兴趣

- [Zend Framework 之程序执行时间的获取](#) 浏览 : 164 评论 : 4
- [猜数字游戏](#) 浏览 : 541 评论 : 3
- [说几个你\(可能\)不知道的招](#) 浏览 : 418 评论 : 2
- [对博客小小优化了下](#) 浏览 : 629 评论 : 6
- [PHP使用zlib扩展实现页面GZIP压缩输出](#) 浏览 : 427 评论 : 0

共有 1 条评论



[xJin](#) 2008, April 22, 7:48 AM

 你的网站是sablog-x啊！自己加了很多的功能啊！
很羡慕你的技术！
这篇文章也很有用啊！

发表评论

<input type="text"/>	您的昵称 (必填,如果已是注册用户请先登陆)
<input type="text"/>	电子邮件 (必填,不会公布,方便联系)
<input type="text"/>	您的网站
<div></div>	

与本文无关的评论请到[留言簿](#)发表,谢谢合作! :)

<input type="text"/>	<div><div>9245</div></div> 验证码(*)
<input type="button" value="发表评论"/>	