TURING 图灵程序设计丛书 Web开发系列

Apress®

Pro PHP Application Performance

Tuning PHP Web Projects for Maximum Performance

高性能PHP应用开发

[美] Armando Padilla
Tim Hawkins 著

盛海艳 刘霞 译

- Yahoo公司技术专家力作
- PHP性能优化修炼秘籍
- 掌握各种重构技术和最佳实践

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Web开发系列

Pro PHP Application Performance

Tuning PHP Web Projects for Maximum Performance

高性能PHP应用开发

[美] Armando Padilla
Tim Hawkins 著
盛海艳 刘霞 译

人民邮电出版社
北京

图书在版编目（CIP）数据

高性能PHP应用开发 /（美）帕蒂拉（Padilla, A.），（美）霍金斯（Hawkins, T.）著；盛海艳，刘霞译. --
北京：人民邮电出版社，2011.11
（图灵程序设计丛书）
书名原文：Pro PHP Application Performance:Tuning PHP Web Projects for Maximum Performance
ISBN 978-7-115-26495-4

Ⅰ. ①高… Ⅱ. ①帕… ②霍… ③盛… ④刘… Ⅲ. ①PHP语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2011)第201252号

内 容 提 要

本书是一本广受好评的PHP性能优化方面的图书，通过介绍PHP的原理和相关的工具集来实现调优性能的目的。它分析和研究了Web应用程序的前端和后端，并系统地提升了其性能和运行效率。本书还介绍了PHP编码最佳实践的运用以及如何使用工具来应用缓存技术。另外书中也涉及了对Web服务器的优化和数据库的优化。

本书适合PHP开发人员阅读。

图灵程序设计丛书

高性能PHP应用开发

◆ 著 [美] Armando Padilla，Tim Hawkins
译 盛海艳 刘 霞
责任编辑 傅志红
执行编辑 李 盼

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：11.75
字数：284千字 2011年11月第1版
印数：1－3 500册 2011年11月北京第1次印刷

著作权合同登记号 图字：01-2011-0805号

ISBN 978-7-115-26495-4

定价：39.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223
反盗版热线：(010)67171154

译 者 序

作为一种HTML内嵌式语言，PHP的历史可追溯至1994年，它是一种在服务器端执行的脚本语言，其风格有点类似C语言，但除了融合它自己独创的语法和C语言的语法之外，其中也有Java语法和Perl语法的影子，因此可称得上是集几大语言精华于一身的优秀语言。正如我们所熟知的，它最出众的特点是能够快速执行动态网页。

回顾一下PHP的发展历史，可粗略地分为几个阶段。1994年，Rasmus Lerdorf创建了PHP，这位PHP之父最初只是用它来统计自己网站的流量。1995年，他以PHP Tools这个名称对外发布了第一个版本，这就是最早的PHP 1.0，功能也十分简单。随着PHP的使用越来越广泛，大量开发人员开始加入到PHP的开发阵营中。在同年，PHP 2.0发布了。这一版加入了对MySQL的支持，从此PHP作为开发动态网页的卓越工具也得到了业内的一致认可。1997年和2000年分别又发布了PHP 3和PHP 4，大量新特性出现在这两个版本中。目前大多数人使用的主流版本是PHP 5，这是在2008年发布的。但无论你使用的是哪个版本，都会从本书中获益，因为本书的核心是基本原理，可以适用在所有版本上。

本书的结构很清晰，分为两大部分，Web应用程序的前端和后端。第一部分针对前端主要介绍了如何发现和消除浏览器在呈现网页过程中的瓶颈，此外还介绍了很多PHP编码的最佳实践以及如何运用缓存技术。第二部分则介绍后端，详细讲解了很多类型的Web服务器软件以及如何优化它们。

本书从基准测试技术开始讲起，开篇即引入了两个测试工具：Apache Benchmark和Siege。然后由浅入深依次介绍了PHP代码优化、Opcode缓存、变量缓存、Apache Web服务器优化，以及最后的数据库优化。值得注意的是，基准测试几乎贯穿了整本书的所有章节，大部分章节中都穿插了ab和Siege的使用，并给出了详细的图解来说明测试的过程和结果，这是本书的一大特色。

毫无疑问，本书最适合PHP开发人员阅读，但同时也适用于对优化大型网络应用程序感兴趣的各类人员，包括项目经理、工程师和测试人员。

最后，衷心感谢人民邮电出版社图灵公司各位编辑在翻译工作中给予的帮助和宝贵意见。由于译者水平有限，在翻译过程中难免会出现一些错误，恳请读者批评指正。

谨以此书献给我的家人、朋友还有我的狗 Snoopy。

——Armando Padilla

谨以此书献给我的伙伴 Ester，他总是承受我埋头工作时的坏脾气，并忍受我残缺不全的回答。

——Tim Hawkins

引　言

如果你我是同道中人，你可能正在当地书店里拿起本书，或者正在网上读这份简介，尝试找找对这本书的感觉。要么你是一位好学的PHP工程师，急于探究构建大型应用程序的奥妙之处，要么你刚刚承担了一项支持高流量PHP应用程序的开发任务。这本书将非常适合像你这样对PHP有透彻了解并且熟悉这种语言的PHP开发人员——一位想深入研究PHP原理和工具集并且想揭开PHP脚本神秘面纱的人。

本书的目的是全面介绍在优化PHP应用程序时所需考虑的组件。它涵盖了所有这些组件，从JavaScript到正在运行应用程序的Web服务器软件。

本书分为两大部分，Web应用程序的前端和后端。第一部分介绍前端，帮助你确定在呈现过程中浏览器遇到的瓶颈以及如何消除这些瓶颈。这一部分还涉及PHP编码最佳实践的运用，如何使用很多可用的工具来应用缓存技术。第二部分介绍后端，介绍了很多类型的Web服务器软件，如何优化软件，以及优化数据库的技巧。

概述

下面是各章节内容的详细说明。

第 1 章：基准测试技术

我们首先确定测量应用程序性能所需的工具。我们将学习业界最受欢迎的两个基准测试工具的安装、结果读取和应用，这两个工具即Apache Benchmark（简称ab）和Siege。还将介绍如何使用并发以及特定时间段内的模拟负载来运行模拟负载实验。

第 2 章：提高客户端下载和呈现性能

应用程序性能不仅仅与PHP代码有关。这一章将重点介绍浏览器如何呈现内容。我们将学习对JavaScript进行基准测试的可用工具，测量浏览器尝试加载的数据量，以及查看浏览器加载内容的效率。我们还将学会如何通过安装和使用Firebug、Page Speed和Yahoo!的YSlow来完成这些工作。使用这些工具，我们可通过确认JavaScript、Image的性能是否提高来优化一个简单的网页。这一章内容比较浅显，所以并不要求你是JavaScript方面的专家。

第 3 章：PHP 代码优化

这一章开始研究PHP代码。我们将学习一些有助于提高PHP性能的最佳编码实践，还将了解如何构造一个快速运行的for循环，如何使用最佳PHP函数来包含文件，并且将重点了解如何使用和安装VLD、strace和Xdebug。在安装VLD和strace之后，就可以对Opcode以及运行PHP脚本所需的Apache C级别的处理进行分析了。使用Xdebug分析代码，我们将发现PHP代码自身的瓶颈所在。

第 4 章：Opcode 缓存

了解PHP的生命周期对于优化来说是非常重要的，所以这一章将介绍生命周期。我们将学习在用户请求期间PHP所执行的操作步骤，并找出有哪些地方可以使用Opcode缓存器进行优化。还将学习如何安装和配置Opcode缓存器，如APC、XCache和eAccelerator，同时还将对前后脚本进行基准测试，以便查看缓存Opcode后的收益。

第 5 章：变量缓存

在第4章介绍的缓存信息的基础上，我们将介绍变量缓存工具，如Memcached，以及使用APC存储信息。我们将学习如何安装、配置和实现一个简单的示例以便使你熟悉软件，另外还将介绍一个使用数据库结果集的实际示例。

第 6 章：选择正确的 Web 服务器

一直以来Apache都是一枝独秀，它是大型部署实际的标准。但最近该领域还出现了一些令人兴奋的新面孔。这一章将详细介绍Apache，并将其与新出现的Lighttpd和Nginx做一下对比。

第 7 章：优化 Web 服务器和内容交付

Apache是非常出色的Web服务器程序包，虽然它是开箱即用的，但稍加调整并且掌握一些技巧后，它的性能、持久性和稳定性会更高，它也会发挥出它真实的功能。这一章还将讨论如何对其进行扩展，以支持更高流量和用户负载的一些秘诀。

第 8 章：数据库优化

在大多数Web应用程序中，数据库服务器都发挥着极其重要的作用。这一章将讨论优化mysql数据库服务器的相关内容，同时还将提供使系统保持最佳状态的方法和工具。

致　　谢

不言而喻，我要感谢Apress的工作人员Jennifer Blackwell和Michelle Lowman，他们给了我写这本书的机会。还要感谢数不清的开发人员和系统管理员，在很多个深夜，他们为我解答了无数个与本书主题相关的问题。感谢你们。

——Armando Padilla

感谢Rasmus Lerdorf，是他启动了这本书的写作工作，他也教给了我许多APC的高超技巧，还要感谢雅虎欧洲公司的前任同事们，他们教会了我要有远见。

——Tim Hawkins

目　录

第1章 基准测试技术

电话响了，一个声音在另一端大声地说："嘿！为什么这个应用程序不能支持200个并发用户呢？"你深吸一口气，尽你所能以一位资深PHP人员的语调低声回答："奇怪，我看看出了什么问题，一会儿给你答复。"回想起这场对话的前几周。那时你的任务是构建一个由数据库驱动的PHP应用程序，据各方所说系统需求描述了一个简单的PHP应用程序。作为一名经验丰富的PHP开发人员，你开始编写代码，创建基本的体系结构层、PHP后端、CSS、JavaScript，并且还因为精通Photoshop，你还创建了图形布局并且发布了这个应用程序产品。

随着应用程序越来越受欢迎，网站访问用户与日俱增，很多抱怨也接踵而至。所有抱怨都指向了一个类似的问题，那就是网站无法响应或者响应速度太慢。现在，你看着代码，并考虑如何找到最后那点儿瓶颈和影响速度的代码并把它们除掉，不管问题到底是不是由代码引起的。你最终还是遇到了这个问题——当有50、100、200或300个并发用户同时请求主机上的文档时，你的Web应用程序的性能如何？此外，你如何在这样的流量负载下测试性能？

本章将深入考查两个用于基准测试的开源工具，它们不仅能够帮助回答这些问题，而且当我们对本书所讨论的应用程序进行性能增强时，它们还能测量性能的变化情况。我们要使用的这两个工具是Apache Benchmark（ab）和Siege。

我们将学习如何安装这两个工具，读取结果以及使用工具来获取不同类型的内容——从简单的HTML到大型图像。最后还将学习一些其他的基础知识，包括HTTP请求/响应生命周期是如何被处理的，一个请求到底执行了什么操作，以及在请求主机资源的时候应该主要关注哪些可能引起延迟（也称作"lag"）的区域。

但首先让我们来了解一下PHP应用程序栈和贯穿本书始终的一个方法。

1.1 PHP应用程序栈

每个PHP应用程序都有一个栈，可视化后，类似于图1-1。

大多数PHP应用程序都是在浏览器中显示给用户的，它们的显示使用了JavaScript（JS）格式的前端代码、层叠样式表（CSS）、Flash、其他前端技术以及诸如图像之类的资源。前端（如PHP应用程序栈的最上面一块所示）可帮助用户在Web应用程序中导航以及触发PHP层。PHP层包含特定于该应用程序的业务逻辑，并且在使用外部存储系统的情况下，它通常会与数据库或Web服

务交互以获取动态数据。最后，所有基于Web的PHP应用程序都有一个共同点：必须将它们安装在诸如Apache或Nginx这样的Web服务器上，而这些服务器也要安装在某个操作系统上。

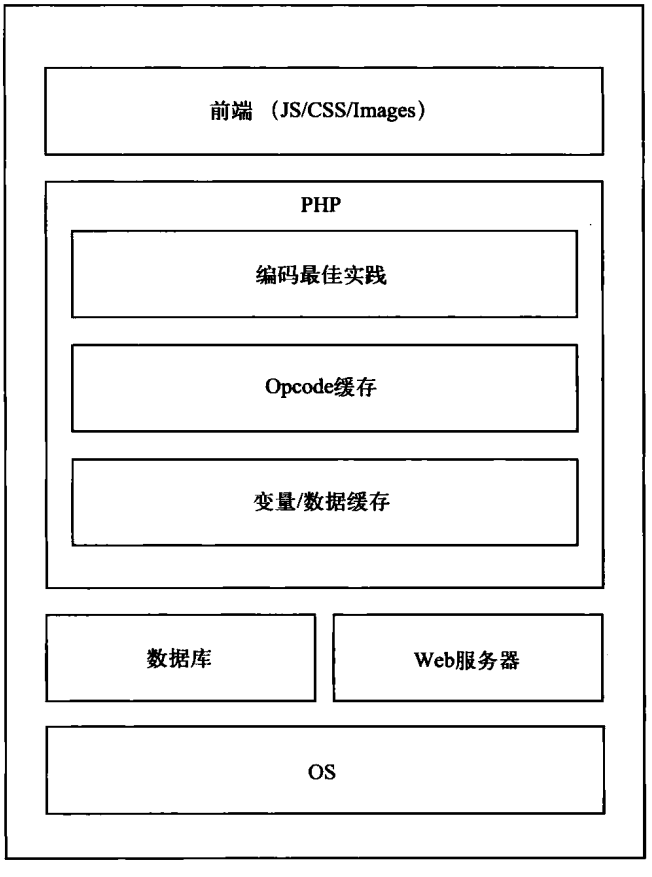

图1-1　PHP应用程序栈以及本书各章的划分

图1-1还表示出了本书所涉及内容的具体划分。PHP应用程序中的每一层都可以进行优化，而且每一层都是后续章的基础。从前端到Web服务器，本书将介绍图中所示的每一层，但我们需要一个工具，它不仅可用于测量当前未经修改的应用程序的执行情况，而且还可用于测量在对其性能进行提升之后它的执行情况。Apache Benchmark和Siege都满足了这样的要求。

1.2　基准测试实用工具

ab和siege同属于一组Web服务器基准测试工具，这些工具提供在各种不同的模拟用户请求发生时有关Web服务器响应的统计信息。它们允许我们模拟任意数量的请求Web服务器上某个特定Web文档的用户，更重要的是，它允许我们模拟任意数量的用户同时访问Web服务器上的文档

(并发请求)。

例如，每个工具提供的信息都与下列内容有关：

- 响应一个请求所花费的总时间；
- 来自服务器的总响应大小；
- Web服务器每秒可以处理的请求总数。

这些工具所不能完成的就是测试功能。这些工具仅能测试在特定Web服务器上运行的单个Web文档的请求。

之所以选择ab和siege，乃是由于以下原因。

- 易于使用：ab和siege都只有一行用于键入，同时只有少量选项。这意味着从一开始就很容易学习。
- 易于安装：它们都非常容易安装，并且只需要很少的安装时间。
- 基于命令行：大多数开发人员都在UNIX或Windows服务器上使用命令行。

1.3 定义请求/响应生命周期

让我们通过考查生命周期来快速探究一下HTTP请求/响应所执行的操作。首先，我们需要了解HTTP请求是什么，以及它所执行的操作，因为这些工具会利用一个请求的生命周期来帮助测量应用程序的性能。

HTTP请求是用户或工具尝试从Web服务器获取内容时所要采取的操作。典型的HTTP请求包含请求正在尝试访问的主机信息、浏览器信息以及对Web服务器有用的其他信息。图1-2显示了用户个人计算机的HTTP请求/响应的过程。

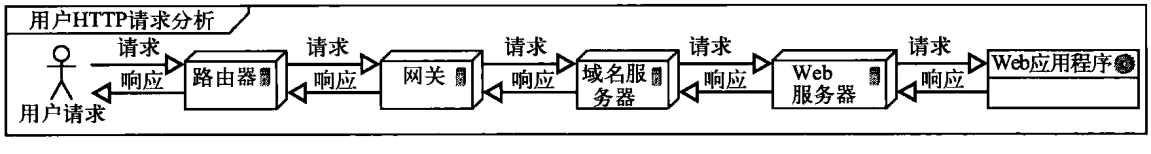

图1-2 HTTP请求生命周期

上图显示了一个简单的用户HTTP请求的过程，它向Web服务器请求获取内容。该请求是从用户的计算机生成的，依次要经过用户的主路由器（如果有的话）、ISP网关和域名服务器（DNS），并在DNS中会查找与请求的域名相关联的IP，然后到达具有指定IP的Web服务器，并最终请求Web应用程序生成特定内容。

该生命周期的第二部分是HTTP响应。一旦请求到达Web服务器之后，Web服务器便通过获取并格式化用户请求数据来准备响应；然后Web服务器就会将数据打包成多个数据包，并以相反的顺序沿着用户请求的相同路径将这些数据包发送给用户。如果数据足够大，则采用多个数据包发送，检查在传送期间是否有错误并由浏览器重建，然后浏览器便可开始其呈现过程。所有这些步骤必须发生在浏览器呈现网页之前。

这些步骤中的每个步骤都会使最终用户的网页执行速度减慢。接下来要讨论的工具可用来测试应用程序的响应时间，从而测试其最佳状态。

1.4　Apache Benchmark

Apache Benchmark（ab）工具是最著名的基准测试工具之一，它是默认的Apache安装的一部分，能够通过模拟对特定URL的任意数量请求来对Web服务器进行负载测试。ab工具提供以下信息：

- 传输的总数据大小（以字节为单位）；
- Web服务器在模拟流量下每秒可以支持的请求总数；
- 完成一个请求所花费的最长时间（以毫秒为单位）；
- 完成一个请求所花费的最短时间（以毫秒为单位）。

使用ab工具还可以运行很多不同的负载模拟，例如：

- 对Web文档的模拟请求；
- 指定时间内的请求；
- 打开Keep-Alive时的请求。

最重要的是，Apache Benchmark是独立于Apache Web服务器的，从而可以在运行ab的同时使运行此工具的计算机上的Web服务器处于非活动状态。

1.4.1　安装 Apache Benchmark

在下面两节中，我们将介绍如何在基于Windows和Unix的系统上安装运行ab工具所需的文件。

1. Unix和Mac安装

如果用的是*nix操作系统，则会有很多安装Apache的选项。可以通过ports、yum、apt-get安装或只是下载源文件并安装。表1-1显示了安装命令的完整列表。

表1-1　使用存储库来安装Apache Web服务器

存 储 库	命　令
yum	`yum install apache2`
ports	`sudo port install apache2`
apt-get	`apt-get install apache2`

Mac用户可以在终端上使用MacPorts并执行表1-1所示的基于ports的命令。

2. Windows安装

Windows用户可在浏览器中打开http://httpd.apache.org/。加载此页之后，单击页面左侧的“Download from a mirror”（从镜像下载）链接，找到适合你的系统的相应下载程序包，即Windows 32 Binary版本，然后下载。编写本书时，最新的Apache版本为2.2.X。

当程序包下载完之后，就可以通过运行安装向导在系统的任意位置上安装该软件。我将Apache安装在默认位置C:\Program Files\Apache Software Foundation，但也可以安装在系统的任意位置。此处所选择的位置就是APACHE_HOME引用所指向的位置。

现在，打开目录<APACHE_HOME>\Apache2.2\bin。应该可以看到类似于图1-3的文件和目录的集合。

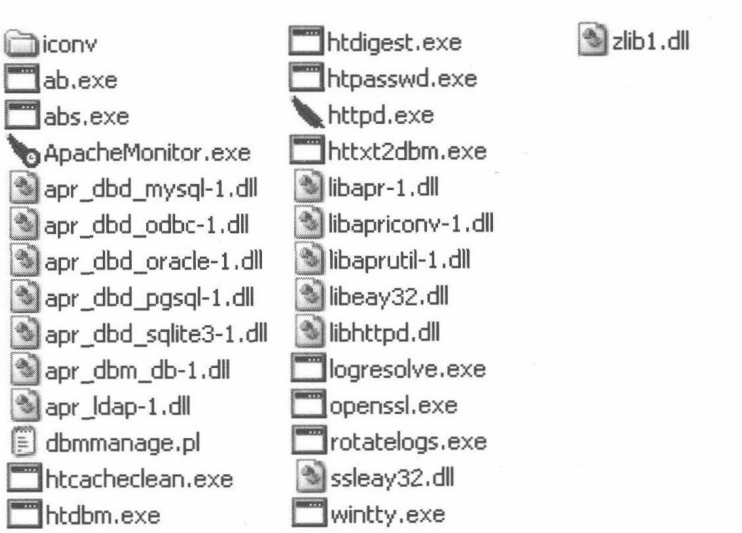

图1-3 安装好后的Windows Apache的bin目录

你已成功安装ab了工具，现在让我们使用它。

1.4.2 运行 Apache Benchmark

我们要运行的第一个基准测试是在www.example.com域上的简单测试。这个基本测试的主要目的就是要让你熟悉以下工具的语法，查看所有的可用选项以及查看完整的响应。

所有ab命令的组成遵循此结构：

```
ab [options] [full path to web document]
```

我们将使用ab语法模拟单个请求。打开命令/shell终端并键入以下命令：

```
ab -n 1 http://www.example.com/
```

这条命令只使用了option部分的一个选项，即n，它表示要在指定的URL上执行的请求数。在这个示例中，ab只请求Web文档一次，但n的值可以是小于50 000的任意数字。默认情况下，n设置为1。

该命令的下一部分是URL部分。对于刚刚执行的ab命令，URL为http://www.example.com/。如果选择测试此域中的某个文档，如test.php（并不存在），则要测试的URL将为http://www.example.com/test.php。

让我们回到用于执行ab命令的命令/shell终端。到现在为止，你已经执行了这个命令，并且屏幕上满是ab工具所返回的数字和常规数据。你的输出应该类似于图1-4。

```
$ ./ab -n 1 http://www.example.com/
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.example.com (be patient).....done


Server Software:        Apache/2.2.3
Server Hostname:        www.example.com
Server Port:            80

Document Path:          /
Document Length:        574 bytes

Concurrency Level:      1
Time taken for tests:   0.094 seconds
Complete requests:      1
Failed requests:        0
Write errors:           0
Total transferred:      860 bytes
HTML transferred:       574 bytes
Requests per second:    10.67 [#/sec] (mean)
Time per request:       93.750 [ms] (mean)
Time per request:       93.750 [ms] (mean, across all concurrent requests)
Transfer rate:          8.96 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       47   47   0.0     47      47
Processing:    47   47   0.0     47      47
Waiting:       47   47   0.0     47      47
Total:         94   94   0.0     94      94
```

图1-4　URL http://www.example.com的ab响应

注意　测试其他计算机时，请小心谨慎并且同时限制对Web服务器以及测试所发出的请求数量。我们都不想损害任何无问题的服务器而陷入真正的麻烦中。

1.4.3　弄清响应的含义

如果你头一次见识刚刚显示的输出中的响应，或者就算以前看到过，还是可能会觉得它有点难以理解。我们来看看其中一些非常重要的项目，以及一些在优化代码时有助于了解优化效果的项目。

图1-4中的数据分为4个主要部分，如图1-5所示。

1. 服务器信息

服务器信息部分包含Web服务器运行的软件。在我们的示例中，软件为Apache 2.2.3。数据包含在第一个字段即Server Software中。该字段的值可能会因为网站所使用的Web服务器软件而存在差异。由于Web管理员使用的安全措施，该字段的值也可能会返回一些你不熟悉的内容。